Procedures and the Logo 1 App

Logo is a classic drawing app that was one of the first programming environments for kids. You can use code to draw using some simple primitive blocks like *forward* and *turn*. We'll look at two versions of Logo in order to learn how to break down a problem with procedures. In Logo 1 the following procedures (blocks) have been defined:

forward -- moves the android ahead 20 pixels
turn -- rotates the android 90 degrees
penup -- if the pen is up you can move android but not draw
pendown -- if the pen is down drawing happens on forward.

Instructor Demo

use the UI to draw something, go over primitives.
programming the "draw" procedure to draw a square
creating a new draw2020 procedure and calling it from draw

Worksheet

1. Open the Logo Starter app, ai2.appinventor.mit.edu/?galleryId=5222319220326400, Play around with drawing things with the user interface by clicking the buttons at the top. Can you draw a square?

2. **Code** a procedure named ***square4040*** that draws a 40-by-40 square. Drag a *to* block from the procedures folder, the one without the return, name it square4040, then add blocks within it to call forward and turn a number of times to draw the square. To test your procedure, modify the *draw* procedure so that it calls your *square40* procedure (note that when you created square40, a new call block was inserted into the procedures folder). Now try clicking the "Draw" button--> it should draw your square.

3. Code a procedure named **twoSquares** that draws two 40x40 squares side-by-side. TwoSquares should call square4040 twice (amongst other things). Change the draw procedure so it now calls twoSquares.

4. With a partner, answer the following questions:

a. In your code, how did the use of procedures eliminate redundancy?

b. Why was the coding square4040 repetitive? Say you wanted to draw 10 squares of different sizes stacked upon each other. Would this be repetitive? How could it be coded differently?

c. Can you use the existing *forward* and *turn* procedures to draw a rectangle with sides of 35 pixels? Why or why not?

d. Can you use the existing *forward* and *turn* procedures to draw a triangle? Why or why not?

Want to learn more. Check out this longer Logo 1 lesson at:
https://docs.google.com/document/d/13_Y6ynzk_NESv4BgO4TkAByUr4Ttb20ZOOBp3hmygfk/
[Video lesson for Logo]()