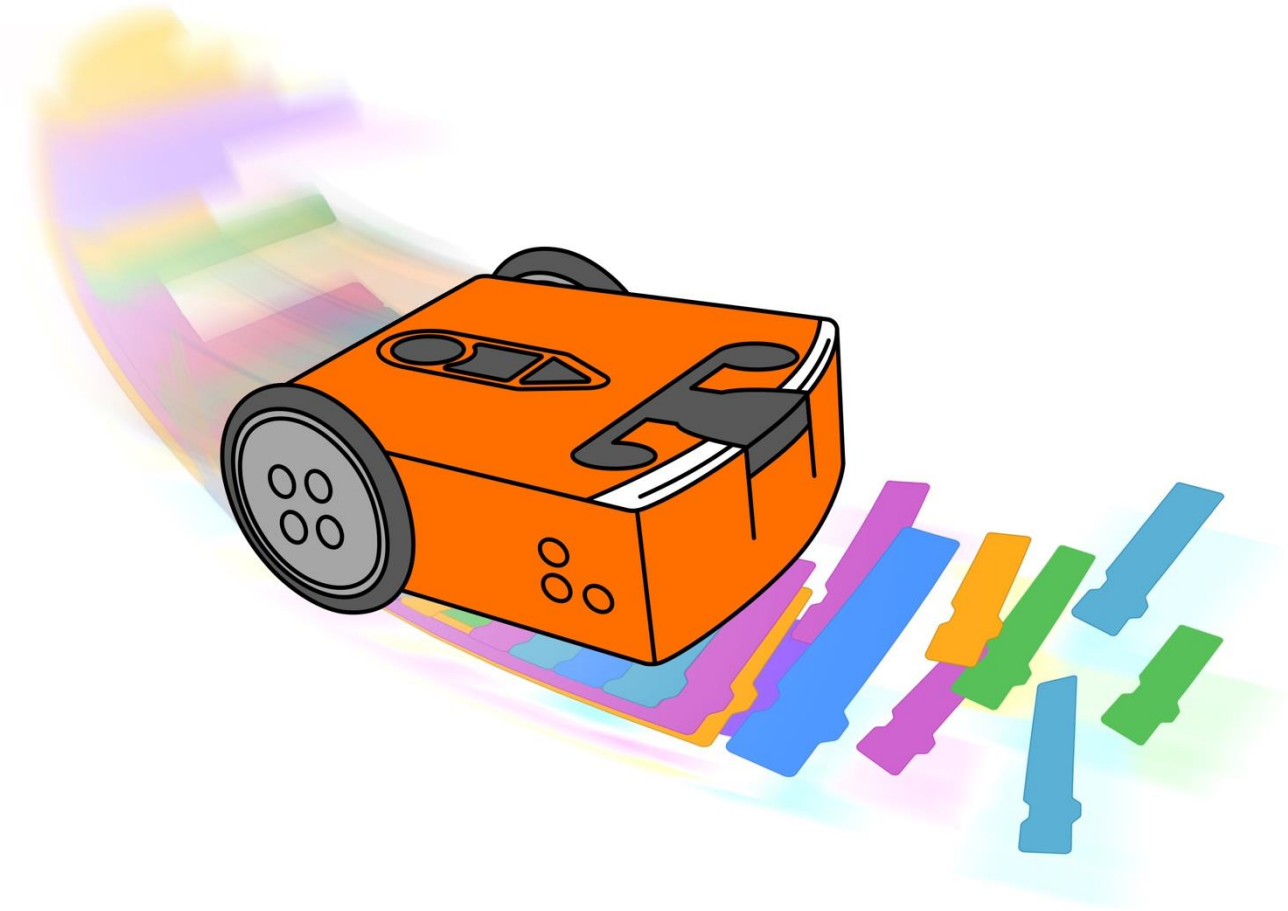
EdScratch lesson activities

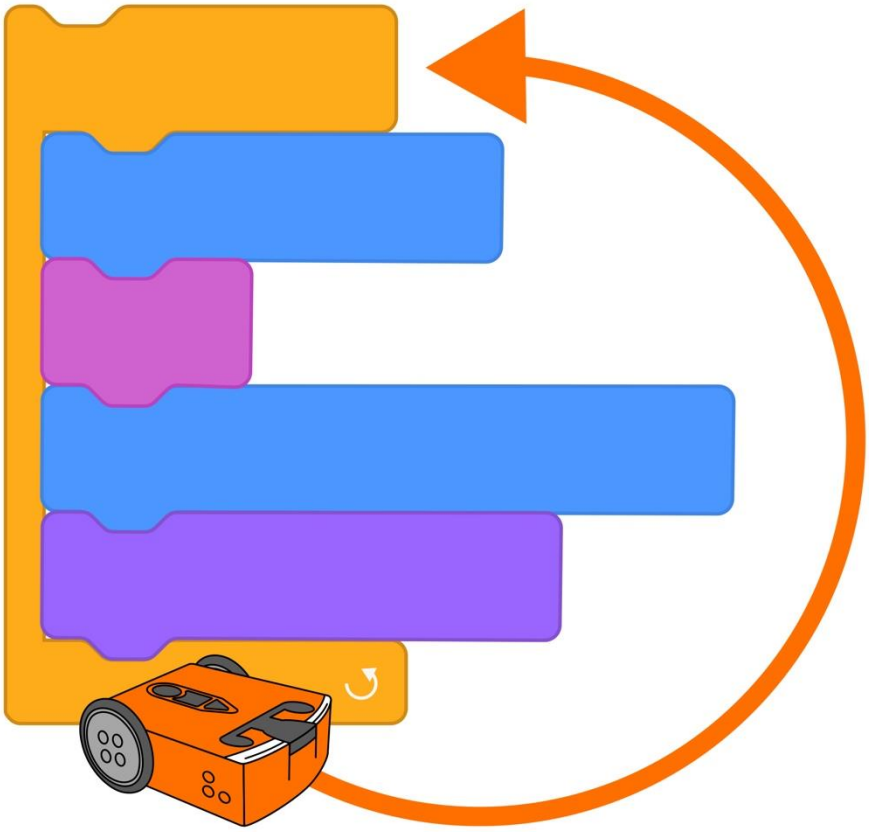
Student worksheets and activity sheets



Document number: 3.2.3.4.1 Rev 1.1

Contents

Unit 3: Got loops?



Edison

Name_____

U3-1.1 Let's explore repeating steps

To get a computer, like the Edison robot, to do what you want it to do you need to give very specific instructions. You need to write code that says exactly which actions you want to happen in exactly which order you want each action to happen.

Don't forget

When you write a program for your Edison robot in EdScratch, you are telling the robot what to do and in what order to do each thing. Each EdScratch block is one action you are telling the robot to take. The order you connect the blocks in your program tells the robot in what sequence to do each action. Edison will do the actions one at a time, starting with the top block.

