EXPANDING SQUARE SEARCH

LESSON PLAN | VERSION 2

LESSON OVERVIEW

Prerequisite Knowledge
- Build Essentials
- Fly Essentials
- Code Essentials

Materials Needed
- Hopper(s)
- safety glasses
- FTW CODE device(s) with Bluetooth capabilities (such as iPads or laptops)
- tape (for the floors)
- measuring tape (up to 20')
- landing pads
- writing utensils
- Wifi capable device(s)
  - **extension only** (such as iPads or laptops)

Time Allotment
Lesson: 1 hour (or 1 – 2 class periods), Setup: 20 minutes

Documents
- Search & Rescue Slide Deck II
- Search & Rescue Student Workbook

Vocabulary
- Expanding Square Search Pattern – an aerial search pattern that starts where the suspected location of an individual is and expands in a square pattern from there, useful when the general location of a lost individual is known

In this Lesson...
Students learn about and discuss how drones are used in search and rescue (SAR) operations. Then, they code Hopper to fly in an expanding square search pattern to simulate an aerial search where the general location of a lost individual is known.

Learning Objectives
- Participate in a group discussion about the use of drones in SAR operations.
- Accurately code Hopper to fly in an expanding square search pattern.
- Use the Engineering Design Process (EDP) and STEM practices to redesign Hopper's code as needed.

LESSON STRUCTURE

Read through the following table before starting the lesson. Approximate times have been given for each section to help with scheduling and time management.

Lesson Section	Description	Approximate Time
Direct Teaching	Open the slide deck titled Search & Rescue Slide Deck II and have the first slide up as the students walk in. Encourage students to think about the bell ringer questions: "How can drones be used in search and rescue operations? What kinds of technologies could be used?" Go through the rest of the slides of the slide deck with the students. Play any videos directly from the slides if possible (as opposed to going to the external website). Reference any presenter's notes as needed for each slide. The last slide presents the scenario of the Expanding Square Search activity to the students.	15 minutes
Discussion & Activity	Ensure the activity is set up prior to the beginning of the lesson. Allow for up to 20 minutes to set up. Separate students into small teams. Choose team sizes based on how many students there are and how many drones are available. Ideally, there would be no more than 3 – 4 students per team. Encourage the use of the steps of the Engineering Design Process, and computer programming terms such as algorithm, command, and bug as students write code. Implement the extension if time permits. Use the questions provided on page 7 to lead a group discussion with the students. Have them fill out a row in their flight log in their Search & Rescue Student Workbook. Sample codes can be found on pages 8 & 9.	45 minutes

ACTIVITY SCENARIO

An elderly man with dementia left his home and did not return. Search and rescue responders will use Hopper to conduct an aerial search of the area around his home.

You will code Hopper to fly in an expanding square search pattern starting with initial sides starting at 5 feet apart. The start of the pattern will be at the man's home since he cannot walk very far.

BUILD|FLY|CODE ®

FOR THE WIN
R O B O T I C S

ACTIVITY SETUP

Tape a 20' × 20' square on the ground which represents the search area. Tape a 5' × 5' grid in this square. Place two landing pads for Hopper that represent the starting and stopping points at the center and the lower left corner.

