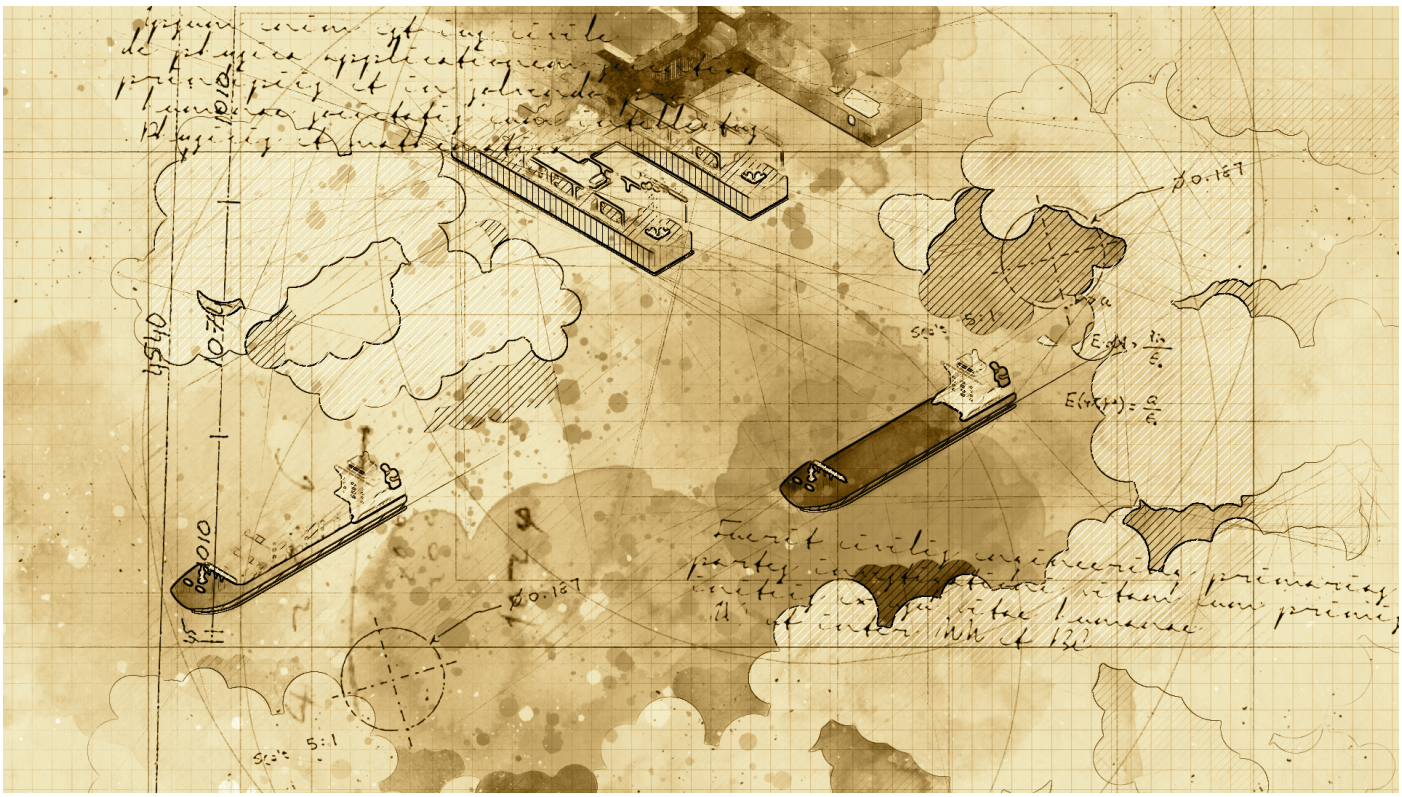
DRONE
LOGISTICS SERVICE

Drone Logistics Services Learning

Math Editions

Lesson 6: The Inside That Counts

Problem Solving | Geometry | Angles | Coding with Functions



"This storm was one for the history books. Cargo collection would have been completely halted if not for the great DroneBlocks coder of the early 2020s.

The DroneBlocks coder used clever geometry to map out a new flight path for the drones to collect cargo from ships dodging the storm clouds. It was amazing to see."

– Epic Drone Flights History Books

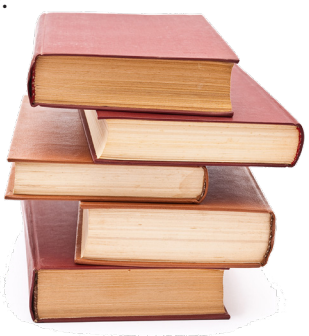
Instructions:

Turn the code from the previous lesson into functions. Make a rightForward and backwardLeft function which allows the drone to fly in the L-Shape either moving left or right.