Task 1: Drive in a square

Write a program for Edison using EdScratch so that your robot can drive in a square. Your program should only use blocks from the Drive category to control the motor outputs. Download your program and use activity sheet U3-1 to test your program. Make sure your program has Edison end in the exact same spot it started.

1. How many blocks do you have in your program not counting the start block?

   _____

2. Look at the blocks in your program. What do you notice? Is there a pattern to the blocks?

_____

_____

_____

Task 2: Use a loop to drive a square

To get Edison to drive a square, you need to program the robot to drive each side of the square and turn at each corner of the square. You might have noticed that this makes a pattern in the code: drive the side, turn, drive the side, turn, drive the side, turn, drive the side, and turn one last time, back to the starting position.

Lots of programs have repetition, where a bit of code is used over and over. Repeating stuff is one thing that computers are really good at doing. Unlike a person, a computer doesn't get bored doing the same thing exactly the same way again and again.

Imagine you wanted to get Edison to do the same thing 100 times. Would you want to write out that program using 100 repeating blocks? Would you find that boring to write? Do you think you would be able to write the whole program without making a mistake?

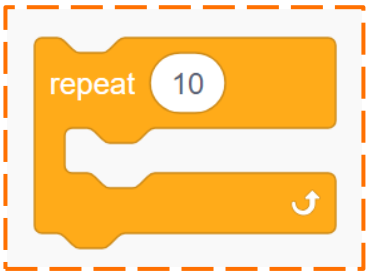
Name_____

There is an easier and more efficient way to get a computer to repeat commands multiple times. You can get the code to repeat by using something called a loop.

Jargon buster

A loop is a special piece of code that tells a computer to repeat something multiple times. Loops are a type of control structure because loops control other bits of code in a program.

In coding, using loops lets us repeat other bits of code multiple times without having to write each command over and over. In EdScratch, loop blocks are in the Control category in the block pallet. One of the loop blocks in EdScratch is the repeat block:



There are different types of loops. The repeat block is a definite loop.

Jargon buster

A definite loop is a type of loop which will repeat for a set number of times. The repeat block in EdScratch is an example of a definite loop. You tell the loop how many times to repeat using this block's input parameter.

Like all loop blocks in EdScratch, the repeat block wraps around other blocks.

Why is that?

Look at the shape of the repeat block. See how it has a shape a bit like a mouth? Other blocks can sit inside the opening of this block's 'mouth'. Any block that sits inside the repeat block is inside this loop. All blocks inside the loop will be repeated.

Remember, Edison will follow each EdScratch block one at a time. The robot will see the loop block first and know that any blocks inside that loop need to be repeated as many times as the repeat block's input parameter says. The robot will then do the action of each block inside the loop in order. When it gets to the bottom of the blocks in the loop, it will move back to the top of the loop and start again!

Name_____

Try using a repeat block to make a program for Edison to drive a square. You should be able to write a program for Edison which uses only **three blocks** after the start block, including one repeat block. Download your program and use activity sheet U3-1 to test your program. Make sure your program has Edison end in the exact same spot where it started.

3. What value do you need to have in the input parameter in the repeat block to get Edison to drive a square?

   _____

4. Why do you need to have that be the value?

_____

_____

_____

Name_____

U3-1.1a Change it up: Drive a triangle

Even little changes to inputs can make the output of a program completely different. A great example of this is changing the number of repetitions in a loop. Imagine if you wrote a program with a loop that repeats four times, then changed the input so that it repeats five times instead. What would happen when you ran the updated program?

What to do

Write a program for Edison using EdScratch so that your robot can drive in a triangle. Your program needs to use a definite loop control structure, so be sure to include a repeat block. Your program should be as efficient as possible, so try to use as few blocks as you can while still completing the task.

Download your program to your robot and use activity sheet U3-2 to test your program. Make sure your program has Edison end in the exact same spot where it started.

1. How many blocks did you need to use in order to write a successful program (not counting the start block)?

   _____

2. What value do you need to have in the input parameter in the repeat block to get Edison to drive a triangle?

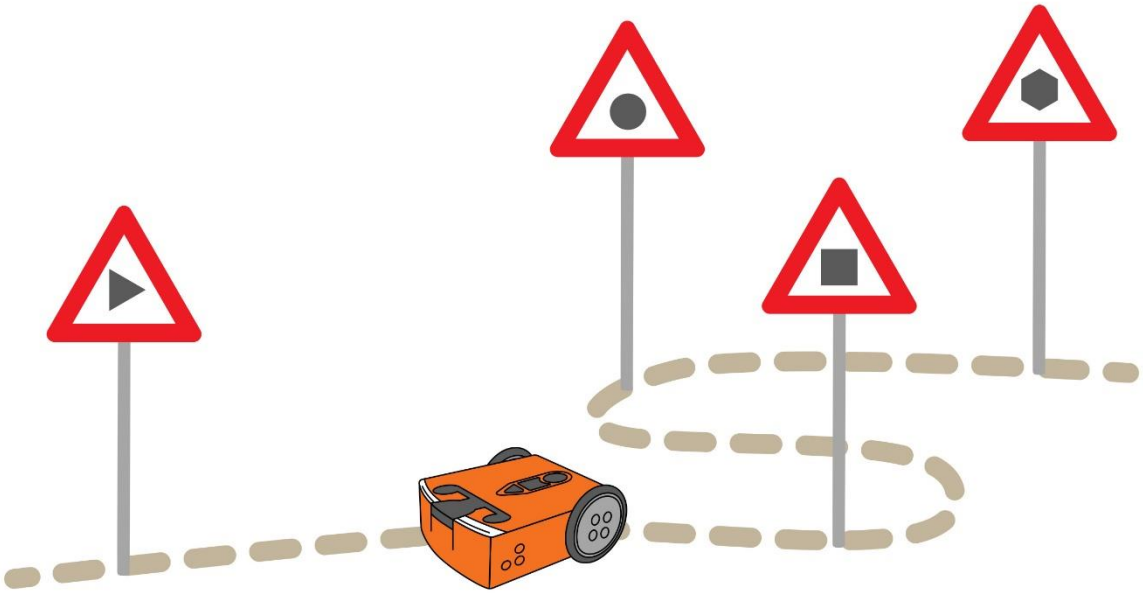
   _____

3. Why do you need to have that be the value?

_____

_____

_____



Name_____

U3-1.1b Change it up: Drive a hexagon

Even little changes to inputs can make the output of a program completely different. A great example of this is changing the number of repetitions in a loop. Imagine if you wrote a program with a loop that repeats four times, then changed the input so that it repeats three times instead. What would happen when you ran the updated program?

What to do

Write a program for Edison using EdScratch so that your robot can drive in a hexagon. Your program needs to use a definite loop control structure, so be sure to include a repeat block. Your program should be as efficient as possible, so try to use as few blocks as you can while still completing the task.