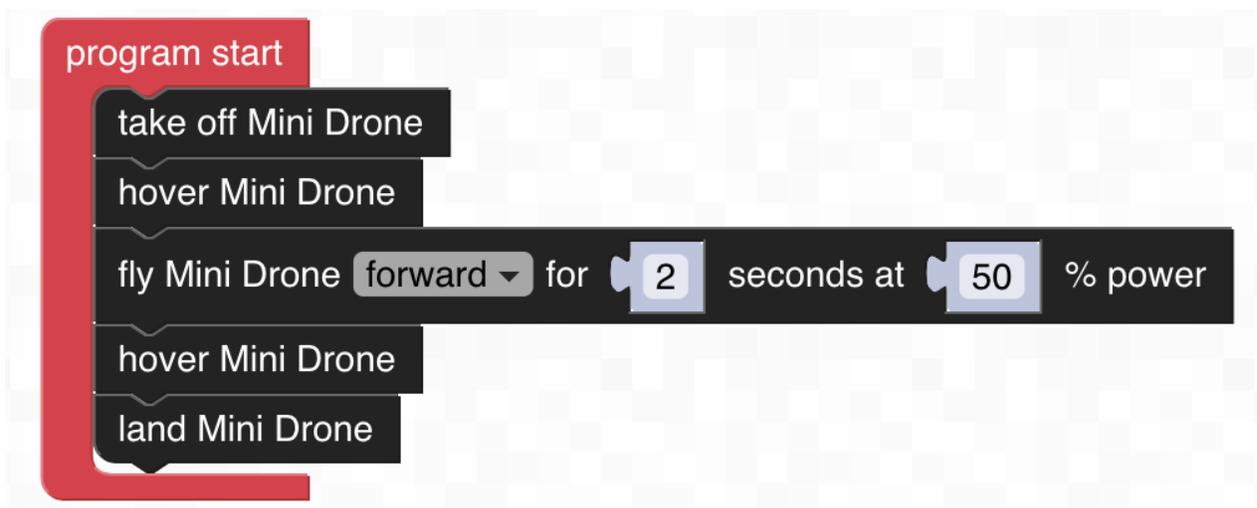An example of the setup is shown below.
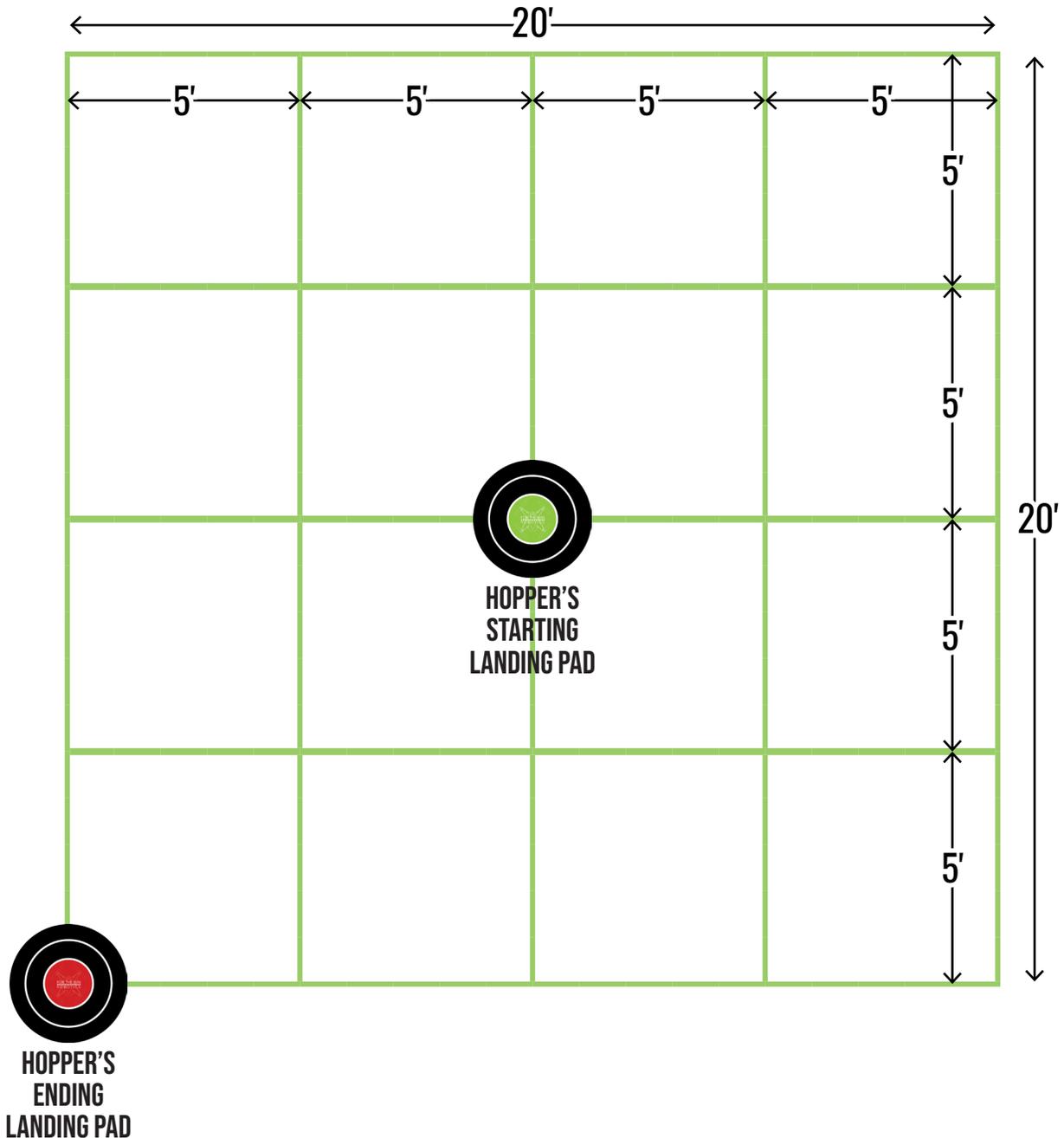
ACTIVITY IMPLEMENTATION

If the groups and Hoppers are staying the same as in the Parallel Track Search activity, finding the speed again is not necessary and this part of the activity can be skipped.