Learn about interior angles and play the missing degree minigame.

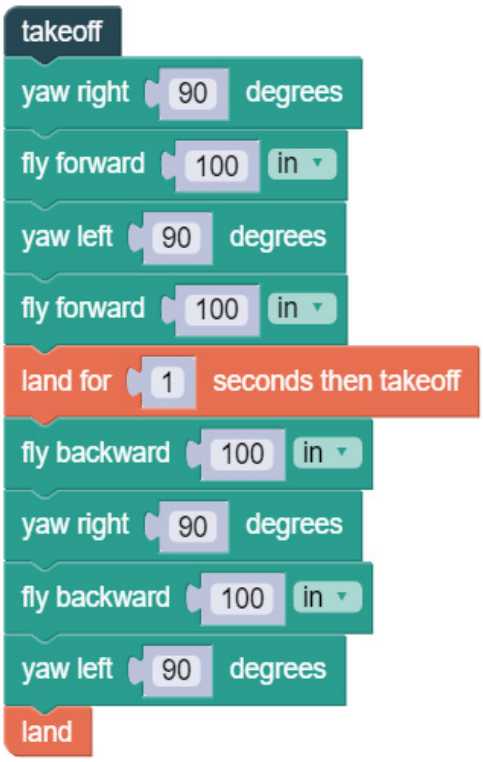
Learn about the rhombus and build a rhombus function.

Go down in history for your epic coding skills.



DRONE LOGISTICS SERVICE

Tool Time:



In the previous lesson we ended the lesson with this code. Very cool.

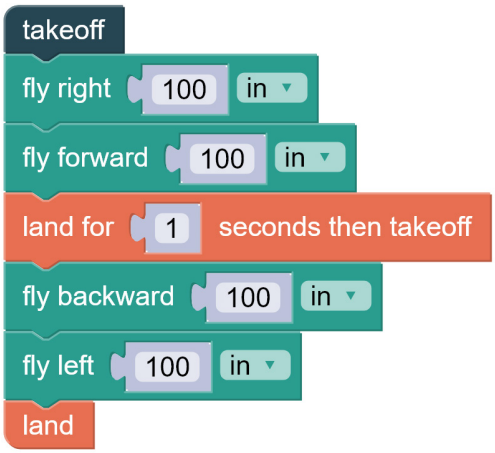
This allowed the drone to fly in an L shape around the storm clouds, collect the payload and return back the way it came.

That's a lot of blocks though. Can we reduce the number of blocks and still achieve the same outcome?

Let's get find out.

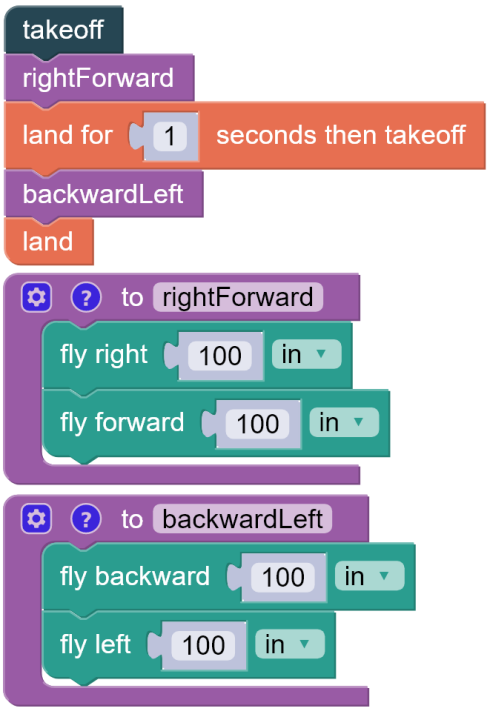
Have look at the code and see if there are any parts that are repeating. Remember "**DRY**" means :
Don't Repeat Yourself

The first thing we might notice is that our initial block is "yaw right 90 degrees" which turns the drone to face right. It then flies forward 100 yards. That's two blocks to turn right and fly forward 100 yards.



Remember "**Looking At The Numbers**" in **lesson 2**, when we flew to the ship, collected the payload and then turned around 180º to return. Turns out it wasted time each time it turned around, so we changed the code to fly forward, then fly backward. We will do the same thing here.

We can replace the "**yaw right 90º** and **fly forward 100 yards**" with **fly right 100 yards.** Our code is shorter already but we still need to tidy this up by making some **`functions`**.



