PLTW COMPUTER SCIENCE

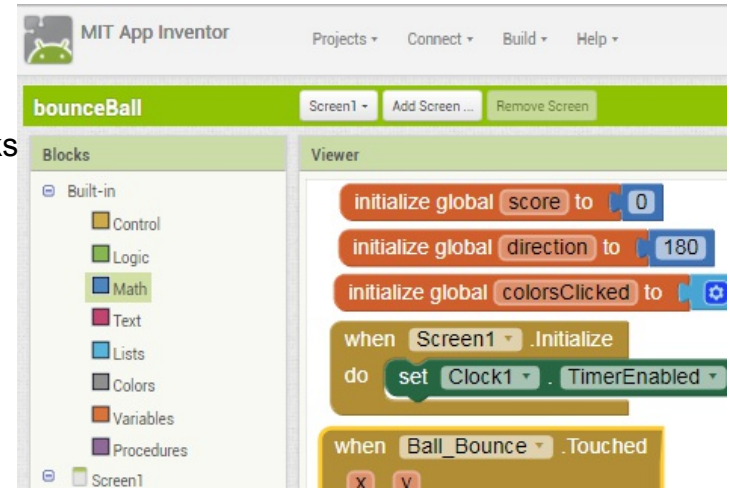
Project 1.2.5

Modifying Code

Introduction

Modifying other people's code accounts for a significant part of a software developer's work. The ability to understand code that has been written by someone else and readily modify it to suit your needs will allow you to accomplish tasks much faster than if you had to create an entire program from the ground up.

Can you think of changes you'd like to see in a program you use?



MATERIALS

- Computer with Internet access
- Android device or emulator

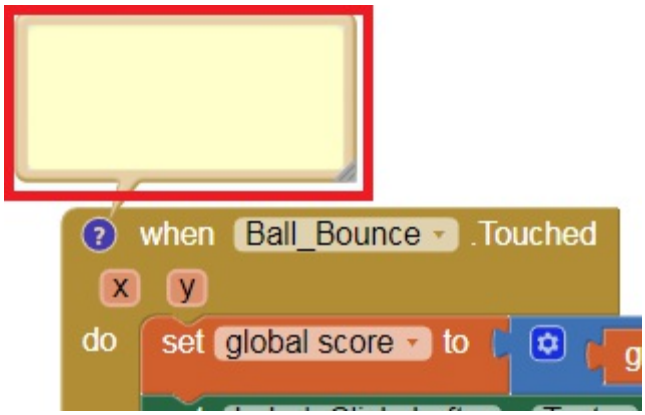
Procedure

Part I: Discovering the Program

1. Form pairs as directed by your teacher. Meet or greet each other to practice professional skills.
2. Open “bounceBall.aia” in the blocks editor. (Refer to Activity 1.2.4 if you do not have this.) You probably still have the program in your App Inventor projects view from the previous activity.
3. Spend a few moments reacquainting yourselves with the code. Using the roles of variables as a focusing point may help you to more quickly decipher what various parts of the code are doing.
4. Add comments into the code to help you remember what different parts of the code are doing. Right-click on a block of code to comment and select **Add Comment**.

Click on the blue circle with the question mark to show the comment text block where you can type.

Type your comments in the box that pops up.



Part II: Errors in App Inventor

Before you begin making modifications of your own, it's important that you know how to identify and fix some of the more common errors that you may introduce into a piece of code by changing it. Error detection and correction are a vital part of programming for everyone who creates code. The best programmers in the world still make errors; one of the qualities that set them apart from other programmers is their ability to quickly detect and eliminate those errors. Here you will perform a few small changes to the program and examine their effects to give you a sense of what this process is

like.

Note: Remember to undo your changes before moving on to the next error test.

5. First we will create an error by giving the wrong kind of value to an object property. Find the Clock1.Timer event handler:

6. Change the variable in the get block from “initialColor” to “colorsClicked”.

7. Before you try running the program on your device or emulator, answer the following questions:

    - What do you think `set Ball_Bounce.PaintColor to` is meant to do?
    - What does `initialColor` represent at this point in the program?
    - What does `colorsClicked` represent at this point in the program?
    - Why do you think making this change might be a problem?

8. Run the program on your device/emulator and record the text that appears in the error window here exactly as it appears there.