Download your program to your robot and use activity sheet U3-3 to test your program. Make sure your program has Edison end in the exact same spot where it started.

1. How many blocks did you need to write a successful program (not counting the start block)?

   _____

2. What value do you need to have in the input parameter in the repeat block to get Edison to drive a hexagon?

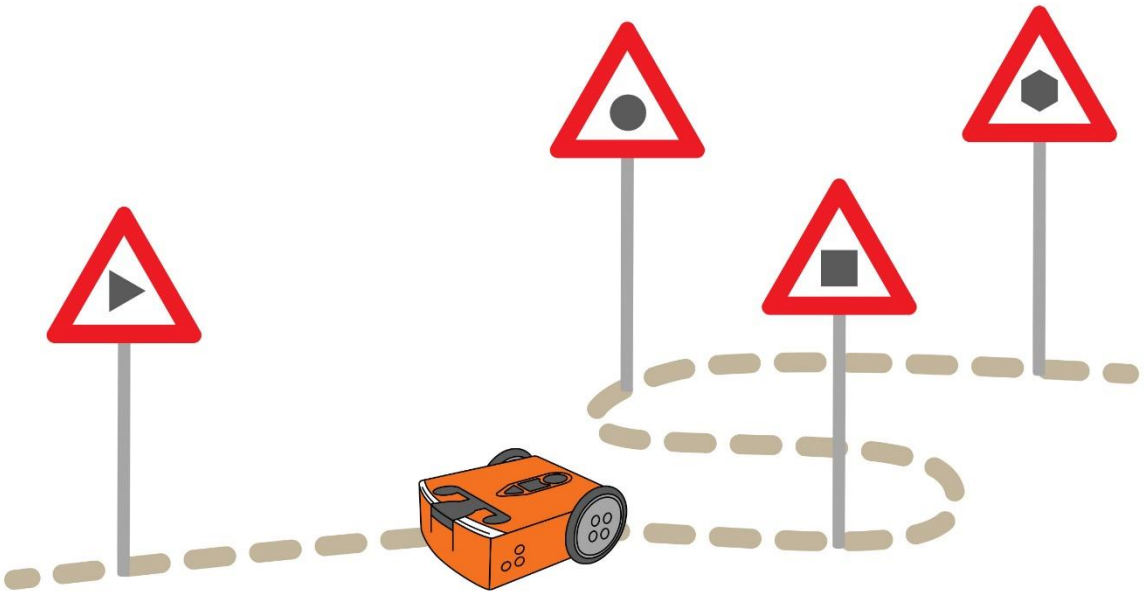
   _____

3. Why do you need to have that be the value?

_____

_____

_____



Name_____

U3-1.1c Challenge up: Choose your shape

Using a definite loop allows you to write a program to get Edison to drive a shape using only a few blocks of code. You can control how many times the program repeats the code commands inside the loop by changing the input of the repeat block.

What do you notice about the number of sides and angles a shape has compared with the input you need in your definite loop? Can you use this pattern to help you write a program to drive any shape?

What to do

Choose a shape which has sides and angles to drive using your Edison robot.

Make a workspace to test your program by either drawing your shape on paper or marking it out on the floor or a desk with coloured tape.

Hint!

You might want to choose a regular shape for this challenge. A regular shape means a shape where all the sides are equal.

Write a program for Edison using EdScratch so that your robot can drive your shape. Your program needs to use a definite loop control structure, so be sure to include a repeat block. Your program should be as efficient as possible, so try to use as few blocks as you can while still completing the task.

Download your program to your robot and test it out using your workspace.

1. What value would you need to have in the input parameter in the repeat block to get Edison to drive a regular (meaning that all sides are equal) 12-sided shape?

   _____

2. There is a pattern between the number of sides and angles a shape has and the number of times you need a loop to repeat in order to drive that shape. Describe how you used this pattern to help you determine the input parameter you needed in the repeat block to get Edison to drive your shape.

_____

_____

_____

_____

_____

Name_____

U3-1.1d Challenge up: Drive a circle

Using a definite loop, like the repeat block, is helpful when you want to write a program for Edison to drive in a shape because shapes have repeating patterns. You have probably noticed a pattern between the number of sides and angles a shape has compared with the input you need to use in a definite loop in a program that gets Edison to drive that shape. Can this pattern help you drive a circle even though a circle has no sides or angles?

What to do

Write a program for Edison using EdScratch so that your robot can drive in a circle. Your Edison needs to drive in the shape of a circle, not just spin in one spot. Your program needs to use a definite loop control structure, so be sure to include a repeat block. Your program should be as efficient as possible, so try to use as few blocks as you can while still completing the task.

Hint!

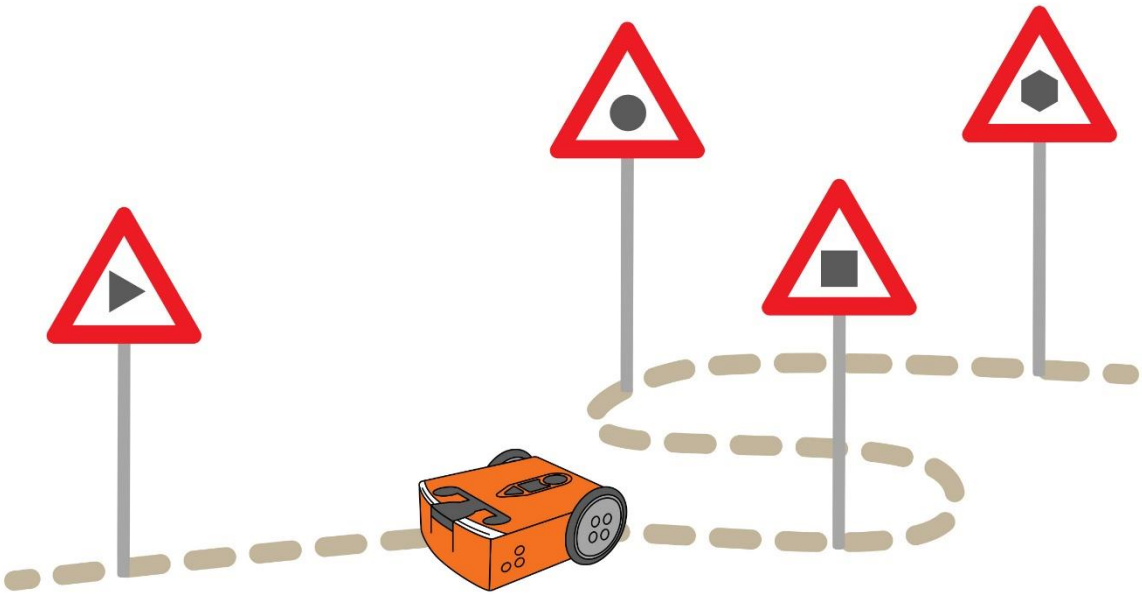
What do you notice about how a shape looks the more sides it has? If you are feeling stuck, try looking at shapes with many sides, such as a decagon and an icosagon. Use the pattern you see to help you write your program.