Let's create **`rightForward`**. It's a new function that has two navigation blocks:
fly right 100 yards
fly forward 100 yards
We create this function once and then we can fly in a nice right angle shape going right and forward.

Let's create the reverse now, **`backwardLeft`**:
fly backward 100 yards
fly left 100 yards

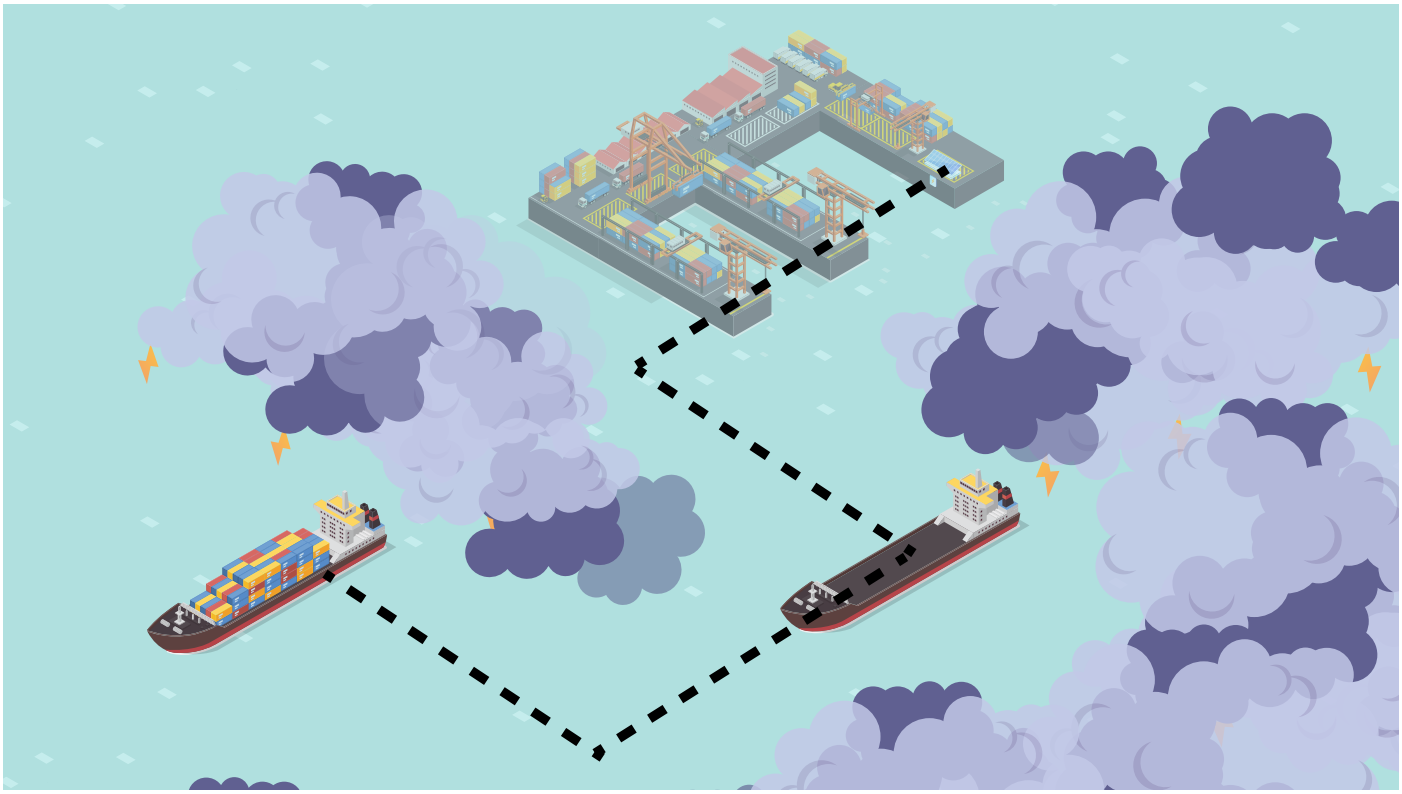
Awesome! We can now fly there and back in a right angle path at any time. How would you add more directions like **`rightBackward`** or **`forwardLeft`**? Try build those two yourself. We might need them later.

Awesome! Now all that is left is to test this code in the simulator and add in a loop to go there and back again.

Riding Out The Storm:

In the last lesson we mirrored a triangle and flew in a perpendicular shape to collect the cargo. Now we need to map out a new path to the next ship.

