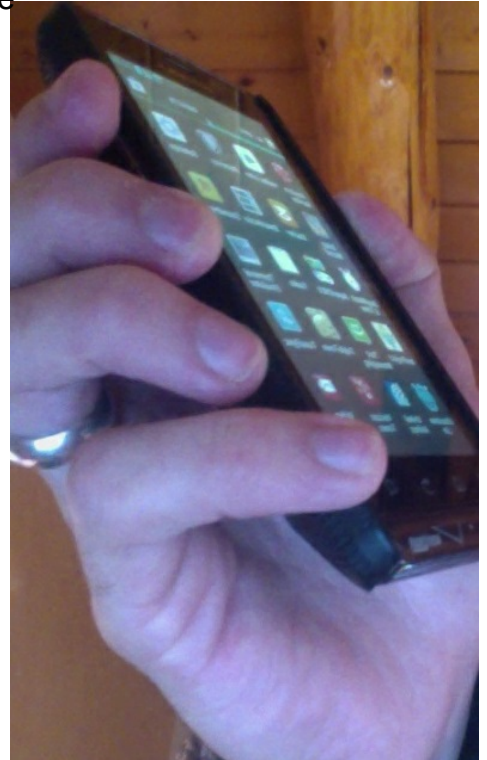


# Analyzing a Program

## Introduction

Now that you have a basic feel for App inventor, it's time to explore it more deeply through the lens of our eight roles of variables. Remember that these roles are: fixed value, most recent, accumulator, aggregator, stepper, walker, best-so-far, and one-way flag.

This activity includes complete source code for an AI2 App. You are provided with this code at six different stages so that you can see how complexity can be added to a program over time.



## Materials

- Computer with Internet access
- Android device or emulator

## Resources

[1.2.4 sourceFiles.zip](#)

[1.2.4.Aa Blocks.docx](#)

## Procedure

## Part I: First Look at New Code

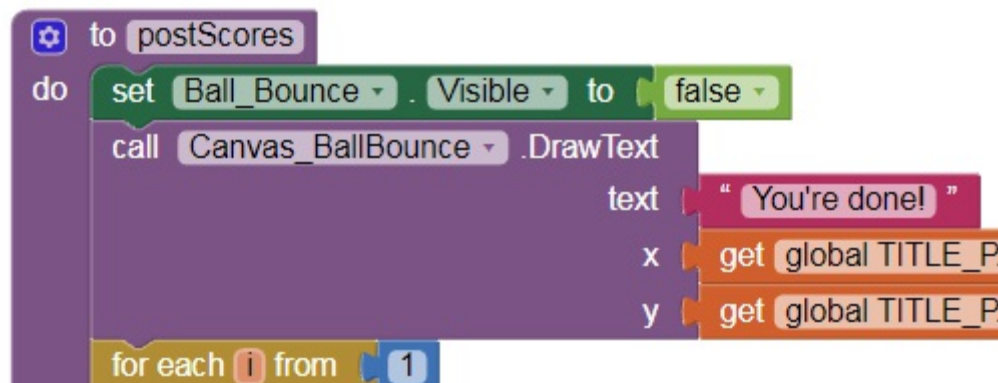
1. Form pairs as directed by your teacher. Meet or greet each other to practice professional skills.

As there is little “driving” to be done in this activity, you may wish to have the code open on both computers to make navigating and analyzing the code easier.