Download your program to your robot and use activity sheet U3-4 to test your program.

1. What does your program look like? Write your program below. Be sure to include all the input parameters you used.

Name_____

2. Does your robot drive in a perfect circle? If not, can you think of a reason why not?



Name_____

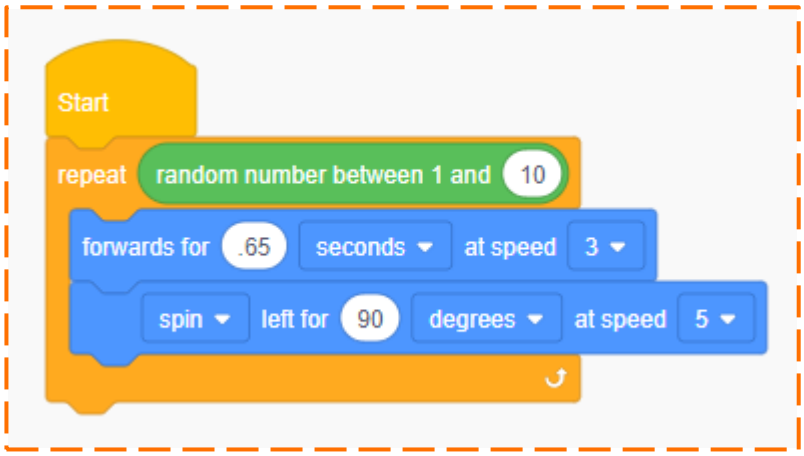
U3-1.1e Change it up: Drive a square?

How many loops does it take to drive in a square? You know that a square has four sides and four angles, so the robot needs to repeat driving and turning four times. That means that if you write a program for Edison to drive in a square using a definite loop, like the repeat block, you need to have the loop repeat four times.

What happens if the loop only repeats three times? How about if it repeats nine times?

What to do

Look at this EdScratch program:



This program is using a special input parameter for the repeat block: the random number block! This block tells Edison to pick a number between 1 and 10 at random. That's how many times the robot will loop the code inside the repeat block.

Write the program in EdScratch. Download your program to your robot and use activity sheet U3-1 to test your program. Try running the program several times to see what happens.

1. What happened when you ran the program? Did the same thing happen every time? Why or why not?

_____

_____

_____

_____

_____

Name_____

U3-1.1f Challenge up: Doodle-bot challenge

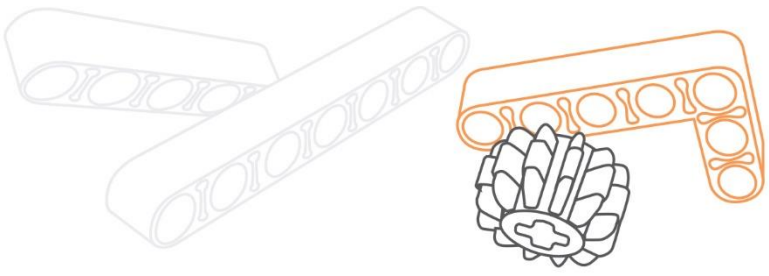
You can program Edison to drive in patterns which make many different shapes. If you attach a pen to your robot, you will be able to get Edison to draw those shapes too!

What to do

Attach a pen to an Edison robot. You can use EdCreate parts or any other materials you like.

Write a program in EdScratch so that when you run the program, your robot will draw a shape. Once you have built your pen attachment and programmed your robot, try running the program to see your shape!

1. Write about what you experienced in this challenge. What problems did you encounter? What did you do to overcome those problems? How did you attach the pen? What shape did you try to draw? Did it work?



Name_____

U3-1.2 Let's explore loops and sequence

Loops are a very useful control structure in coding. Loops can help make programs more efficient by letting you repeat commands without needing to write the same blocks of code multiple times.

When you use loops in programs, you still need to think carefully about sequence. This is especially true when you make programs which have some code inside of a loop and some code outside of the loop.

Let's try making a program that will let Edison drive a quadrilateral. Your program will need to have some of the code inside of a loop, but some of the code will need to be outside of the loop.

Don't forget

Sequence means going in order, step-by-step.

Try it out!

A quadrilateral is a four-sided shape. Squares are quadrilaterals, but not all quadrilaterals are squares. Look at the quadrilateral on activity sheet U3-5. This quadrilateral has four sides and four angles, but they are not all the same.

You need to write a program for Edison using EdScratch so that your robot can drive the shape of the quadrilateral on activity sheet U3-5. Your program should use blocks from the Drive category to generate the motor outputs you need. Your program also needs to use a loop from the Control category.

You need to work out the best place to start Edison on the activity sheet. Be sure your program also has Edison end in the exact same spot where it started.

Hint!

Think about the sequence of actions you want Edison to take. Remember, when you make an EdScratch program for Edison, the robot will start with the top block and do each action in order, one-by-one.

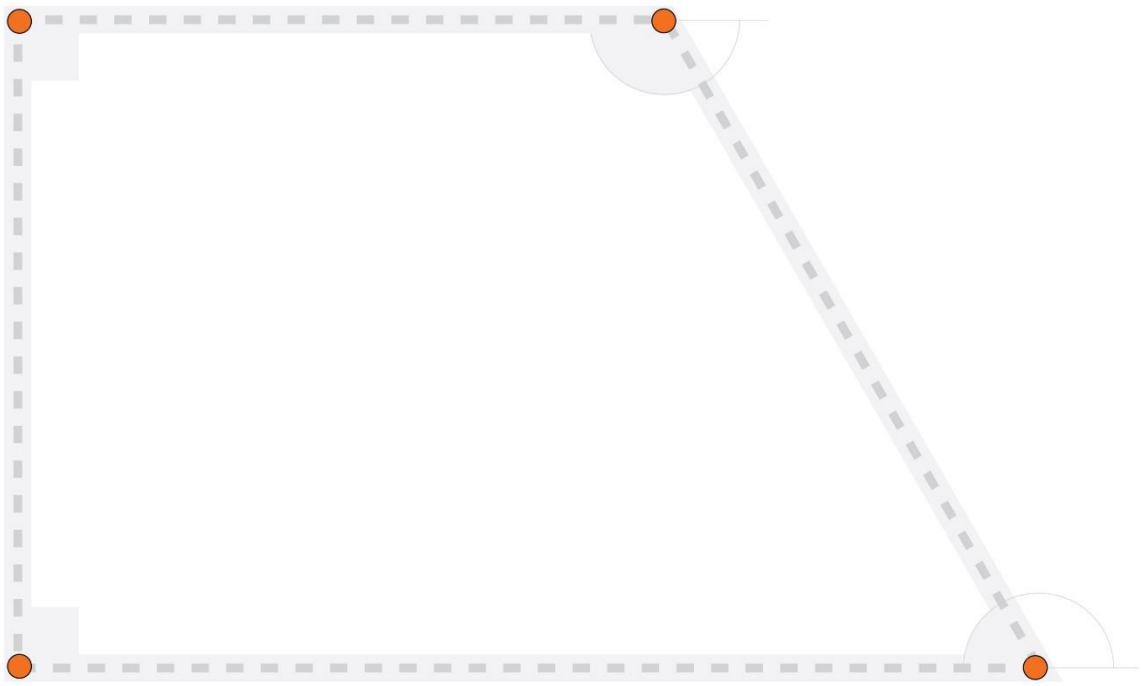
Write your program in EdScratch. Then download your program and use activity sheet U3-5 to test it out.