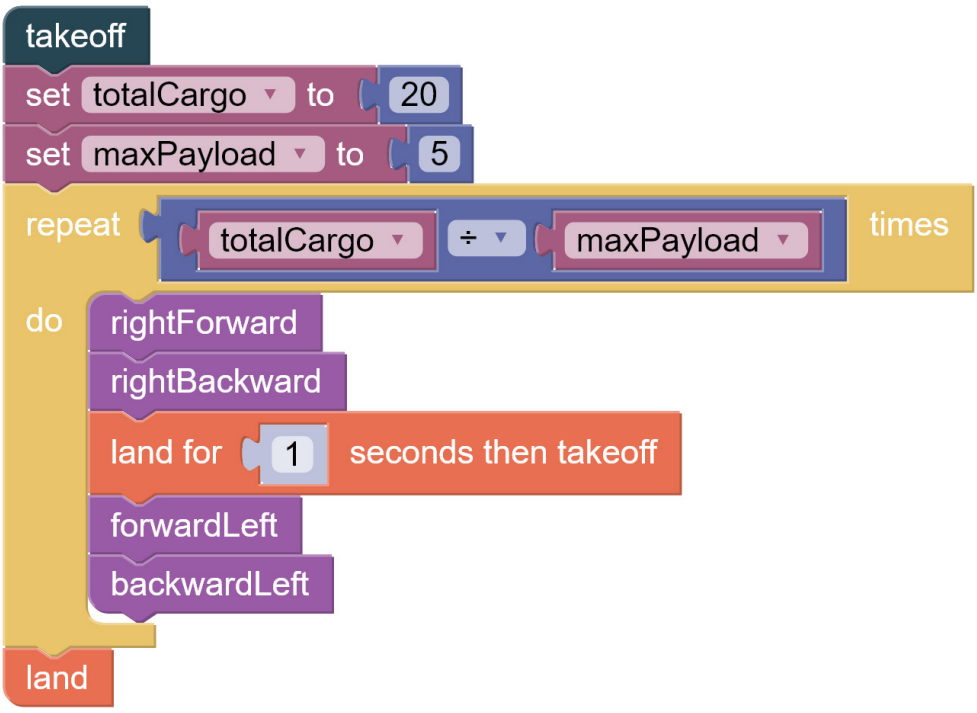
We know we are going right and then forward to get to the first ship. What then? Well, we'd want to fly right and then backward. Let's see what that looks like.



Amazing! Now we reverse that and we're all done. What a clever path. It's coding time and once again DroneBlocks to the rescue. Let's code up our own storm... so to speak.

Tool Time Take Two:

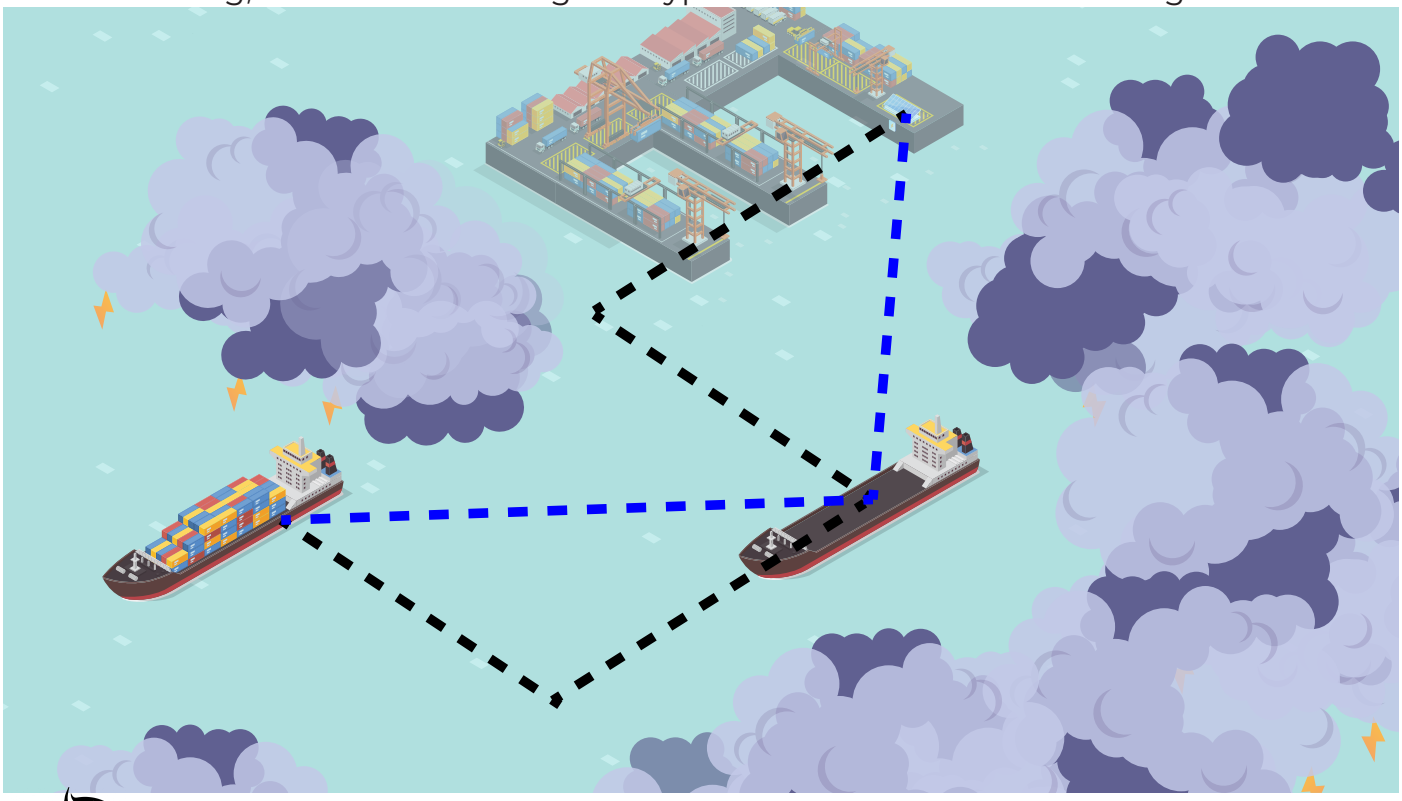
So we've created a few functions to make our code nice and readable. We start by going right and forward, then right and backward, collect payload, forward left and then backward left. Amazing!

