Learner Full Name: _____ **School:** _____

Workstation Device: _____ **Class:** _____

Session 17: Loops, Decisions & Debugging

In Session 16, you made a sprite walk, speak, and turn once. But real computer programs don't just execute 3 blocks and stop dead. They keep running in a continuous cycle, make decisions when obstacles appear, and keep score. Today, you become a genuine game developer!

1. Plain Talk: The Fufu Pounding Analogy (What is a Loop?)

Think about preparing *Fufu*: one person pounds the pestle into the wooden mortar, and the second person turns the cassava. *Pound, turn. Pound, turn. Pound, turn.*

You do not hire 50 different people to strike the mortar once each! One person repeats the exact same action rhythmically until the fufu is smooth. In computer science, an automatic repeating cycle is called a **Loop** (or **Iteration**).

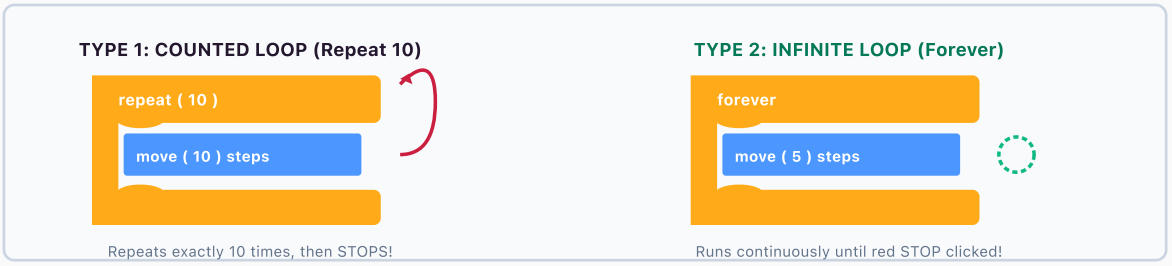


Figure 17.0: Loops in Scratch. Blocks placed inside the "C-mouth" repeat automatically!

2. Why Loops Save You Hours of Work

> 💡 **LAZY PROGRAMMERS ARE SMART PROGRAMMERS!**
> If you want your sprite to walk 100 paces, you *could* drag out 100 separate blue move 1 step blocks and stack them until your screen is cluttered and laggy.
> Or... you can simply drag **ONE** repeat 100 block and drop a single move 1 step inside it. You just saved 99 blocks of effort!

1. What is a "Condition"?

In life, you make decisions based on real-world conditions every day:

- *"**IF** it rains at 2:00 PM, **THEN** open my umbrella."*
- *"**IF** my stomach rumbles, **THEN** buy roasted plantain."*

A **Condition** is a true/false question. If the answer is **YES (True)**, the computer runs the special code tucked inside the block. If the answer is **NO (False)**, the computer ignores it completely!

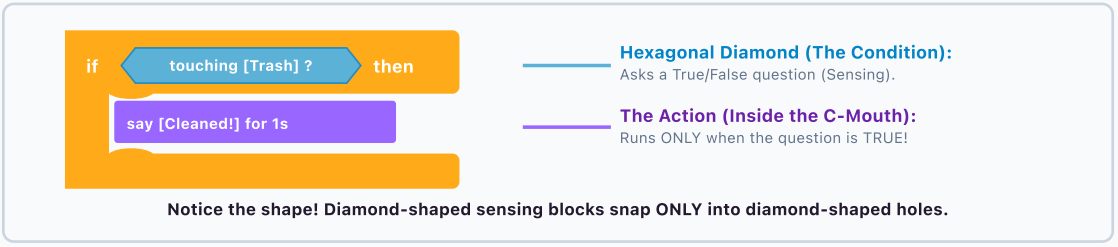


Figure 17.1: Anatomy of an IF-THEN block. The condition evaluates whether the action inside runs.

2. Popular Sensing Blocks (The Senses of Your Sprite)

Sensing Block	What It Checks For	Game Example
<touching [mouse-pointer]?>	Did the player hover over the sprite?	Button changes color when hovered.
<touching [edge]?>	Did the character hit the wall of the stage?	Bounces back or resets position.
<key [space] pressed?>	Is the player tapping the spacebar right now?	Character jumps over a puddle!

1. What is a "Variable"?

Think about a traditional clay pot or wooden calabash kept in a village kitchen. You can put 5 cowrie shells inside it. If you earn 2 more, you drop them in and it now holds 7. If you spend 3, you pull them out and it holds 4.

In computer programming, a **Variable** is a named digital container that holds a value that can **change (vary)** while the game is running!

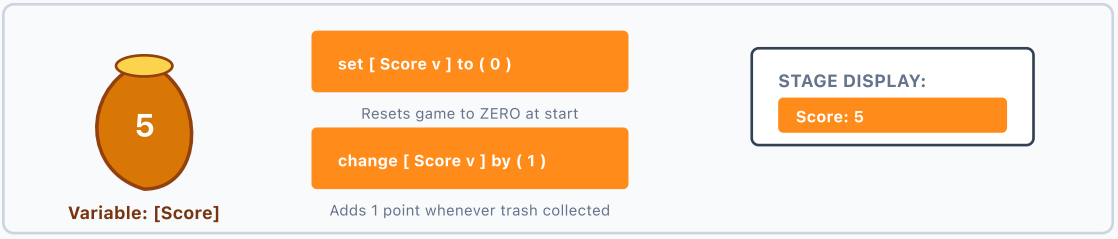


Figure 17.2: Variables store changing information like points, health, or countdown timers.

2. Common Variables in Games

🏆 Score / Points

Starts at 0. Increases by +1 or +10 each time the player achieves a goal or collects an item.