Hint!

You still want to make your program as efficient as possible, so try to use as few blocks of code as you can.

Name_____

1. Where on the quadrilateral did you start Edison? Mark where you started Edison, including which direction you had the robot drive.



2. Look at your start point and your program. How did using this start point affect the sequence of your program?

Name_____

U3-1.3 Let's explore forever loops

Loops let us repeat steps in a program without having to write the same code over and over. When you want a program to do the same thing many times, using a loop is easier than having to write each command again and again. This makes our code more efficient because you can tell the computer to do the same thing using much less code.

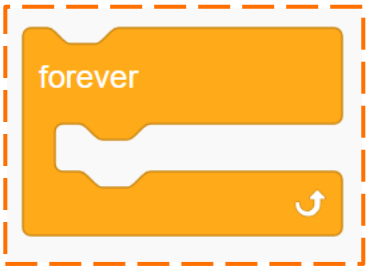
In many programs with repeating commands, you know how many times you want the program to loop. If you want to get your robot to drive in a square, for example, you know you need to have Edison drive and turn four times. In that program, you can use a definite loop which repeats four times.

To use a definite loop, you need to know how many times the loop needs to repeat. What if you don't know that? Or, what if you want to make a program that loops forever?

Don't forget

There are different types of loops. A definite loop is a type of loop which will repeat for a set number of times. The repeat block in EdScratch is an example of a definite loop.

To have something repeat forever in EdScratch, you need to use a special loop block in the Control category in the block pallet called the forever block:



The forever block is an **indefinite loop**.

Jargon buster

An **indefinite loop** is a type of loop which will repeat for an undefined number of times. The forever block in EdScratch is an example of an indefinite loop. This loop block tells Edison to keep repeating the code blocks inside this loop forever.

You can think of the forever block in EdScratch as working the same way as the repeat block does, but with the input parameter for the number of loops set to infinity!

The shape of a block in EdScratch can give you some clues about how you use it in the language. Look at the shape of the forever block. Just like all the other loop blocks in EdScratch, the forever block wraps around other blocks. All the blocks that sit inside the loop block will be repeated. What else do you notice about the shape of this block?

Name_____

1. If you write a program using a forever block, do you think you will you be able to add commands for Edison to do after the loop? Why or why not?

Try it out!

Let's turn Edison into an egg timer! Use the forever block to write a program so that Edison will wait a certain number of seconds and then sound an alarm forever.

Hint!

You can always stop a program by pushing the stop (square) button on your Edison robot.

Think about the sequence of things that need to happen for the egg timer program to work. What needs to be inside the loop? What needs to be outside of the loop? Download and test your program with your robot.

2. What does your program look like? Write your program below. Be sure to include the input parameters you used.

Name_____

U3-1.3a Challenge up: Earworm

An earworm is a song that gets stuck in your head for what feels like forever. In this activity, you need to give Edison an earworm by programming the robot with a tune and a forever block!

What to do

Program Edison to play a song or tune over and over using a forever loop block. Write your program in EdScratch, then download it and test it with your robot.

Don't forget

You can always stop a program by pushing the stop (square) button on your Edison robot.

Name_____

U3-1.4 Let's explore stacking and nesting loops

Writing programs using loops lets you be more efficient because you can get a computer to repeat actions without needing to write out the commands multiple times. Using loops also lets you tell a program to do something forever, which you couldn't do if you had to write out every command one by one.

You can also use more than one loop in a program, and you can use the loops in different ways: by **stacking** the loops together or **nesting** loops inside other loops.

Jargon buster

In block-based programming languages like EdScratch, adding blocks together is sometimes called **stacking** blocks and a program is sometimes called a **stack** or a **block stack**. That's why if you use multiple loops together in a program one after another, you can say you are **stacking** the loops.

You can also put a loop block inside another loop block. This is called **nesting** loops.

Why would you use loops in stacks or by nesting them together? Using multiple loops together in this way lets you write programs with repeating patterns. You can even write programs with patterns that repeat inside of other patterns.

Why is that?

Think about an alarm clock on a mobile phone. The alarm can be set to go off in the morning at 7:00 AM. You can set the phone to repeat that alarm every day. When the alarm goes off, it beeps on and off a set number of times. If you snooze the alarm, it stops for a certain amount of time, then comes back on, beeping on and off again for a set number of times.

Can you see how there are repeating patterns inside of other repeating patterns?

This is an example where using stacked and nested loops to write a program would be very helpful. That's because stacking and nesting loops lets you repeat whole sets of commands inside your program.

Stacking and nesting loops have different uses. By stacking loops, we can write programs to get Edison to do different sets of actions multiple times, then move on to a new set of repeated actions. By nesting loops together, however, we can write programs to get Edison to repeat whole patterns multiple times.

Name_____

Task 1: What's going to happen?

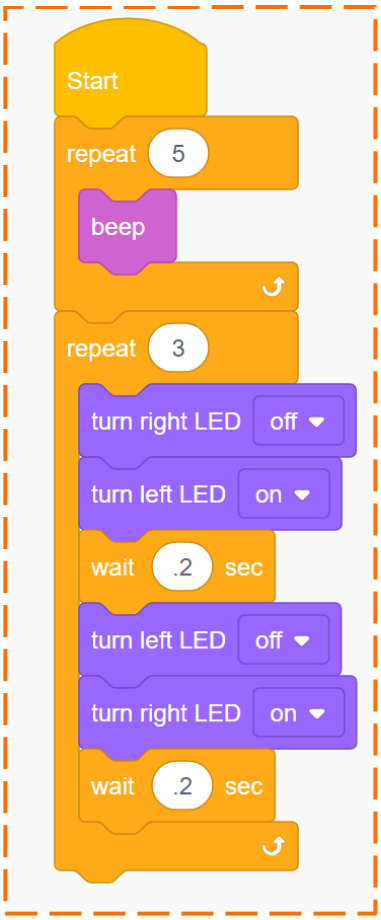
Programs that use multiple loops, especially nested loops, might seem a bit confusing at first. To understand what the program is going to do, you need to think about each action that is going to happen in sequence.

