

**Flight Events**  
**Grades 4-8**  
**Designed by Robolink**

**Summary:**

Students will be able to fly their drones! However, they will need to understand how a drone is able to fly in order to use the right commands in their programming. This lesson will go over takeoff and landing.

**Guiding Question(s):**

How is a drone able to fly and stay in the air?

How will each flight command help you control your drone?

**Learning Objectives:**

Students will be able to:

- Explain how a drone takes off, flies, and lands
- Create and run a program in Blockly so a drone can take off, hover, and land
- Create and run a program in Blockly that has a drone do pushups within a time constraint
- Engineer a solution to lifting as large of a payload as possible

**Materials needed:**

- CoDrones with batteries (1 drone for 2 students, or 1:1 if there are enough drones)
- Technology for showing online videos with sound and shared screens
- Laptops or desktop computers with internet access
- Engineering journal or worksheets

**Lesson Title:** Flight Events

**Time:** 1 hour 30 minutes

**Concepts:**

- Interpreting visual, oral, and text information
- Following multi-step and complex directions
- Problem-solving
- Using appropriate tools strategically
- Constructing programs

**Engagement: (Introduction)**

- CoDrones are quadcopters, which are based off of (much larger) helicopters. Demonstrate the CoDrone taking off and landing, and then have students read the article below. How are a helicopter, and the CoDrone, able to take off, hover, and land? Have students explore Blockly together and ask what menus and blocks they think they will need to have their CoDrone perform these tasks.

**Article:** [What Is a Helicopter?](#)

### **Exploration: (Activity)**

- Have students complete [Flight Events](#) in Robolink Basecamp. Students might need to increase wait time to fully see the drone hovering. Hold a small competition to see which student can get their CoDrone to do the most drone pushups in 10 seconds!

### **Explanation: (Recap)**

- Ask students to use pseudocode to explain to a partner how they got their drone to take off, hover, and land in Blockly using both their own words and the appropriate academic language.

### **NGSS Extension**

- Ask the top three winners of the payload challenge to demonstrate their engineering. Ask the class to think of what in their design allowed their CoDrones to lift heavier payloads. How would they redesign their engineering to lift a heavier payload?

### **Elaboration: (Extension)**

- Have students find different ways to modify their CoDrone pushup program. This will most likely be just changing the delay time.

### **NGSS Extension**

- Show students the video below about drone delivery in the Australian outback. Afterwards, ask for other situations when drones would need to make deliveries. What problems could arise?

**Video:** [Project Wing tests real-world drone deliveries in Australia](#)

- Explain the payload challenge to students: they need to find a way to have their CoDrones lift a payload when they run the takeoff and landing code. Supply materials like sewing thread, magnets, tape, Legos, pipe cleaners, and anything else that might be able to be used without damaging the drone. When students are finished engineering, they can attach their payload to their drone, take off, and land. Have them add more to their payload until they are no longer able. Weigh each of the payloads. Whoever has the highest wins!

**Evaluation:**

- In a journal or on a worksheet have students answer the following questions. Examples of engineering portfolios are available on [Google Sites](#) and [ProjectBoard](#).
1. Scientifically, how is your CoDrone able to take off?
  2. Explain your flight event program using pseudocode. Why did you choose to make your program the way you did?
  3. How did changing your CoDrone pushup code affect its flight?
  4. Did you have any problems running your code or using your drone? If so, what did you do to fix them?
  5. What did you learn?
  6. (only if payload challenge was completed) Explain the engineering behind the payload challenge. Why did you choose the materials and design that you did? How would you redesign your engineering in order to lift a heavier payload?

**Related Vocabulary:** air pressure, airfoil, block, Blockly, code, flight event, hover, landing, lift, menu, program, propeller, pseudocode, rotor, takeoff, value

**Standards:****CCSS:**

[ELA-LITERACY.RI.4.7](#): Interpret information presented visually, orally, or quantitatively (e.g., in charts, graphs, diagrams, time lines, animations, or interactive elements on Web pages) and explain how the information contributes to an understanding of the text in which it appears.

[ELA-LITERACY.RI.5.7](#): Draw on information from multiple print or digital sources, demonstrating the ability to locate an answer to a question quickly or to solve a problem efficiently.

[ELA-LITERACY.RI.6.7](#): Integrate information presented in different media or formats (e.g., visually, quantitatively) as well as in words to develop a coherent understanding of a topic or issue.

[ELA-LITERACY.RST.6-8.3](#): Follow precisely a multistep procedure when carrying out experiments, taking measurements, or performing technical tasks.

[ELA-LITERACY.RST.6-8.7](#): Integrate quantitative or technical information expressed in words in a text with a version of that information expressed visually (e.g., in a flowchart, diagram, model, graph, or table).

[ELA-LITERACY.RST.6-8.9](#): Compare and contrast the information gained from

experiments, simulations, video, or multimedia sources with that gained from reading a text on the same topic.

[MATH.PRACTICE.MP5](#): Use appropriate tools strategically.

[MATH.PRACTICE.MP7](#): Look for and make use of structure.

### **NGSS:**

[3-5-ETS1-1](#): Define a simple design problem reflecting a need or a want that includes specified criteria for success and constraints on materials, time, or cost.

[MS-ETS1-1](#): Define the criteria and constraints of a design problem with sufficient precision to ensure a successful solution, taking into account relevant scientific principles and potential impacts on people and the natural environment that may limit possible solutions.

### **CSTA:**

[1B-CS-03](#): Determine potential solutions to solve simple hardware and software problems using common troubleshooting strategies.

[1B-AP-15](#): Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.

[1B-AP-17](#): Describe choices made during program development using code comments, presentations, and demonstrations.

[2-CS-03](#): Systematically identify and fix problems with computing devices and their components.

### **ISTE:**

[5D](#): Students understand how automation works and use algorithmic thinking to develop a sequence of steps to create and test automated solutions.

[6A](#): Students choose the appropriate platforms and tools for meeting the desired objectives of their creation or communication.