9. The error that you just saw is the result of an assignment from an improper type. The type of a variable refers to what kind of thing it is. In the case of our example, initialColor has a color stored in it, whereas colorsClicked is an empty list at the point where the code generates an error. The types of data that you will be managing in this program are typically integers, lists, colors, text, or Boolean values. For each of the variables below, document the type of data that it contains.

Variable Identifier	Type of Data
RED_CHANNEL	
direction	

colorsClicked	list
initialColor	color

10. When setting the various properties of objects in a program, a specific type of data is usually expected. In our example Ball_Bounce.PaintColor expects to receive a color. What type of data do each of the following expect?

Object and Property	Type of Data Expected
Ball_Bounce.Heading	
Ball_Bounce.Visible	
Canvas_BallBounce.PaintColor	
Ball_Bounce.PaintColor	color

11. There are other parts of the program that expect a certain type of data in order to operate properly. Within the Ball_Bounce.Touched event handler, change the list creation block so that it looks like the one shown below.

12. Run the program. You will need to click on a ball at least once to get the error message. You may need to scroll around within the error message window in order to see the whole message. Record the message here exactly as it appears on your device/emulator.

13. This is also a type error. The reason that this particular error occurred was that “make color” expects a list containing exactly three elements, all of which must be real numbers. The two examples you've just seen are runtime errors. This means that they cannot be detected until the program begins to run. Another important type of error is a logical error. Logical errors occur when a programmer has made a mistake reasoning out how the program should work. Within the Ball_Bounce.Touched event handler, remove the call postScores block from the conditional block.

14. What effect do you think this will have on the program when you run it?
15. Run the program and describe how it behaves differently than it did before you removed the call block.
16. Change the initial value of `GREEN_CHANNEL` to 1 and the initial value of `BLUE_CHANNEL` to 1. What effect do you think this will have on the program?
17. Run the program 4 times and record the values shown on the score screen here. Remember that the score values reflect the red value of the color of the ball minus the sum of its green and blue values, meaning a very red ball should have a positive score.

Run #	Score 1	Score 2	Score 3	Score 4	Score 5
1					
2					
3					
4					

18. What do you notice about the values that you see? How do you think the changes that you made to the program affected this?

Part III: Modifying the Program

Choose several of the following tasks to complete as directed by your teacher. Typically, each team will complete three to five of these tasks.

Value-Tweaking Tasks:

These tasks require only some small modification to the code, usually only a change in one or two values.

1. Modify the program so that the color of the ball never includes any green.
2. Modify the program so that the ball starts out moving to the left instead of to the right.
3. Modify the program so that the ball is twice as large, making it easier to click.

4. Modify the program so that it allows ten user clicks instead of only five.

Tasks requiring some strategizing:

These tasks require deeper understanding of the code that has already been written, but they still only require small changes to the code.

1. Change the code so that, in addition to displaying the score, the program also displays the number of times that the ball has changed colors.

2. Change the program so that the color of he ball changes every time it hits an edge of the screen.

   Hint: Before attempting this task in any other way, first try using a “set” block from the variable built-in palette to change the value of “currentBallColor” in the “EdgeReached” event handler. You will notice "currentBallColor" is not an option.

   The variable “currentBallColor” is a **local** variable. A local variable can only be used in a specific set of blocks, known as the **scope** of the variable. Global variables have a scope that includes the entire program. Using a local scope helps prevent many problems associated with large complicated programs, but to complete this task you will need to create a global variable to replace “currentBallColor” and make some other small additions to the code.

Tasks Making a Larger Contribution:

These tasks will require significant modifications to the code beyond simply changing one or two variable values or adding or removing a couple blocks.

1. Modify the code so that the ball travels in a different pattern of your choosing.

2. Add a piece of text directly over the ball that counts milliseconds to add a feeling of rush to the game. This piece of text should stay in the same position relative to the ball regardless of where the ball moves.

3. Modify the program so that instead of using a Ball to animate the game, the program uses the DrawCircle Canvas method.

Conclusion

1. Choose three of the tasks that you worked on during this project and explain the roles of variables you modified.

2. In a journal style entry, discuss what it was like working with code that was developed by someone else. Following are some key points to address. If you think of something else that you thought was very important, please discuss it as well.

   - There were no comments in the original code. How did this affect your ability to work?
   - How did the comments you added to the code in your initial previewing help or hinder

you as you worked on the tasks?

- What was it like working with another person to modify code that neither of you had written?
- What was the toughest challenge that you faced?
- What were the advantages of working with code that had already been somewhat established?
- In your initial readings of the code, did focusing on the roles of variables help you to find meaning?