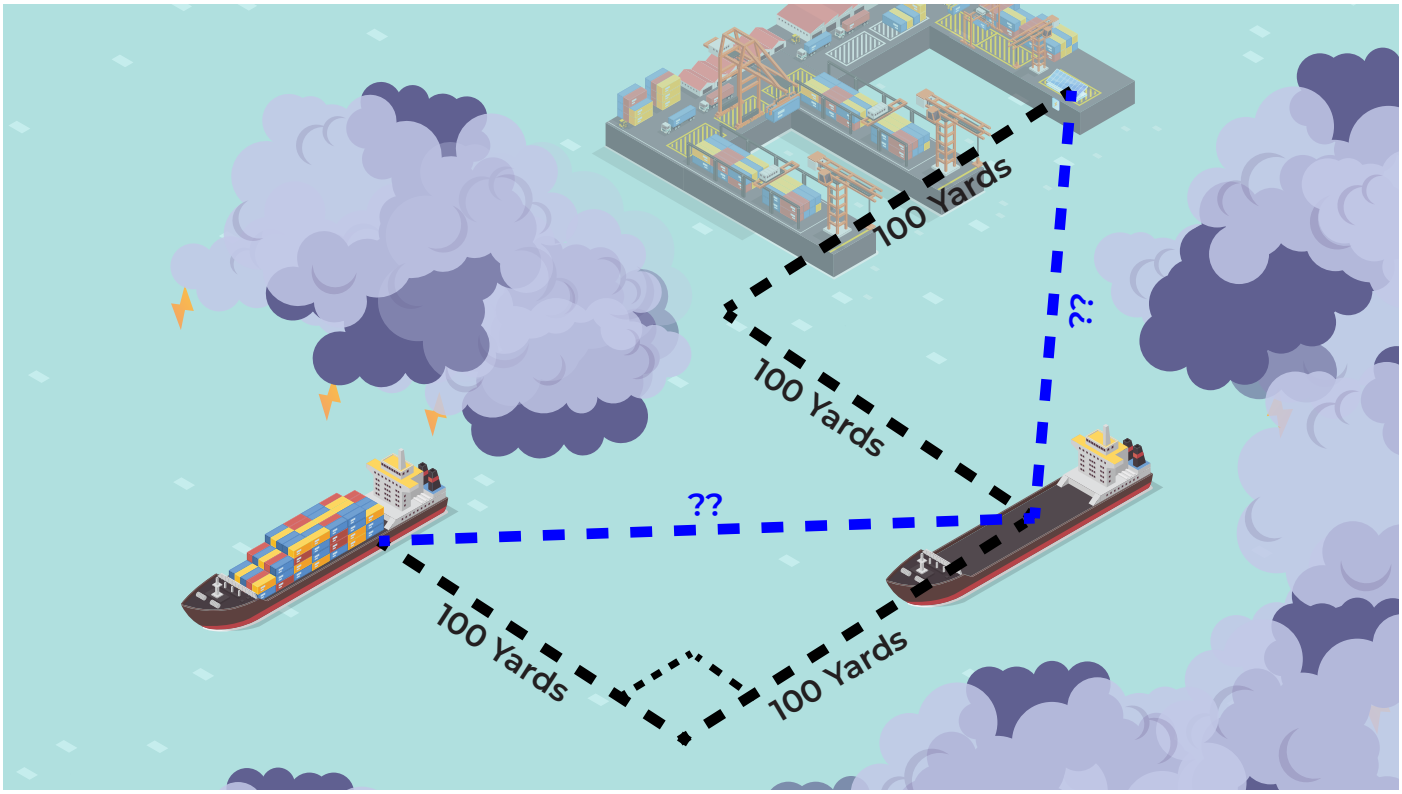


We Can See Clearly Now:

Just like in life, all storms eventually come to an end. This storm is ending which means we can use some clever math do make our flight path much more efficient. Let our previous lessons combine. Let's have a look to see how we can do better.

That's amazing, we can travel along the hypotenuse and cut down our flight time.





