Writing Code to Simulate
 a Coin Toss

**Algebra**

**Unit 3 Line Master 10a**

Let’s explore probability through coding by creating a coin toss simulator application in Scratch.

Start with an existing Scratch application that has a **coin** sprite with two costumes ready to go: one for heads and one for tails. Click on this link.

<https://scratch.mit.edu/projects/692982103/editor>

If you have a Scratch login, save the project in your Scratch account by selecting Remix at the top of the screen. A login is not required to work with the code, but you will not be able to save your changes without it.

1. You can see this application already has a **coin** sprite. If you select the **Costumes** tab (the middle tab on the top left), you will see that the costume that has “H” for heads has **0** for its name and the other costume has a “T” for tails and has **1**
for its name.

 

Return to the code, by clicking on the **Code** tab.

 Writing Code to Simulate
 a Coin Toss (cont’d)

**Algebra**

**Unit 3 Line Master 10b**

2. Start writing the code for the coin toss.
 Under **Events**, select the **when green flag** **clicked** block
 and drag it to the code editing area as shown.

 

3. Create a variable to hold a random number between 0 and 1.
 Later in the code, if the random number is 0, you will display
 the coin in its heads costume and if the random number is 1,
 you will display the coin in its tails costume.

 To make a variable, under **Variables**, select **Make a Variable**.
 Name the variable **coinToss** and select **OK**.

 

 Writing Code to Simulate
 a Coin Toss (cont’d)

**Algebra**

**Unit 3 Line Master 10c**

4. Under **Variables**, select the **set** block and

 ensure **coinToss** is selected from the

 pulldown menu. Place it underneath the
 **when green flag clicked** block.

****5. Set the **coinToss** variable to a
 random number between 0 and 1.
 To do this, go to **Operators** and
 drag the **pick random** block into
 the **set coinToss** block as shown.
 Change the number range to **0** to **1**.

6. Now, switch the **coin** costume to display the H **coin** costume
 if the random number picked is 0, and T if the random number
 picked is 1. Under **Looks**, select the **switch costume to** block
 and drag it beneath the **set** block.
 Because the costumes have already been labelled as **0** and **1**
 in the **Costumes** tab, you just need to use the **coinToss** variable
 block from under **Variables** and place it into the **switch costume to**
 block, since the **coinToss** variable will either be set to 0 or 1.

 

 Writing Code to Simulate
 a Coin Toss (cont’d)

**Algebra**

**Unit 3 Line Master 10d**

7. You need a way to keep track of whether heads or tails
 is tossed. To do this, you need two additional variables.
 Under **Variables**, select **Make a Variable** and name
 it **totalHeads**. Make a second variable called **totalTails**.

***conditional statement*:** A type of coding instruction used to compare values and make decisions. A conditional statement tells a program to execute an action only if a certain condition is true or false. It is often represented as an if-then or if-then-else statement.

You will use a *conditional statement* to determine if you need to increase the **totalHeads** variable by 1 because heads is tossed,
or to increase the **totalTails** variable by 1 because tails is tossed. Follow Steps 8 to 11 to code the statement.

8. Under **Control**, select the
**if-then-else** block and place it
beneath the **switch costume to**
block.

9. Inside the **if-then-else** block,
 you need a comparison operator
 to check if something is true or not.
 Under **Operators**, select the **equal** (=) comparison operator and place

 it inside the first part of the **if-then-else**
 block.

 Writing Code to Simulate
 a Coin Toss (cont’d)

**Algebra**

**Unit 3 Line Master 10e**

****

10. You want the program to check if
 heads was tossed and, if it was,
 increase the **totalHeads** variable
 by 1 as a way of keeping track of the
 total number of heads tossed.

 Under **Variables**, select the **coinToss**
 block and place it inside the **equals (=)** operator.
 Change the second part of the **equals** (=) operator to **0**,
 rather than 50.

 Next, select a **change** block from
 **variables** and place it below thefirstportion of the **if-then-else** block.
 Ensure the **totalHeads** variable is
 selected from the pulldown menu and
 that the number **1** is displayed in the
 **change** block so that each time a head
 is tossed, the **totalHeads** variable will
 increase by 1.

11. If heads is not tossed, that means tails must have been tossed.
 Place a change block inside the else portion of the conditional.
 Ensure the totalTails variable is selected in the pulldown menu
 and that the number 1 is displayed in the change block so that
 each time tails is tossed, the totalTails variable will increase by 1.

 Writing Code to Simulate
 a Coin Toss (cont’d)

**Algebra**

**Unit 3 Line Master 10f**

12. Click on the green flag above the stage to execute the code.
 Do this many times. The values of three variables are displayed
 on the stage. The coin should switch to an H or a T depending
 on the random number that is picked. Each time an H appears,
 the **totalHeads** variable increases by 1. Each time a T appears,
 the **totalTails** variable increases by 1.
 If these things don’t happen, check your code to make sure
 it matches what is shown in the images at each step.

 Your stage should look similar to
 this but with different numbers
 of heads and tails.

13. Add a variable to keep track of the
 total number of tosses. Under **Variables**,
 select **Make a Variable** and call it
 **totalTosses**. Each time the green flag
 is clicked, you’d like the **totalTosses**
 variable to increase by 1. Insert a **change** block below first **set**
 block in the code and ensure the **totalTosses** variable is
 selected from the pulldown menu. To insert a block, drag it to
 where you want it to appear and let go.

 The first 4 blocks in your code should look like what is
 shown above.

 Writing Code to Simulate
 a Coin Toss (cont’d)

**Algebra**

**Unit 3 Line Master 10g**

14. Finally, add code to reset all of the variables to 0 when the
 user presses the spacebar. Under **Events**, select **when
 space key** **pressed**, and place it in the code editing window.

 Under **Variables,** select three **set** blocksand place them
 beneath the **when** **space key pressed** block.
 Select **totalTosses** from the pulldown menu for the first one, **totalHeads** for the second one, and **totalTails** for the third one.

 

15. Before you run your program again, make sure only certain
 variables are displayed on the stage.

 Under Variables, deselect the checkbox for coinToss.
 Only the three variables with the checkboxes selected
 will be displayed.

 Writing Code to Simulate
 a Coin Toss (cont’d)

**Algebra**

**Unit 3 Line Master 10h**

16. Execute the code several times and then press the space bar.
 Did the variable values reset back to 0?
 If not, look carefully through the code to make sure you have
 everything correct and haven’t missed a step.
 We call this process “debugging”.

***theoretical probability:*** when all outcomes are equally likely, the number of favourable outcomes for an event divided by the total number of outcomes

***experimental probability:*** the number of favourable outcomes in an experiment divided by the total number of trials

17. When we toss a fair coin many times, theoretical probability
 tells us that we can expect to get half heads and half tails.

 a) Reset the variables. Then click on the green flag 10 times.
 How many of the 10 tosses are heads? How many are tails?
 How close is this experimental result to what theoretical
 probability would lead you to expect?

 b) Combine your results with 4 other pairs. How many of the
 50 tosses are heads? How many are tails?

 c) Suppose you could toss a fair coin 10 000 times.
 What would theoretical probability lead you to expect
 might happen? What do you think might happen?