Lesson 5.6: Popping Balloons - Part 3

Objectives

In this lesson, students will:

- Experience building a program by experimenting and iterating.
- Be able to describe what a variable is and what they are used for.
- Be introduced to the computational concept of data.
- Practice using a variable by keeping score in the Popping Balloons game.

Preparation

- ❑ Print student activity worksheet (one per student pair)
- ❑ Become familiar with the solution project

Agenda

1. Adding Score to our Game Using Variables	15 mins
2. Setting the Score	10 mins
3. Student Activity - Keeping Score	15 mins
4. Wrap Up and Reflections	10 mins

Resources & Links

- ❑ Link to solution project: https://scratch.mit.edu/projects/277805206

1. Adding a Score to our Game Using Variables

Engage students in a teacher led discussion and explanation:

What important game feature is missing from the game?

The score allows the game to track how well the user does playing the game. It can also be used to compare players ability by comparing their score.

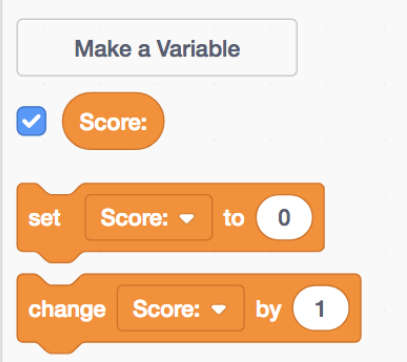
To keep track of a player's score, programming languages like Scratch use what is called a **variable**. A variable is a placeholder in the computer's memory to keep track of numbers, text, pictures and more. Variables are like boxes that hold stuff. In programming the content of a variable is called data. Every variable has a name which is like a label for the box that holds the data.

Variable name is Treasures.
The value or data is $100.

Demonstrate how to create the variable for the balloon game.

To create a variable in Scratch do the following:

1. Go to the ***Variable*** category and click on ***Make a Variable***
2. Give the variable a name, in our case we will call it Score:. Click **OK.**
3. Notice the variable shows up on the stage. You can move it to a different location by dragging it.

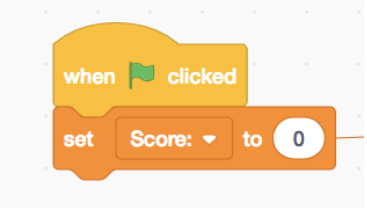


4. In the Variables category are statements to initialize the variable (set Score to 0) and to change the value (change Score by 1).

You can change the location of the variable by dragging it to a different place on the stage.

When the game starts, we want the initial value of our Score variable to be 0. Prompt students how that might be coded. You can initialize the score in the balloon sprite.

Solution:



Remind students that the code that sets all initial values and starting positions is called **initialization code**.

Students Turn:

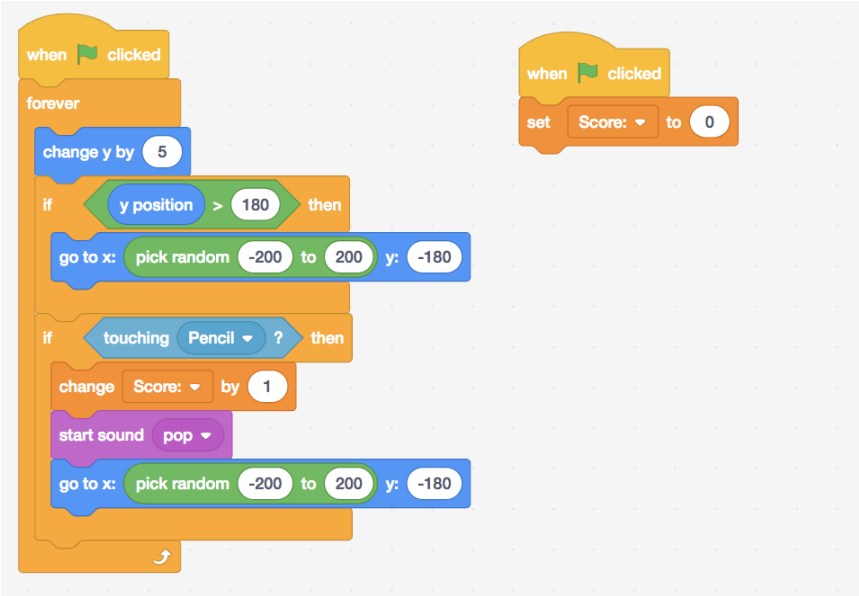
Students open their balloon game projects and create a variable called ***score*** and initialize it to 0.

2. Setting the Score

In our game, the player earns a point each time they pop a balloon.
Prompt students to point out where in the code we know the balloon has been popped.

When the pencil touches the balloon, it means the balloon has popped. We add the block **change Score by 1** to add a point to the Score (see solution for where to place the block).

Solution:



If we wanted to subtract 1 from the score, how would we do that? We would use -1 (negative 1).

3. Student Activity: Keeping Score

Distribute the ***Keeping Score*** activity worksheet. It might be helpful to explain what the activity is about.

Extended Activity: Adding a Timer

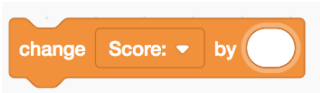
For more advanced students or if time permits, they can move on to the extended activity.
The solution to the extended activity is in the solution project.

4. Wrap Up and Reflection

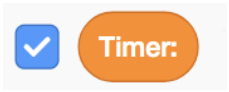
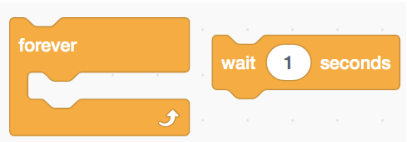
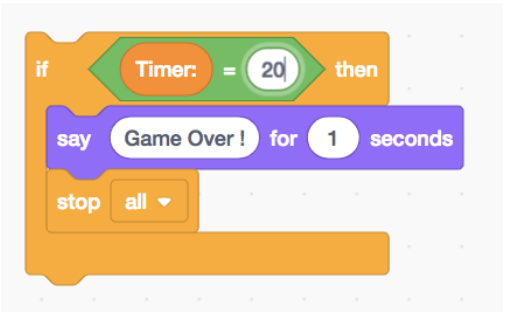
Reflection Points:

- What did you learn during this lesson?
- What are variables and what are they used for?
- How do we make the score go down?

Student Activity Worksheet: Keeping Score

What to do:	Using/Details:
Open your balloon project. For the blue balloon sprite, locate the code that runs when the balloon touches the pencil. Add code to increase the score by 1.	
Save and test your code to make sure the score works.	
The second balloon you added is something the player needs to avoid. If the player touches the second balloon, they lose a point. When the player touches this balloon, subtract 1 from the score.	 
Explore adding more balloons to pop or other obstacles.	

Extended Activity: Adding a Timer

What to do:	Using/Details:
Create a variable named ***Timer*** In your first balloon script, initialize the timer variable to 0 when the program starts, right after you initialized the score.	
We will use the ***wait 1 seconds*** command to count seconds going by. Add a forever loop after the command to initialize the Timer and inside the loop, add the ***wait 1 seconds*** command. This loop will run every 1 second.	
We want the player to see the seconds go by. Add the command ***change Timer by 1*** after the **wait** command.	
Once 20 seconds go by, we let the player know the game is over and we stop the program from running. To do this add these blocks after you change the timer	
Save and test your code to make sure it works.	