⏳ Timer / Countdown

Starts at 60 seconds. Decreases by -1 every second until it reaches 0 (Game Over!).

1. What is a "Bug"?

In 1947, computer pioneer **Grace Hopper** was working on a massive room-sized computer when it suddenly malfunctioned. When her team opened the machine, they found a **real moth** trapped inside one of the electrical relays! They taped the moth into the logbook with the note: *"First actual case of bug being found."*

Today, any mistake, logic flaw, or unexpected behavior in code is called a **Bug**. The detective process of finding and fixing that mistake is called **Debugging**.

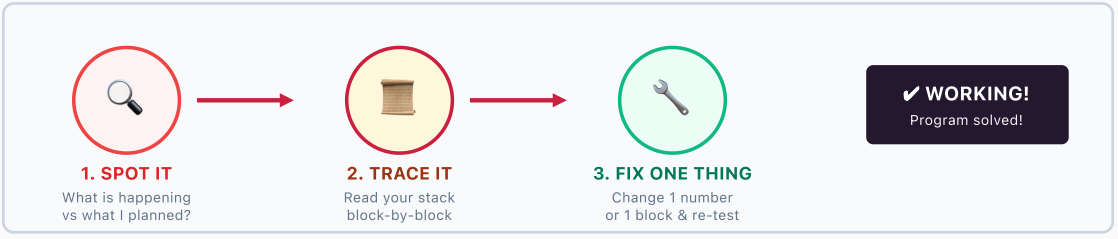


Figure 17.3: The 3-Step Debugging Protocol. Never change 5 things at once; change one block and test again!

2. The Cardinal Debugging Law

> 🧠 **BUGS ARE NOT FAILURES; THEY ARE CLUES!**
> Every great software engineer in the world writes bugs every single day. Having an error in your code does not mean you are bad at math or computers. It just means you gave the computer an incomplete instruction. Be a calm detective!

1. Lab Mission: Code a Clean-Up Mini-Game

Follow these 6 steps in Scratch to build a functional educational game teaching community sanitation:

1. **Make the Sachet & the Variable [Score]:**
   Choose a Sprite → Paint a sachet (or pick one). In the sprite pane, set **Direction → Left/Right** so it never flips upside down.
   Click the orange **Variables** dot → **"Make a Variable"** → Name it: Score → OK.

2. **Initialize the Game:**
   Drag when green flag clicked into the workspace.
   Snap set [Score v] to (0) underneath it so points reset every new game.

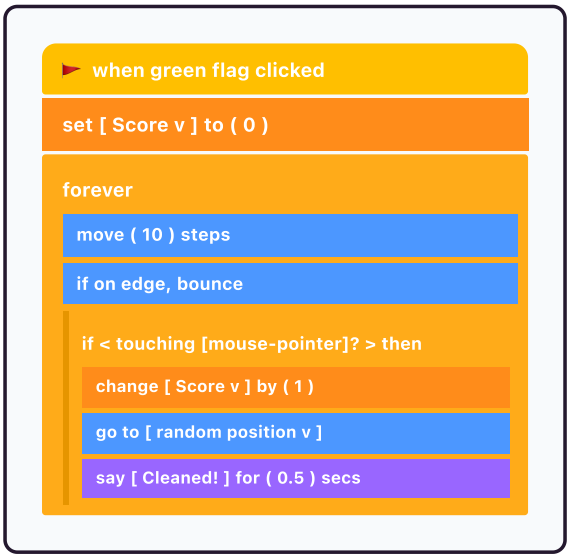
3. **Add the Movement Loop:**
   From **Control**, snap a forever loop under your reset block.
   Inside the loop, snap: move (10) steps → then if on edge, bounce. *Your sachet now patrols back and forth!*

4. **Add the Decision Condition:**
   Inside the forever loop (below the bounce block), drop an if <> then block.
   From **Sensing**, snap <touching [mouse-pointer]?> into the diamond hole.

5. **Award Points & Speak:**
   Inside the C-mouth of the if block, snap:
   • change [Score v] by (1)
   • go to [random position v] (from **Motion**: the sachet jumps away, so one touch = one point)
   • say [Cleaned!] for (0.5) secs

6. **Play, Debug & Save:**
   Click the **Green Flag**. Hover your mouse over the moving sprite!
   Watch: Each touch adds 1 to Score; the sachet jumps and says "Cleaned!"
   Save: **File → Save to your computer** → Name: Cleanup_Game_Ama.sb3.

2. Visual Target (The Complete Game Script)



1. "Be the Computer" Logic Tracing Challenge

Read the code snippet below carefully and predict what happens:

```
set [ Score ] to 10
repeat ( 3 ) {
  change [ Score ] by 5
}
```

What is the final number inside [Score] after the loop finishes?

---

2. Spot the Bug Challenge

Kwesi built a game where the player gets points by clicking a balloon. But when he clicks the green flag to start a second game, his Score starts at **45** instead of **0**!

What block did Kwesi forget to put at the top of his script?

---

3. Self-Assessment & Mentor Sign-Off

Tick the column that reflects your confidence at the end of the Programming Module:

Programming & Logic Competency	With Help	Almost Alone	All By Myself!
1. I can explain the difference between a `repeat` loop and a `forever` loop.	○	○	○
2. I can use an `if-then` condition with a diamond sensing block.	○	○	○
3. I can create a Variable (Score) and update it dynamically in code.	○	○	○
4. I know the 3-step debugging protocol to trace and fix errors calmly.	○	○	○

Mentor Signature: _____ **Date Completed:** _____