Have each team find Hopper's approximate speed when coded to fly at a certain power percentage and for a certain number of seconds. It is recommended to stay at 50% power or below.

A team's power percentage should stay *roughly* the same throughout this activity.

To stabilize Hopper after takeoff and before landing, it is recommended to command Hopper to hover. An example of a code students could write is shown below.

```
program start
    take off Mini Drone
    hover Mini Drone
    fly Mini Drone [forward ▼] for ( 2 ) seconds at ( 50 ) % power
    hover Mini Drone
    land Mini Drone
```

The takeoff and landing spots of Hopper should be measured.

Then, have each team use the formula rate = $\dfrac{\text{distance}}{\text{time}}$ to find the rate (speed) in feet per second of Hopper at the power percentage they chose.

Review with students that the formula for finding the rate is derived from the well-known formula:

$$\text{distance} = \text{rate} \times \text{time}$$
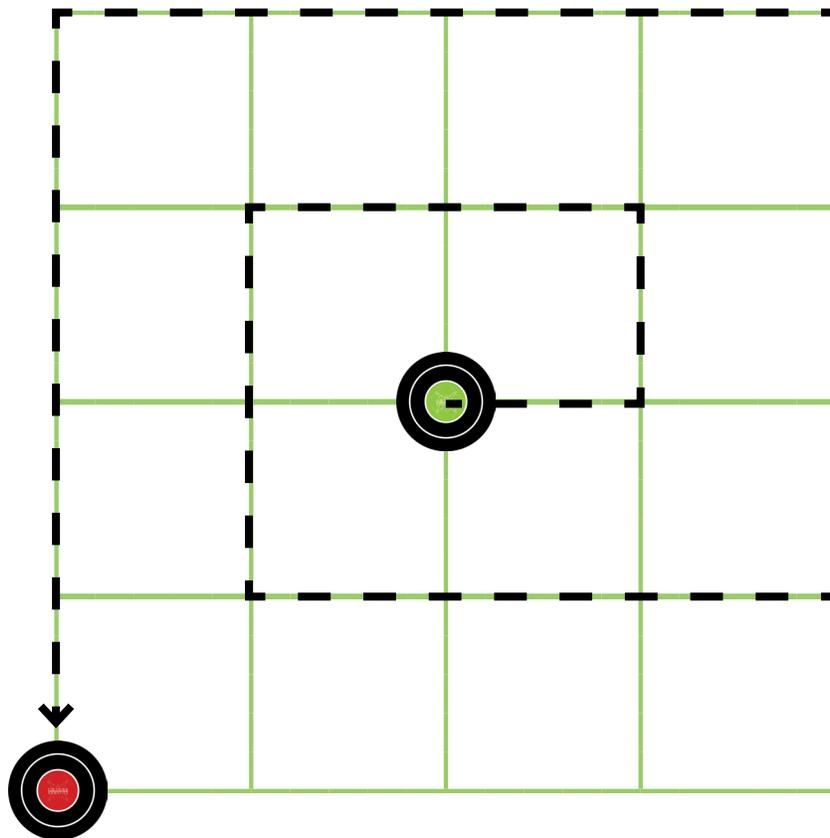
ACTIVITY IMPLEMENTATION

Activity Facilitation

Go through the following steps with the students to facilitate the activity.

1. Place Hopper on Hopper's starting landing pad. Each team can decide the direction that Hopper's eyes face depending on how they write their code.

2. Have each team code Hopper to fly in an expanding square pattern with initial sides starting at 5 feet apart. They should use the taped 5' × 5' grid as a guide when writing and testing their code. From there, each side of the square should increase by 5 feet.

   Encourage students to draw and label where they want Hopper to go, and to write down what they want Hopper to do in words before coding as needed. They can keep the answers to any calculations they do in exact form for coding. They can use the operation command in the Math tab for improper fractions, or they can convert to decimals.

3. If a team was not successful in the accuracy of coding Hopper, have them adjust their code and try again. If Hopper ever flies too far outside of the 20' × 20' square, the student should click on the red Emergency Land button.