In the last lesson we learnt about Pythagoras. We can use that to figure out the length of the blue path. The current distances of the black paths are still 100 yards each. What was the magic formula? $c^2 = a^2 + b^2$. Cool let's work that out.

$c^2 = 100^2 + 100^2$

$c^2 = 10\ 000 + 10\ 000$

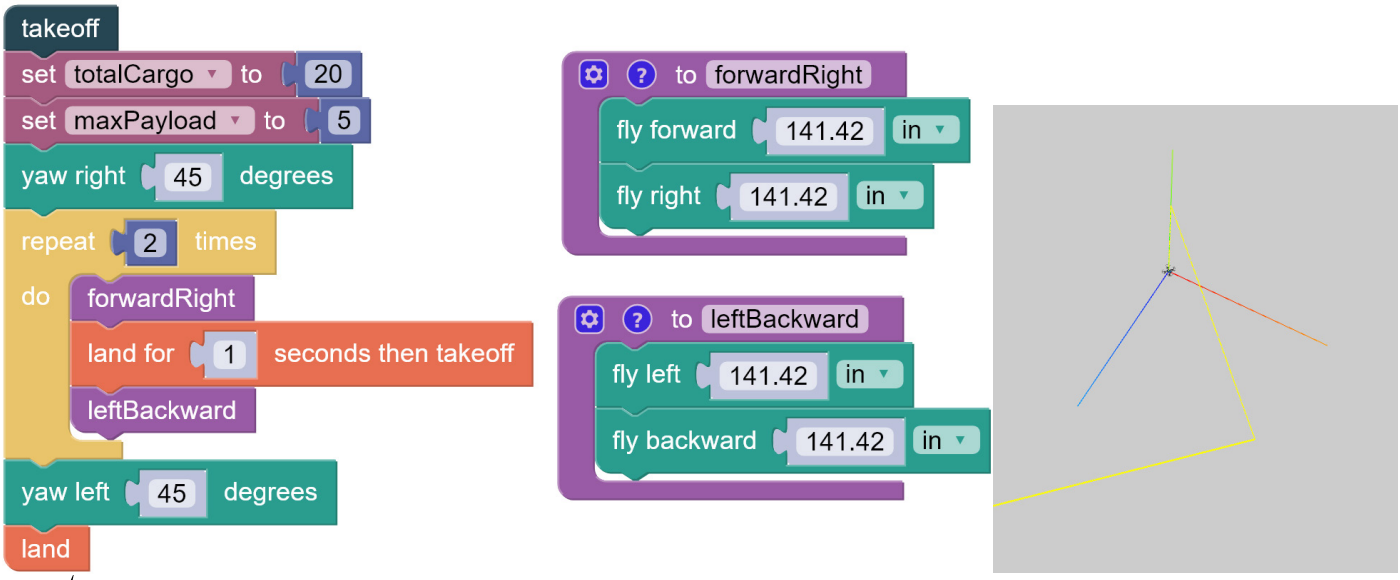
$c = \sqrt{20\ 000}$

$c = 141.42$

That's the length covered.

