Unit 4

Description of Material:

This unit reviews and reinforces the programming concepts covered in the past two units, especially data and types. It also introduces the idea of persistence and how to use TinyDB and TinyWebDB components. It explores techniques for debugging programs.

Learning Objectives:

- understand the definition data persistence and what that means for mobile applications
- predict the results of the program named Data
- utilize TinyDB an TinyWebDB in an app inventor program
- assess the errors in a program and fix them (debug code)
- recommend sources of solutions

Let's Begin!

This Unit is all about data. Data is set of facts that can be processed. In the context of mobile apps, data is your score, your username and password combo, your friends on the app, the contacts on your phone, etc. Data persistence is the idea of an app remembering data about you after you exit or close the app. This is often achieved by using databases to store any information the app wants to remember. Remembering your high score, your username, how often you play, your location, and more might all be information an app would store in a database.

This unit you'll learn how to create databases using App Inventor. There are two main components that you use to do this: TinyDB and TinyWebDB. Read more about databases and these components in the programming concepts section below.

Programming Concepts -- Databases

Recall how using variables, lists, and the "get" blocks for component properties allowed you to access and save certain information. This information is also called data. This data can be very useful in an app. But if any of the variables or component properties change while using the app, none of them will remember that new value when the app is closed and opened again. To remember this data, we need data persistence. We can achieve this by using a database.

Almost all apps that we use today store data of some kind in a database. Apps that require accounts have a database of your username, password, and possibly email. Game apps often have a database of all players and their corresponding high score. A phone's camera app has a database to store all of your photos taken previously. Without this database, anything you do in an app would be lost the second you exit it. Databases allow apps to have *persistent data*, the characteristic of state that outlives the process that created it

To experiment with both persistent data by using App Inventor's database, TinyDB and other data stored in variables or component properties, download this aia file called Data, and upload it into your projects. Download the app to your phone (or just run using the emulator) then complete the following tasks.

- Click the different buttons and see how the screen changes color. Exit the app and reopen it. What color is the screen?
- Now download this source code, Data_part2, which is the same app just with the TinyDB component and a few different blocks. Click the different buttons. Then exit the app and reopen it. What color is the screen?
- Think about the difference in blocks and how this affected the color.

App Inventor Concepts -- TinyDB

TinyDB is a non-visible component that you can use to store data on your phone. Every phone has a small database that the TinyDB component accesses to store and retrieve information.

Apps created with App Inventor are initialized each time they run. If an app sets the value of a variable and the user then quits the app, the value of that variable will not be remembered the next time the app is run. TinyDB is a *persistent* data store for the app, that means that the data stored there will be available each time the app is run. One example is an all-time high score for a game app - it's saved on the device each time the user plays, and retrieved each time the app is opened.
Data items are stored under *tags*. To store a data item, you specify the tag it should be stored under. Subsequently, you can retrieve the data item that was stored under a given tag. If there is no value stored under a tag, then the value returned is the empty text. Consequently, to see if a tag has a value stored under it, test whether the return value is equal to the empty text (i.e., a text box with no text filled in).

TinyDB can be used to store information you want an app to remember about the owner/user on that phone. A personal high score, who is logged in, or what your location is are types of data you might want to store. Play around with this version of MoleMash that uses the TinyDB component to see how to use it in your own apps.

If you'd like to read more about databases and TinyDB, check out this corresponding reading. ***Note that this guide was made for App Inventor Classic so the blocks are formatted slightly differently.*

- Chapter 22 Databases

App Inventor Concepts -- TinyWebDB

TinyWebDB is a database on the cloud. This database is much bigger than the one on your phone and can also be accessed by any user of the app. It is not phone specific. Because the data is stored on the web instead of a particular phone, TinyWebDB can be used to facilitate communication between phones and apps (e.g., multi-player games). Thus TinyWebDB can be used to store information about all users of the app.

Similar to TinyDB, TinyWebDB also stores data items under tags. So if you haven't read the section on databases and TinyDB above, please read that first.
Since TinyWebDB stores data in a database on the web, you may be wondering where on the web this data is stored. By default, the TinyWebDB component stores data on a test service provided by App Inventor, http://appinvtinywebdb.appspot.com/. This service is helpful for testing, but it is shared by all App Inventor users, and it has a limit of 1000 entries. Because so many people are using this database, it often overwrites past entries. Thus it is not useful for any apps that will be used frequently. To create your own database, follow the provided instructions on the App Inventor website. [*Note that this is a more advanced/challenging App Inventor feature*]

Play around with this version of MoleMash that uses the TinyWebDB component to see how to use it in your own apps.

Challenge Time -- App Creation

To practice using TinyDB and TinyWebDB, check out the provided source code and tutorials. Your assignment for this unit is to create a new app or a modified version of an app that uses TinyDB or TinyWebDB.

If you're stuck and are unsure where to start. Here are some ideas for apps:

- Modify an app you built in the past such as your game app to store the high score in the game
- Modify the Magic8Ball app to allow users to enter new fortunes and save them for future uses of the app.

Programming Concepts -- Debugging

Another important concept in programming is learning how to debug your code. Remember in the unit 1, you learned that in programming, we call bugs the problems in your code. Perhaps clicking a button brings you to the wrong page or calls the wrong person or doesn't bring you anywhere at all; these are all bugs. This unit, we'll be giving you further instructions on how to debug your and your students code.

To get into the heart of debugging, feel free to review Chapter 15 from the App Inventor book on engineering and debugging. We also recommend to follow along the provided debugging instructions available on our website.

- Debugging instructions from the App Inventor site
- Chapter 15: Engineering & Debugging

Find the bug in this app: MiniGolf (link to .aia file). Here is a corresponding tutorial for MiniGolf if you'd like some extra help.

Challenge Bug Finding *[Optional]*

This is an app, AgNoteMAX, from a group that created an app last year for the App Challenge. Try to see if you can find the blocks that cause the bug. The bug is that the stock screen contains a problem with displaying lists. It works fine at first but when you click the button to refresh, the price list and the stock list aren't in agreement anymore. The bug is on the "Stocks" screen of this app. To get to the "Stocks" screen, click the "Screen1" button located to the right of the name of the app and select the "Stocks" screen.

(This bug even stumped some of the course staff, so if you'd like an explanation of the bug and how to prevent it, check out this document).

Assignments

App Inventor Assignments

- Play with sample app to learn about data in App Inventor
- Create your own app that uses a database
- Find the bug in the corresponding app.