FOR THE WIN
ROBOTICS

ACTIVITY IMPLEMENTATION

Extension

If time permits, challenge the students to simulate finding a lost individual using their code and Hopper's camera. Place another landing pad in the search area while the team has their back turned to the search area and have them identify the location of the landing pad while viewing Hopper's camera feed while they run their code.

Post-Activity Discussion Questions

Use the following questions to lead a group discussion after implementing the activity.

1. Was your initial calculation of Hopper's speed accurate? Or did you have to adjust it while coding the scenario?

2. Did you keep your calculated values as simplified, improper fractions? Why or why not?

3. Did you write down or draw your code before creating it in FTW CODE? If so, what did you create and how was it helpful?

4. Compare the code from your group to the codes that other groups wrote. Are they different? If so, how?

5. After comparing codes, would you make any changes to yours? If so, how would you make improvements to your code to make it more efficient?

Flight Log

Have students fill out a row in their flight log in their Searh & Rescue Student Workbook. An example of what it could look like is shown below.

Date	Drone Model	Location	Flight Time	Notes
04/02/2025	Hopper	Redwood Middle School Auditorium	20 minutes	My partner Mary and I coded Hopper to fly in an expanding square search pattern to simulate locating a missing man.

CODING EXAMPLE

Sample Code 1

```
program start
    take off Mini Drone
    hover Mini Drone
    fly Mini Drone [right ▼] for ( 10 [÷ ▼] 7 ) seconds at ( 50 ) % power
    hover Mini Drone
    fly Mini Drone [forward ▼] for ( 10 [÷ ▼] 7 ) seconds at ( 50 ) % power
    hover Mini Drone
    fly Mini Drone [left ▼] for ( 20 [÷ ▼] 7 ) seconds at ( 50 ) % power
    hover Mini Drone
    fly Mini Drone [backward ▼] for ( 20 [÷ ▼] 7 ) seconds at ( 50 ) % power
    hover Mini Drone
    fly Mini Drone [right ▼] for ( 30 [÷ ▼] 7 ) seconds at ( 50 ) % power
    hover Mini Drone
    fly Mini Drone [forward ▼] for ( 30 [÷ ▼] 7 ) seconds at ( 50 ) % power
    hover Mini Drone
    fly Mini Drone [left ▼] for ( 40 [÷ ▼] 7 ) seconds at ( 50 ) % power
    hover Mini Drone
    fly Mini Drone [backward ▼] for ( 40 [÷ ▼] 7 ) seconds at ( 50 ) % power
    hover Mini Drone
    land Mini Drone
```

BUILD|FLY|CODE ®

FOR THE WIN
ROBOTICS

CODING EXAMPLE

Sample Code 2

```
program start
    take off Mini Drone
    hover Mini Drone
    fly Mini Drone forward ▼ for [ 10 ÷ ▼ 7 ] seconds at [ 50 ] % power
    rotate Mini Drone [ 90 ] degrees counterclockwise ▼
    fly Mini Drone forward ▼ for [ 10 ÷ ▼ 7 ] seconds at [ 50 ] % power
    rotate Mini Drone [ 90 ] degrees counterclockwise ▼
    fly Mini Drone forward ▼ for [ 20 ÷ ▼ 7 ] seconds at [ 50 ] % power
    rotate Mini Drone [ 90 ] degrees counterclockwise ▼
    fly Mini Drone forward ▼ for [ 20 ÷ ▼ 7 ] seconds at [ 50 ] % power
    rotate Mini Drone [ 90 ] degrees counterclockwise ▼
    fly Mini Drone forward ▼ for [ 30 ÷ ▼ 7 ] seconds at [ 50 ] % power
    rotate Mini Drone [ 90 ] degrees counterclockwise ▼
    fly Mini Drone forward ▼ for [ 30 ÷ ▼ 7 ] seconds at [ 50 ] % power
    rotate Mini Drone [ 90 ] degrees counterclockwise ▼
    fly Mini Drone forward ▼ for [ 40 ÷ ▼ 7 ] seconds at [ 50 ] % power
    rotate Mini Drone [ 90 ] degrees counterclockwise ▼
    fly Mini Drone forward ▼ for [ 40 ÷ ▼ 7 ] seconds at [ 50 ] % power
    hover Mini Drone
    land Mini Drone
```

BUILD|FLY|CODE ®   STOP   **9**   STOP

FOR THE WIN
ROBOTICS