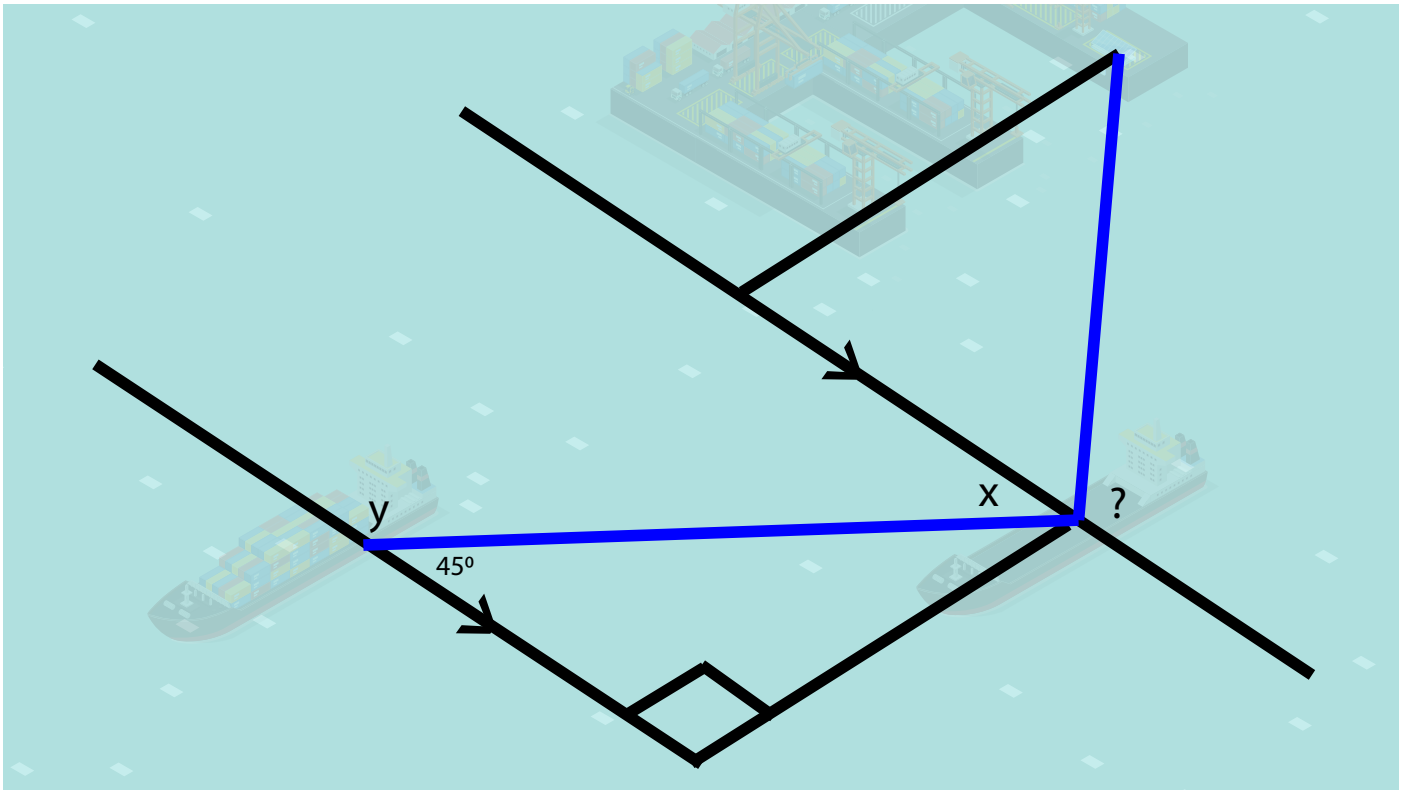
Now what about the angle. Once again, the right-angled isosceles triangle has a 90 degree right angle and 2 x 45 degree angles. Fantastic. Let's code that up quickly.

The collectionDistance is now 141.42 which means we will get the drone to fly in a right angle with each side being 141.42 yards in length. Amazing!



Interior Angle Minigame:

The storm is over. You're amazing and you're going down in the history books for sure! Now let's look at our flight paths and learn something new and exciting!



Using two amazing rules we can work out all the angles in the above image.

1. Alternate angles (Z Shapes) are equal
2. Co-interior angles on parallel lines add up to 180 (the two angles are supplementary)

Solve for x:

We need to start somewhere. x has an alternate angle. That means **x = 45⁰**. We can also solve for x by looking at the bisected right angle:
x = 90⁰ / 2
x = 45⁰