Don't forget

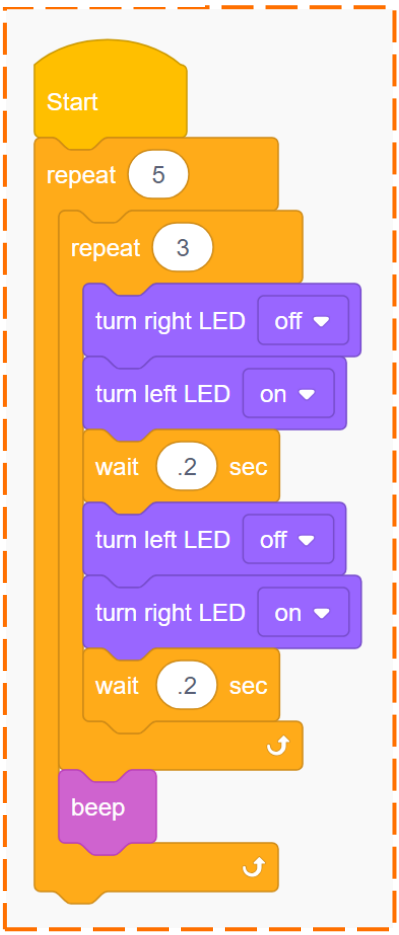
When you make a program for Edison in EdScratch, the robot will start with the top block and do each action one by one. Once it completes a block, it will move to the next block. This is true of all EdScratch programs – whether there are zero loops, one loop or multiple loops!

Look at the following programs and answer the questions about what is going to happen in each program.

Program 1:



Program 2:



Name_____

1. In program 1, how many times will the right LED turn on?

   _____

2. In program 1, which will finish first: all of the beeps or all of the LED flashes?

   _____

3. In program 2, how many times will the right LED turn on?

   _____

4. In program 2, which will finish first: all of the beeps or all of the LED flashes?

   _____

Hint!

Can you follow along with what is happening in each program? If you want to double check your answers, try writing each program in EdScratch, then download it to your Edison robot. Run the program to see what happens.

Task 2: Drive the pattern

Just like a loop lets you repeat a pattern multiple times, nesting loops allows you to repeat multiple patterns! For this activity, you need to use activity sheet U3-6.

5. Look at the pattern on activity sheet U3-6. How would you describe the pattern?

_____

_____

_____

You need to write a program so that Edison will drive the pattern on activity sheet U3-6. Your program can have Edison go across the same line more than once, but the robot must touch all the lines.

Name_____

6. How do you think you can use a nested loop to help you write an efficient program for your Edison robot to drive the pattern on the activity sheet?

_____

_____

_____

Try writing an EdScratch program so that you get your Edison robot to drive the pattern on activity sheet U3-6. Test your idea for using a nested loop to see if it works.

Hint!

You can write a program that completes the activity sheet using just five EdScratch blocks!

Name_____

U3-1.4a Change it up: Edison the designer

Lots of things that run using computer programs have repeating patterns. There are also many programs that have patterns that repeat inside of other patterns. These programs often use nested loops to repeat whole sets of commands inside a program.

What to do

Try using loops to write a program for your robot which makes Edison drive a pattern. If that design has a pattern with a repeating pattern inside of it, try using a nested loop.

Look at activity sheet U3-7 and choose one of the designs to use. For this activity, you will need to create a workspace to test your program. Make a workspace that is large enough to test your program with Edison. You could draw the pattern onto a large sheet of paper or mark it out using dark coloured tape on the floor. Copy out the design onto your workspace. Then write a program in EdScratch that gets Edison to drive that design.

Hint!

Stuck? Try breaking down the pattern into smaller sections and writing code to get Edison to drive each part of the pattern. Link all of the chunks together to get Edison to drive the whole pattern. This can help you to find places where the code repeats. Make your program more efficient by replacing repeating code with loops.

If there is a pattern inside a pattern, be sure to try a nested loop!

Mini challenge!

If you want, you can also design your own pattern to use for this activity. Make sure your design has a pattern repeating inside another pattern. Test your design by writing a program that gets Edison to drive your pattern.

Does your pattern need nested loops?

Name_____

U3-1.4b Challenge up: Dance party!

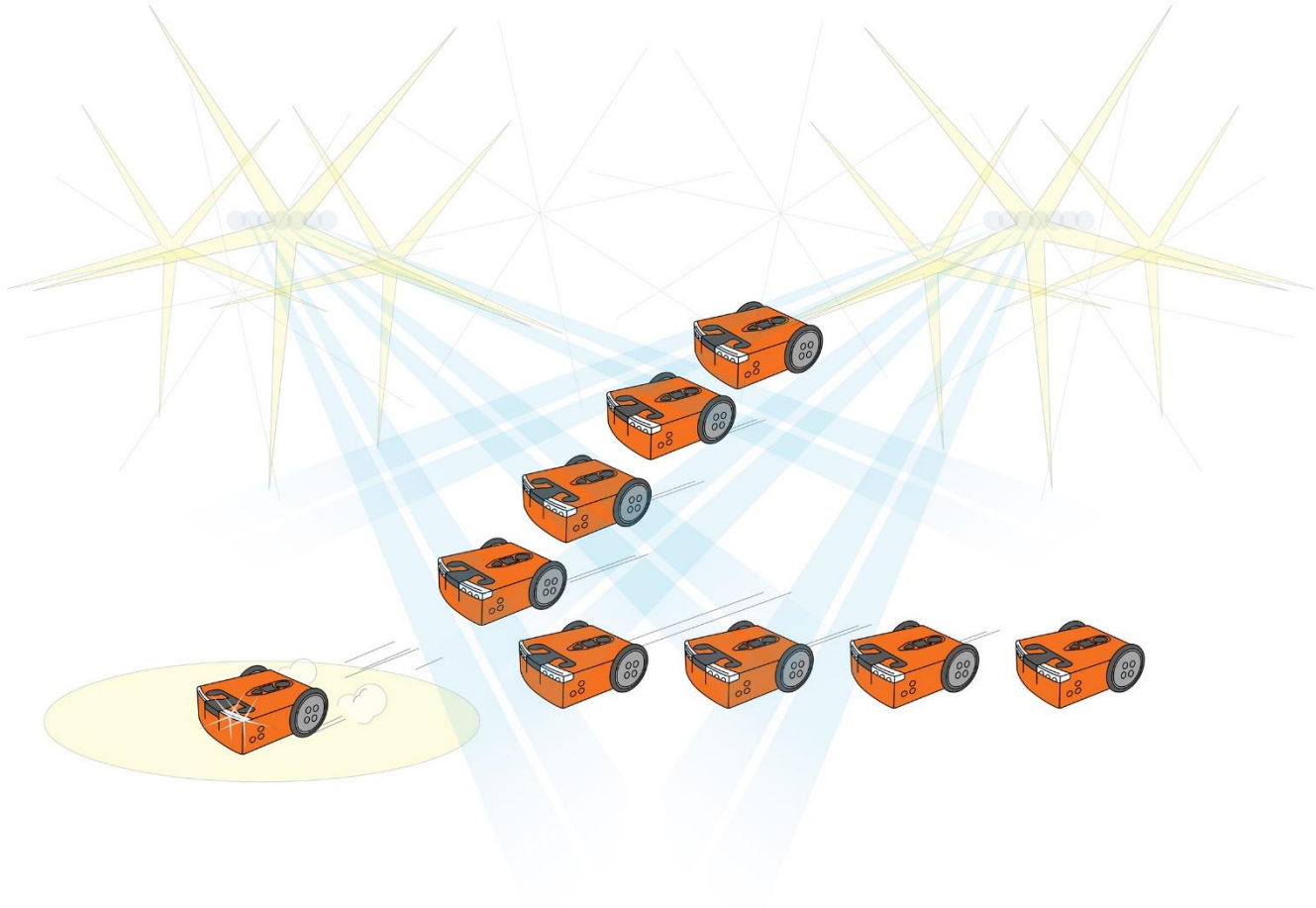
Music and dance often have repeating patterns. The chorus of a song repeats and the notes inside a song often have a pattern as well. Many dances repeat moves too. All this repetition means that if you have a robot dance party, your program probably needs loops!