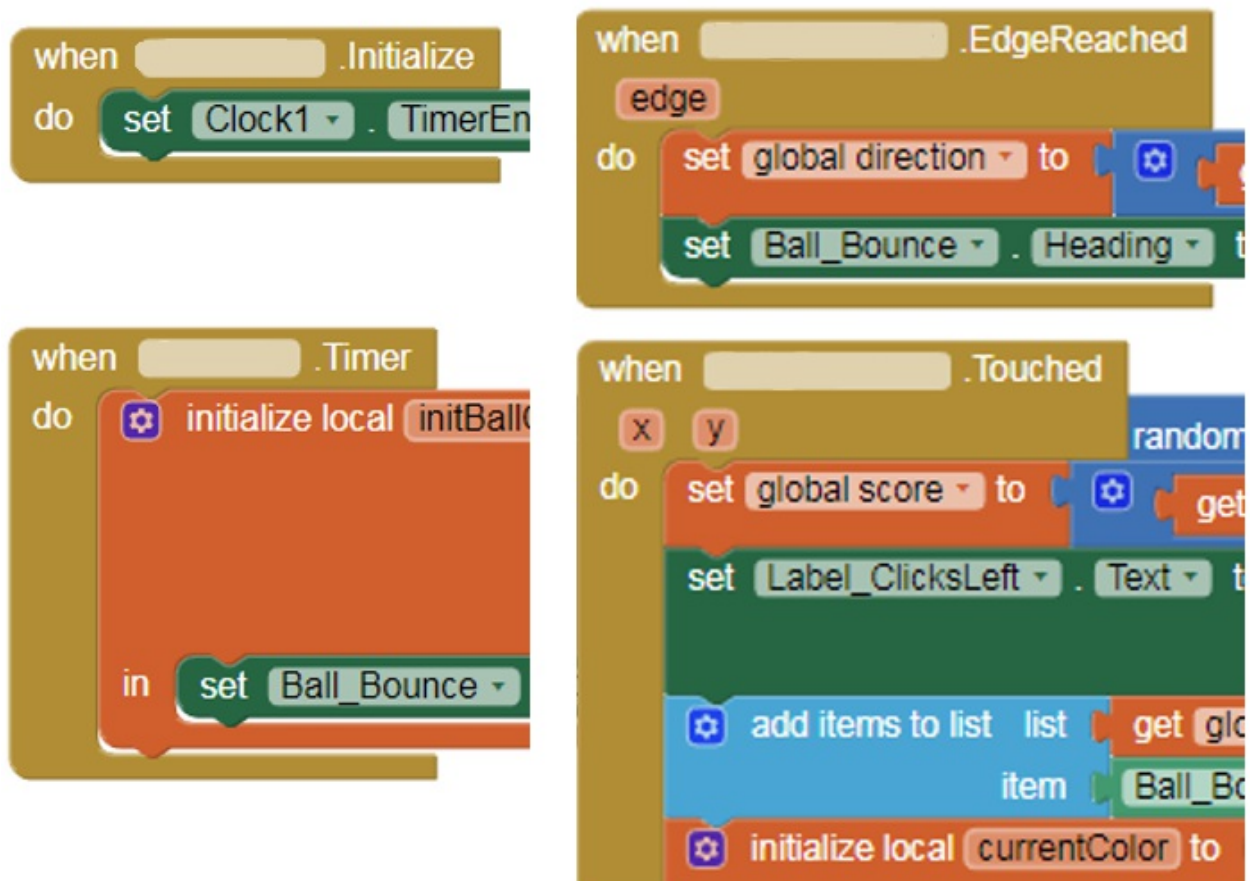
2. Download the source code and place “bounceBall.aia” on your computer, according to your teacher's instructions.
3. Using Google Chrome™, navigate to <http://ai2.appinventor.mit.edu/>. Log in to your Google account, making sure that the sign in screen is using the https protocol from the google.com domain.
4. Choose Continue to move beyond the splash screen. Choose **Project>Import project (.aia) from my computer**. Browse to select your copy of “bounceBall.aia” that you downloaded in step 2.
5. Open the program by selecting its name in the projects view.
6. Once the program opens in the GUI Designer view, choose Blocks to switch to the Blocks editor.
7. Compare the behavior of the app on your Android device to what you see on the **Blocks** screen in App Inventor. Analyze the program following these steps:
  - There are many global variables in this program. Each one is created by a single block as shown below. A global variable can be used in any part of the program. How many global variables are there?



- The larger chunks of blocks are either procedure definitions or event handlers. Procedure definition blocks begin with “to” and have the procedure name as shown in the figure below. Look through the code to find and fill in the name of this procedure.



- The other large chunks are event handlers. These tell the program what to do when the user or program causes an event, like a mouse click, to occur. The event is described after the “.” in the name of the block. Look through the code and fill in the names of the “when” events that are being handled by the chunks of code shown below.



when \_\_\_\_\_ .Initialize  
 when \_\_\_\_\_ .EdgeReached  
 when \_\_\_\_\_ .Timer  
 when \_\_\_\_\_ .Touched

8. Choose one block of code that is accounting for some behavior. Discuss with your partner what parts of the block you think are accounting for the aspects of that behavior.