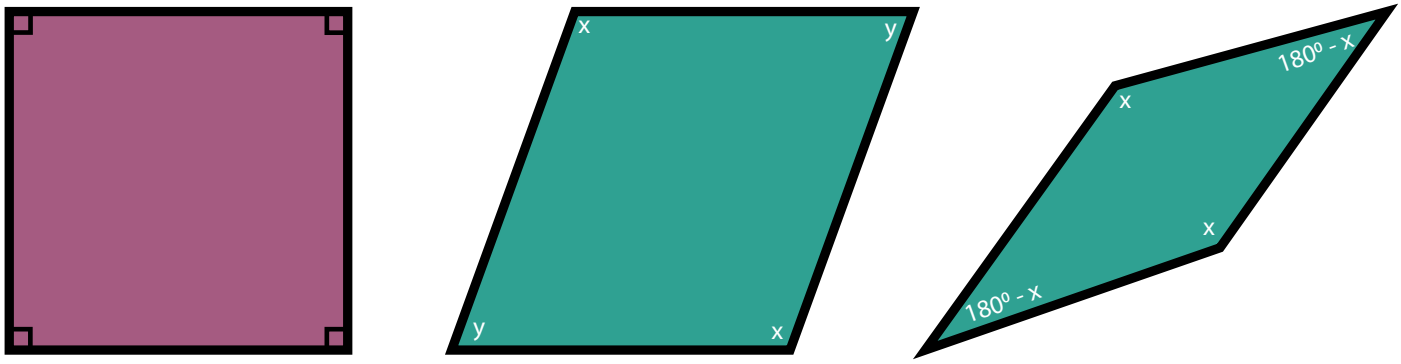
Now that we have our x value we can figure out y. Co-interior angles add up to 180⁰ so:
180⁰ = x + y
y = 180⁰ - x
y = 180⁰ - 45⁰
y = 135⁰

Lastly let's solve the ? angle:
x = 45⁰
? = 180⁰ - x
? = 135⁰

That's amazing! Math time! Interior angles are great fun! Now let's finish off our lesson with seeing how angles all add up in a rhombus.

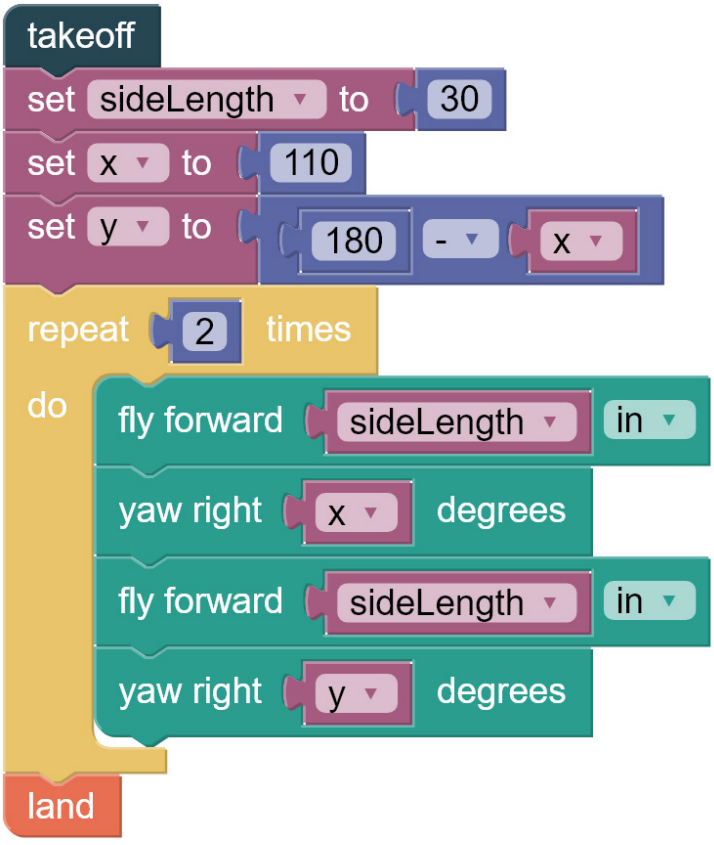
All Aboard The Rhombus

A rhombus is a quadrilateral with four sides of the same length. It is safe to say every square is a rhombus, but not every rhombus is a square.



Squares have equal lengths on each side as well as all angles are 90° (right angles).
A rhombus just has equal lengths on each side, but the angles can differ.
Opposite angles are equal, co-interior angles add up to 180° and all the interior angles add up to 360°.

Let's create a new program for a Tello Drones to fly in a rhombus, but in a very clever way!



In this program, all we need to do is enter in the sideLength and x angle. y angle is calculated automatically. How cool is that.

Save your code and load it into the simulator and let's see this in action.

I love how awesome drones can demonstrate shapes.

Play around with some x values, starting with 90° and watch how the square slowly change shape.

Another day, another problem solved. What a great way to end the lesson.

In our next lesson we are going to recover some lost supplies using the coordinate plane! We will learn how to graph points as well as interpret plot points. Extremely exciting stuff!