What to do

Work together to program multiple Edison robots to have a dance party. You can either have all of the Edison robots dance along to a song you play from another device or have one Edison play the song while the other robots dance.

Use any of Edison's outputs (using blocks from the Drive, LEDs and Sound categories) along with loop blocks from the Control category to make a dance program. Will all of the robots do the same thing at the same time? Is one robot the star with the other robots acting as backup dancers?

It's up to you and your team!



Name_____

U3-2.1 Let's explore interrupting the main program

Different computer programming languages have different syntaxes, or rules, which make them look and feel a bit different from each other. No matter the syntax, however, all computer languages work using the same underlying logic. This is why all computer programs behave in similar ways and follow core programming logic, like sequence.

Don't forget

Logic is the organised way of doing things that makes sense to a computer. Logic determines the flow of a program.

Syntax is the rules of how a programming language works.

In EdScratch, the logical flow of a program is to start with the top block and complete each action one block at a time. Programs with loops also follow sequence. When the program sees a loop block, it executes the commands inside that loop in order. When it gets to the bottom of the loop, it goes back to the top of the loop and starts again. Even though loop blocks make programs look a bit different, these programs still follow the logical flow of top-to-bottom sequence.

There is a way to interrupt this sequential flow. You can disrupt a computer program's flow by using an **interrupt**.

Jargon buster

An **interrupt** is a special bit of code that stops the flow of the main code. It's called an interrupt because it *interrupts* the main code. An interrupt is usually used to pause the main code in order to run a **subroutine**.

A **subroutine** is a distinct set of code that is separate from the main program. You can think of a subroutine as a mini program.

To understand how interrupts work, we need to understand what is being interrupted.

What is the main program?

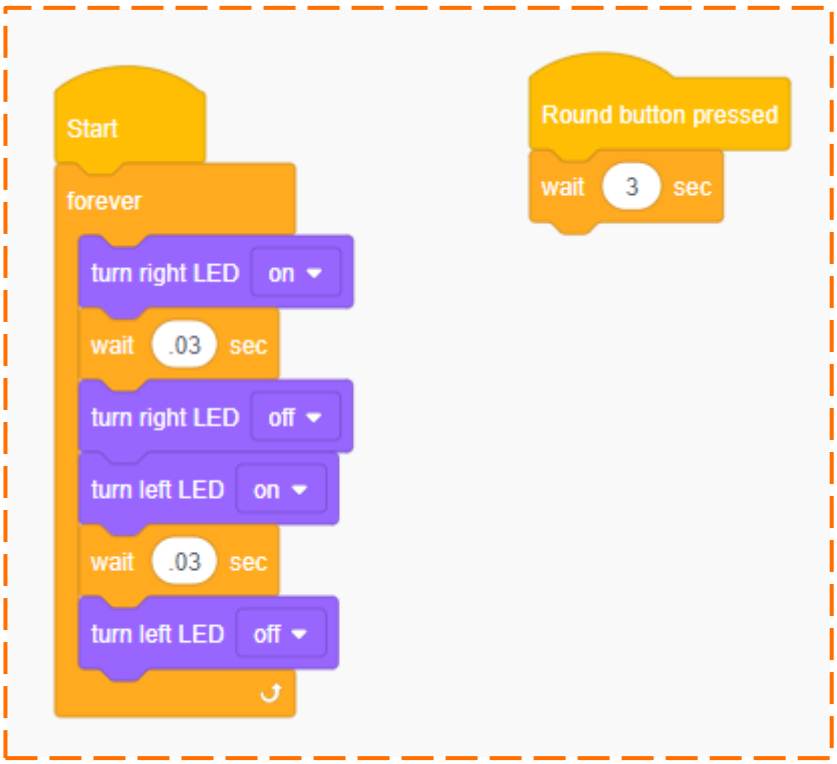
In EdScratch, the main program is whatever is attached to the yellow start block.

Whenever you write a program for Edison, you need to have at least one block in the main program attached to the start block. When you run a program with your Edison robot by pressing the play (triangle) button, this main program runs block-by-block in sequential order until it reaches the end of the program.

An interrupt can disrupt this flow.

Name_____

Look at the following program:



This program has two parts: the main program and the subroutine.

1. What does the main program tell Edison to do? *Hint:* only the blocks attached to the start block are in the main program.

_____

_____

_____

In addition to the main program, there is also a subroutine. What causes the main program to be interrupted and run the subroutine?

Subroutines will only run when a specific **event** happens.

Jargon buster

In programming, an **event** is something that happens outside of the program code that affects how the program runs. An event might be a button being pressed or information being relayed from a sensor.

In EdScratch, you need to use a block from the Events category at the start of any subroutine.

Name_____

2. Look at the blocks in the Events category in EdScratch. What do you notice about the shape of these blocks?

_____

_____

The Event blocks are interrupts that tell the program to look out for that particular event. If the event happens, the Event block interrupts the main program immediately and runs the subroutine. Once the subroutine code is complete, the program returns to wherever it left off in the main program.

Why is that?

We use interrupts in programming because interrupts allow a program to react to an event at any time while the program is running. By using interrupts, you don't have to predict when an event will occur. Without interrupts, you would have to know exactly when something was going to happen, even when that event is out of your control!

Try it out!

Write a program in EdScratch that contains both the main program and the subroutine just as they appear in the picture from earlier in this activity. Download the program to your Edison robot. Press the play (triangle) button on your robot. This will start the main program, causing Edison's LEDs to flash on and off. Now press the round button on the robot. This interrupts the main program and runs the subroutine. This subroutine tells Edison to wait 3 seconds, then return to the main program.

You can use this program to turn your Edison robot into a decider bot!

