**Sphero Programming Curriculum: Years 3-6**

* **Year 3: Foundation Programming Concepts** - Sequential instructions, making code more efficient, debugging, and using loops.
* **Year 4: Conditional Logic and Interaction** - 'If/then/else' rules, using sensor data, generating random outcomes, and layered rules.
* **Year 5: Event-Driven Programming** - Triggering events, managing concurrent processes, using Boolean logic, and real-time response systems.
* **Year 6: Advanced Programming and Real-World Applications** - Algorithm optimisation, sensor integration, data management, system design, and project management.

## 

## 

## 

# **Sphero Programming Curriculum: Years 3-6**

## **Year 3: Foundation Programming**

**Focus:** Sequential programming and basic efficiency

Students learn to create command sequences that direct the Sphero robot through specific tasks. They develop skills in writing clear, efficient code and learn systematic debugging techniques when programs don't work as expected. The introduction of loops allows students to create more sophisticated patterns and reduces code repetition.

**Key concepts:**

* Sequential command programming
* Loop structures for repetition
* Debugging methodology
* Code optimization basics

**Example projects:** Programming navigation routes, creating geometric patterns, designing simple animations

## **Year 4: Conditional Logic and Sensors**

**Focus:** Decision-making and input processing

Students learn conditional programming that allows the robot to respond differently based on various inputs. They work with the Sphero's built-in sensors and incorporate random elements to create more dynamic and engaging programs.

**Key concepts:**

* If/then conditional statements
* Sensor integration (collision, accelerometer)
* Random number generation
* Multi-outcome logic structures

**Example projects:** Interactive games, responsive behaviors, sensor-triggered actions

## **Year 5: Event-Driven Systems**

**Focus:** Responsive programming and complex logic

Students develop event-driven programs that respond to specific triggers rather than running continuously. They learn to manage multiple simultaneous events and use Boolean logic to create sophisticated conditional structures.

**Key concepts:**

* Event handling and triggers
* Multi-event coordination
* Boolean operators (AND, OR, NOT)
* Real-time system design

**Example projects:** Interactive installations, multi-sensor systems, reaction-based games

## **Year 6: Advanced Programming and Integration**

**Focus:** Optimization and real-world applications

Students tackle advanced programming concepts including algorithm optimization, multi-sensor data fusion, and comprehensive data management. They complete substantial projects that integrate all previously learned skills.

**Key concepts:**

* Algorithm efficiency and optimization
* Multi-sensor data integration
* Data collection and analysis
* Large-scale project development

**Example projects:** Autonomous navigation systems, environmental monitoring, complex interactive applications

# **Programming Curriculum by Year Level**

| **Concept** | **Year 3** | **Year 4** | **Year 5** | **Year 6** |
| --- | --- | --- | --- | --- |
| Sequence | Use simple command sequences | Combine sequences with if conditions | Trigger sequences using events | Optimise sequences for speed and clarity |
| Loop | Use loop to repeat actions | Use loop with if conditions | Use loop forever in event-driven programs | Use nested loops and optimise loop logic |
| If Statement | Not introduced | Use basic if statements | Use if else for multiple outcomes | Use complex conditional logic across systems |
| Event | Not introduced | Not introduced | Use sensor-based triggers as simple events | Use multiple events with priority handling |
| Variable | Not introduced | Use simple variables (e.g., counters) | Store sensor or random values in variables | Use variables for data logging and decision-making |
| Sensor | Not introduced | Use basic sensors (collision, shake) | Use sensors to trigger if statements | Combine sensors with Boolean logic |

## **Advanced Concepts Simplified Explanations**

### **Optimise sequences for speed and clarity**

Write code that runs as fast as possible and is easy for people to read and understand.

### **Use nested loops and optimise loop logic**

Use loops inside of other loops to do more complex tasks (like drawing a checkerboard) and make sure they run as efficiently as possible.

### **Use complex conditional logic across systems**

Create complicated rules with multiple conditions for how the robot should behave.

### **Design real-time systems with event queues**

Build a program that can respond instantly to things as they happen, like a video game or a reaction timer,

**Use variables for data logging and decision-making**

Use "storage boxes" (variables) to save information from the robot (like how far it has traveled) to help it make smart choices later.

### **Integrate multiple sensors for smart decision-making**

Use information from more than one sensor at a time (like light and accelerometer sensors) to make the robot's decisions more accurate and reliable.