Sphero Programming Unit - Year 6 Overview

Unit Structure

6 lessons building programming concepts systematically Core + Extension model: All students complete core objective, then access challenge cards

---

Lesson 1: Sequential Programming

Programming Concepts: Events, Sequences, Basic I/O

Core: Create square using 8 movement blocks **Extensions:**

- Add LED lights to corners
- Create triangle with angles
- Design pentagon/hexagon

Lesson 2: Loops & Efficiency

Programming Concepts: Loops, Repetition, Code Efficiency

Core: Rebuild square with loops + draw spiral pattern **Extensions:**

- Create star patterns
- Add animations to movement
- Design custom mathematical patterns

Lesson 3: Advanced Events

Programming Concepts: Event-Driven Programming, Multiple Event Handlers

Core: Respond to collision, landing, and free fall events **Extensions:**

- Add spinning detection
- Create interactive art installation
- Include system state awareness

Lesson 4: Conditional Logic

Programming Concepts: If/Else, Boolean Logic, Sensor Data

Core: Motion detector (shake = red, still = green) **Extensions:**

- Add sound feedback
- Create 3-level detection system
- Build motion-based games

Lesson 5: Variables & Data

Programming Concepts: Variables, Data Storage, Counters

Core: Create scoring system with event-triggered counter **Extensions:**

- Build timer systems
- Track multiple variables
- Create smart scoring with conditions

Lesson 6: Integration & Games

Programming Concepts: System Integration, Game Logic

Core: Complete game using events, conditionals, loops, variables **Extensions:**

- Multi-level difficulty games
- Two-player competition systems
- Adaptive games that change based on performance

Key Features

- **Mixed ability support:** Core ensures all succeed, extensions prevent boredom
- **Real assessment:** Teacher verification for core, peer verification for extensions
- **Hands-on learning:** Physical robots with art creation and game development
- **Concept transfer:** Focus on programming logic that applies across languages