Flight Movements: Throttle and Yaw

Grades 6-12

Designed by Robolink

Summary:

This can also be called the flight basics! Students will be learning the flight directions (throttle, pitch, yaw, and roll) and how to program them in Python. They will then use what they learned to have their drone fly to moving landing pads and simulate drone delivery paths.

Guiding Question(s):

- How do flying objects move in a 3D space?
- How does a program communicate with a robot?
- How can you program your CoDrone EDU to fly in multiple directions?

Learning Objectives:

Students will be able to:

- Program a friend through an obstacle course using the flight directions.
- Write programs to have their CoDrone EDU fly in multiple directions.

Materials needed:

- CoDrone EDU, remote, and USB cable for each student or student group
- Laptop with Internet access and PyCharmEDU installed for each student or student group
- Charged CoDrone EDU batteries and extra chargers
- Obstacle course materials. Desks and chairs work well for this!
- Blindfold or eye mask
- Landing pads

Lesson Title: Flight Movements **Time:** 1 hour 30 minutes

Concepts:

- Interpreting visual, oral, and text information
- Following multistep and complex directions
- Problem-solving
- Using appropriate tools strategically

Engagement: (Introduction)

- Before class: show flight directions from Flight Movements on the board and arrange one part of the classroom into a simple obstacle course with tables, desks, and chairs.
- Ask students how they think robots are able to receive their instructions on what to do. (Answer: through programs)
- Explain to students that they are going to program their friends to go through an obstacle course, but they can only use drone pilot language! Show the flight directions and go over all of them as a class. They can also go through it physically (like sliding to the left for negative roll).
- Split students up into pairs and explain the Blind Drone activity in the tutorial. The drone will need to start sitting down in a chair and move through the obstacle course to sit down in a different chair. The drone has to have their eyes covered and can only follow the directions of the other person, the programmer, in their pair. Just like a real drone, they cannot talk or ask any questions! Multiple obstacle courses can be set up in the room, but if there is room for only one, another pair can switch the obstacle course around for other pairs.
- If there is time, have students in the same pair switch roles!
- After students have finished, ask them what challenges they faced. How were they able to overcome them? What worked well for their team? How does all of this relate to CoDrone EDU programming?

Exploration: (Activity)

- Have students complete lesson Flight Movements on Robolink Basecamp, including the Puddle Jumper challenge at the end!

Explanation: (Recap)

- Have students verbally share how they programmed their CoDrone EDU to fly with the Puddle Jumper challenge. What difficulties did they face, and how did they solve them? What creative solutions did they come up with?

Elaboration: (Extension)

- Students discussed energy efficiency with drone deliveries in the Flight Events lesson, and in this lesson, they will be simulating it. Show students the video below and then ask for questions, thoughts, and observations. How does a normal drone delivery work?
  Video: Watch Amazon's latest drone demo
- Set up obstacles or landing pads to simulate delivery drop-off sites as well as a

warehouse, and then give students their challenge: they will need to fly their drone out to a site, land, and then come back to the warehouse. At least three sites should be set up, and one (or more) should combine two flight directions. For example, one site could be diagonal from the warehouse.

Extras:

- Set a timer so students can see if they can complete all deliveries in a given amount of time
- Put up obstacles to simulate anything that could get in a drone's way, like tall buildings, statues, and other low-flying aircraft.

Other challenges:

- The Floor is Lava: use a mat with multiple colors, and assign one color to each student. Students should take off and land on all spaces with the color that they were assigned.
- Obstacle course: set up tables, chairs, gates, and hoops for students to fly around and through.
- Assign a shape or pattern to each student and have them program their drone to fly in that shape.
- Have students program their drone to fly in their initials.

Evaluation:

- Explain the concept of pseudocode: it's using plain English to describe a program while still using its syntax. A definition and example are available on the GCSE Computer Science website.
- Have students answer the following questions in their engineering journals. A Robolink example is available on Google Sites:
    1. How did programming your friend to go through the obstacle course help you program your drone throughout this lesson?
    2. Include the pseudocode for the collection challenge. This can be either through words or a flow chart.
    3. What challenges would a drone face when making a delivery? What challenges did you face in the drone delivery challenge?
    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?

Related Vocabulary: drone, function, landing, library, movement command, negative, pitch, positive, Python, roll, takeoff, throttle, yaw

Standards:

CCSS:

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.RI.7.7: Compare and contrast a text to an audio, video, or multimedia version of the text, analyzing each medium's portrayal of the subject (e.g., how the delivery of a speech affects the impact of the words).
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.9-10.3: Follow precisely a complex multistep procedure when carrying out experiments, taking measurements, or performing technical tasks, attending to special cases or exceptions defined in the text.
ELA-LITERACY.RST.9-10.7: Translate quantitative or technical information expressed in words in a text into visual form (e.g., a table or chart) and translate information expressed visually or mathematically (e.g., in an equation) into words.
ELA-LITERACY.RST.11-12.3: Follow precisely a complex multistep procedure when carrying out experiments, taking measurements, or performing technical tasks; analyze the specific results based on explanations in the text.
ELA-LITERACY.RST.11-12.7: Integrate and evaluate multiple sources of information presented in diverse formats and media (e.g., quantitative data, video, multimedia) in order to address a question or solve a problem.
MATH.PRACTICE.MP1: Make sense of problems and persevere in solving them.
MATH.PRACTICE.MP5: Use appropriate tools strategically.

CSTA:

2-CS-03: Systematically identify and fix problems with computing devices and their components.
2-AP-16: Incorporate existing code, media, and libraries into original programs, and give attribution.
2-AP-19: Document programs in order to make them easier to follow, test, and debug.
3B-AP-16: Demonstrate code reuse by creating programming solutions using libraries and APIs.

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.