# **LESSON 4: ADVANCED EVENTS & INTERACTIVE SYSTEMS**

## **SLIDE 1: LESSON TITLE**

**Advanced Events & Interactive Systems**

Core Objectives:

* Understand different types of events beyond "start program"
* Create programs that respond to collision, landing, and free fall
* Build interactive systems that react to the physical world
* Recognize how devices around us use event-driven programming

## **SLIDE 2: LESSON INPUT**

**What are Advanced Events?**

Events are things that trigger programs to run

* "When start program" - you've used this!
* "When collision" - Sphero hits something
* "When landing" - Sphero stops falling
* "When free fall" - Sphero is dropped

**Real World Examples:**

* Phone responds when you touch the screen
* Car alarm triggers when someone hits the car
* Automatic doors open when you approach

**Your Challenge:** Create responses for collision, landing, and free fall events

## **SLIDE 3: CORE ACTIVITY**

**Building Interactive Response System**

**Core Challenge:** Master the three basic physical events

**Programming Focus:** One event = one simple response

**Requirements:**

* Program response to collision (LED + sound)
* Program response to landing (LED + sound)
* Program response to free fall (LED + sound)
* Each event has different color and sound

## **SLIDE 4: MINI PLENARY**

**Check Your Understanding**

1. What's the difference between "start program" and "collision" events?
2. Which event happens first: free fall or landing? Why?
3. How is this like a phone responding to touch?
4. What other events might be useful for interactive programs?

**Success Check:**

* All three events trigger appropriate responses
* You can safely test each event type
* You understand programs can "wait" for events to happen

## **SlIDE 5: Event Explorer (Easy)**

**Example:**

[On Collision] → Red LED + "crash" sound

[On Landing] → Green LED + "ding" sound

[On Free Fall] → Blue LED + "whoosh" sound

**Success Criteria:** All three events trigger clearly different responses

**Time Estimate:** 15-20 minutes

**Difficulty:** ⭐ (Basic event blocks only)

## **SLIDE 7: Interactive Art Installation (Medium)**

**Core Challenge:** Create artistic multi-element responses

**Programming Focus:** Each event triggers visual + audio + matrix animation

**Requirements:**

* Collision: LED effects + sound + matrix pattern
* Landing: Color transitions + musical tones + matrix animation
* Free fall: Rainbow effects + sound sequence + spinning matrix display
* Use timing blocks to create sequences

**Example:**

[On Collision] → Flash red 5x + alarm sound + lightning matrix + explosion animation

[On Landing] → Blue→green fade + chime + heart pattern + gentle pulse

[On Free Fall] → Rainbow cycle + spiral animation + continuous until landing

**Success Criteria:** Rich, artistic responses that feel like interactive art

**Time Estimate:** 25-35 minutes

**Difficulty:** ⭐⭐ (Multiple effects, timing, matrix animations)

## **RED CARD: Multi-Event Storyteller (Hard)**

**Core Challenge:** Create a story that unfolds through event sequences **Programming Focus:** Complex combinations + longer sequences + creative narrative

**Requirements:**

* Create a complete story using all three events in sequence
* Each event advances the story with dialogue (scrolling text on matrix)
* Use longer, more complex animation sequences
* Include "story reset" when sequence completes
* Add background "idle" behaviors between story events

**Example Story: "Space Adventure"**

[Story Start - Idle State]

├── Slow blue pulse (waiting for adventure)

├── Display "READY" scrolling text

[Event 1: Free Fall = "Launch"]

├── Rainbow rocket launch colors + ascending musical scale

├── Display "LAUNCHING!" scrolling text + rocket animation

├── Play countdown "3-2-1-BLAST OFF!"

[Event 2: Landing = "Planet Landing"]

├── Green success colors + victory fanfare

├── Display "PLANET FOUND!" + planet animation

├── Play alien greeting sounds

[Event 3: Collision = "Alien Encounter"]

├── Red/yellow flashing + suspense music

├── Display "ALIENS DETECTED!" + alien animation

├── End with "MISSION COMPLETE!" + return to idle

## **SLIDE 9: PLENARY**

**Prove Your Event Mastery**

Assessment Questions:

1. How do events make programs more interactive?
2. What's the advantage of event-driven programming?
3. Name three devices that use event responses in real life
4. Why don't we need loops for event programs?

**Prove It Task - Choose One:**

* Demonstrate all your events and explain what triggers each
* Design a new event response for a different situation
* Test someone else's program and identify which events they used
* Explain how event programming could solve a real problem

**Exit Ticket:**

* Which event was most interesting to program?
* One real-world device that uses events like these
* One new event type you'd like to try