## Part II: Variable Roles in App Inventor 2

The program that you just examined is the final version of a program that went through many stages of modification in order to get to its current form. Acquire “bounceBallStage1.aia” and open it in App Inventor in the same manner as described in steps 2 through 6 above.

9. The variables used in the first stage of the code’s development are listed in the table below.

Referring to the code for this stage shown in the supplementary material *1.2.4.Aa Blocks* or in the source code seen within the App Inventor IDE, identify the roles of each of the following variables.

Explain what purpose each variable serves for the program as a whole. Your teacher might direct you to examine the stage 1 program running on an Android device.

Variable Name	Role	Purpose
Direction		
TO_FLIP		

10. Load the stage of the program's development “bounceBallStage2.aia” following steps 2 through 6. Identify the role fulfilled by each of the newly introduced variables listed below.

This stage of the code introduces a **concatenation** block. Concatenation is the word we use to describe attaching two things side-by-side. Concatenation is used commonly to put two strings of text side-by-side, converting them into one string of text.

Variable Name	Role	Purpose
Score		
MAX_BALLS		

11. Load “bounceBallStage3.aia” in the block editor. As in previous steps, identify the newly introduced variables' roles and explain their purpose.

This stage in the code introduces **pseudorandom numbers**. **Pseudorandom** numbers are the “random numbers generated by a computer program. Although they are unpredictable because they are constructed from the computer's memory using parts of memory unrelated to the program, they are not actually random. We perceive pseudorandom numbers as being random because App Inventor takes care of “seeding” the sequence by picking an unrelated location in memory using the system clock.

Variable Name	Role	Purpose
currentColor		
colorsClicked		

12. Load “bounceBallStage4.aia” and repeat for the new variables listed below.

This stage introduces using a procedure to break up code into more manageable and logically arranged chunks. A procedure is a separate piece of code that can be called upon from

multiple locations in your code. This procedure handles the displaying of scores once the game is finished.

**Padding** is a common name for blank space placed around the elements of a display, and these variable names incorporate this term.

Variable Name	Role	Purpose
TITLE_PAD_X		
TITLE_PAD_Y		

13. Load “bounceBallStage5.aia” and repeat for these variables:

Variable Name	Role	Purpose
i		
RED_CHANNEL		
GREEN_CHANNEL		
BLUE_CHANNEL		
SCORE_PAD_Y		
SCORE_PAD_X		
blueValue		
greenValue		
redValue		

14. Open the supplemental resource *1.2.4.Aa Blocks*. The steps of the pattern associated with each role are shown below. Choose a variable that serves each role. Record the line numbers from the final program that demonstrate each step for each role.

- Fixed Value:
- Assign value:
- Use value:

- Most Recent:
- Assign value:
  - Use value:
- Stepper:
- Initialize value:
  - Increment value:
- Walker:
- Assign element from collection:
  - Use element:
- Accumulator:
- Initialize:
  - Accumulate:
  - Use accumulation:
- Aggregator:
- Initialize:
  - Aggregate:
  - Use aggregation:

### Part III. Extension (Optional)

15. Adding in variables that take on the roles of Best-so-far and One-way flag will certainly make the program a little more engaging, but they also add a significant amount of complexity to the code for the program. If you have extra time, you should open “bounceBallStage6.aia” in the blocks editor and fill in the table below. In addition, complete parts a. and b. as you did in the last step.

Variable Name	Role	Purpose
colorRed		
colorGreen		
colorBlue		
splitColor		
currentRed		
mostRedYet		
sufficientRed		
sufficientRedSeen		

16. As you did in Step 14, refer to the supplemental resource *1.2.4.Aa Blocks* to record the line numbers that demonstrate each step for each role.

- Best-so-far:
- Initialize to impossibly bad:
- Check for record breaker:
- Set new record:
- Use overall record:
- One-way flag:
- Clear flag:

- Check for condition:
- Raise flag:
- Check if flag ever raised:

## Conclusion

App Inventor 2 and Scratch are two different programming languages. They are similar in look and feel. The roles of variables that you have learned in these two languages will appear again in any computer language.

1. For each of the eight roles of variables, write about an app that you use or that you would like to make that could or would need to use a variable to fulfill that role. You might be able to think of one program or app that uses all 8 roles for variables. Explain how each role will play an important role in your app.