Think of a question that you can answer with a 'yes' or a 'no'. The decider bot will help you answer that question. If the right LED is lit up when you press the round button, the answer to your question is 'yes', but if the left LED is lit up, then the answer is 'no'.

Why is that?

Remember, an interrupt pauses the main program instantly, so it is possible that neither LED will be lit up when the subroutine runs. If the main program has turned off the right LED but hasn't yet turned on the left LED when the interrupt occurs, then both LEDs will be off. This is a bit like flipping a coin and having it land on its edge – rare, but it can happen!

Name_____

U3-2.1a Change it up: Try a clap instead

Remember Edison's sound sensor? This is the special bit of tech that's both a buzzer and a sound sensor. This is the bit of Edison that makes noise, like beeps or musical notes, but can also detect sounds, like a clap.

You can use Edison's sound sensor to trigger an interrupt in a program. That way, you can make a decider bot that responds to the sound of a clap!

What to do

Write a program to turn Edison into a decider bot. Your main program should have Edison flash its two LEDs on and off forever. You also need a subroutine that will interrupt the main program if the robot detects a clap. Your subroutine should tell Edison to wait for a few sounds so that you can see which LED is on and get your answer.

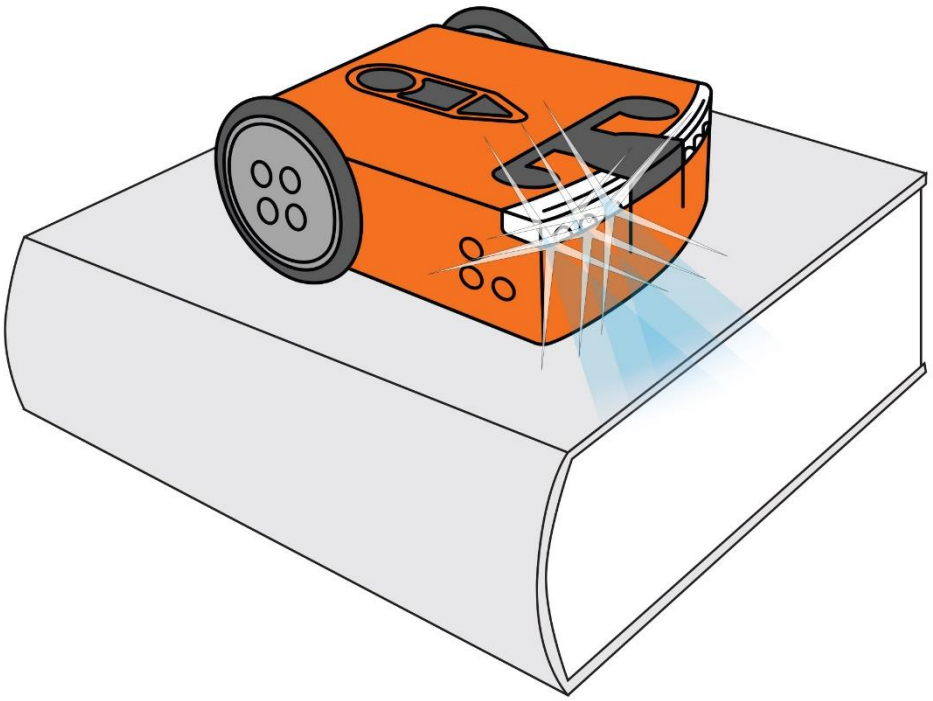
Don't forget

In EdScratch, you need to use a block from the Events category to act as an interrupt and be the first block at the start of any subroutine.

Download your program to your Edison robot and test it out.

Hint!

You can use the decider bot program from activity U3-2.1 as a base for your program.



Name_____

U3-2.1b Challenge up: Cheater bot

Using a decider bot is a fair way to choose between two options. The LEDs flash so quickly that it's very hard to 'cheat' and get the robot to answer the way you want. You can build a decider bot with a 'cheat' however… but you need to add a second subroutine to do it!

What to do

Write a program to turn Edison into a decider bot with a secret cheat. Your main program should have Edison flash its LEDs on and off forever. You also need two subroutines: one subroutine that is fair and one that cheats. The 'fair' subroutine needs to interrupt the main program if the robot detects a button being pressed, and tell Edison to wait for a few sounds so that you can see which LED is on to get your answer.

Don't forget

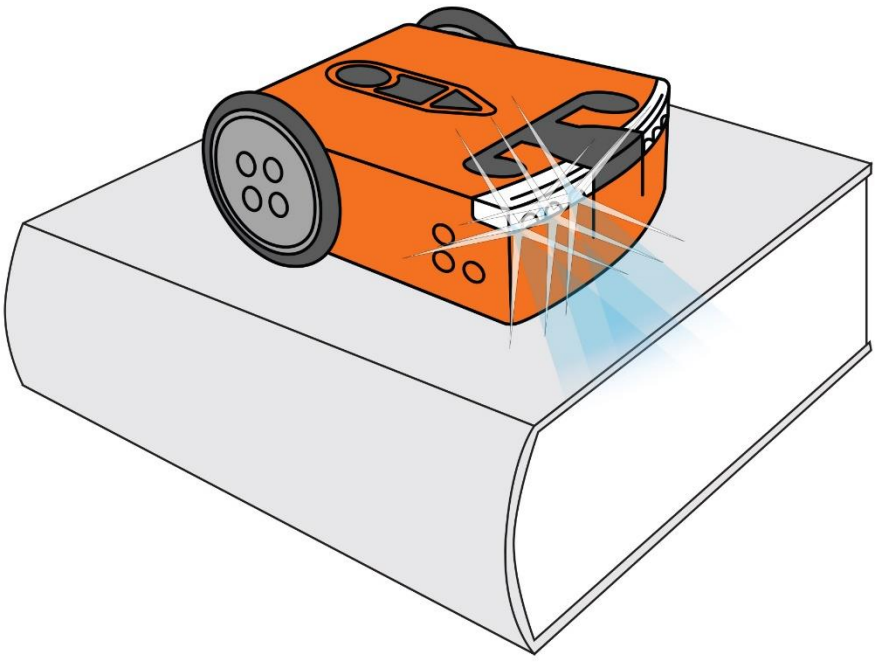
In EdScratch, you need to use a block from the Events category to act as an interrupt and be the first block at the start of any subroutine.

The second subroutine should also interrupt the main program if the robot detects a different button has been pressed. Instead of just waiting, however, you need to design that subroutine to give you a set answer.

Download your program to your Edison robot and test it out.

Hint!

You can use the decider bot program from activity U3-2.1 as a base for your program.



Name_____

U3-2.1c Challenge up: Pick one

A decider bot selects an answer from two different options. When you use Edison as a decider bot using its LEDs, you have to remember which choice the right LED represents and which choice the left LED represents. Instead of remembering, why not create a way for Edison to light up the answer!

What to do

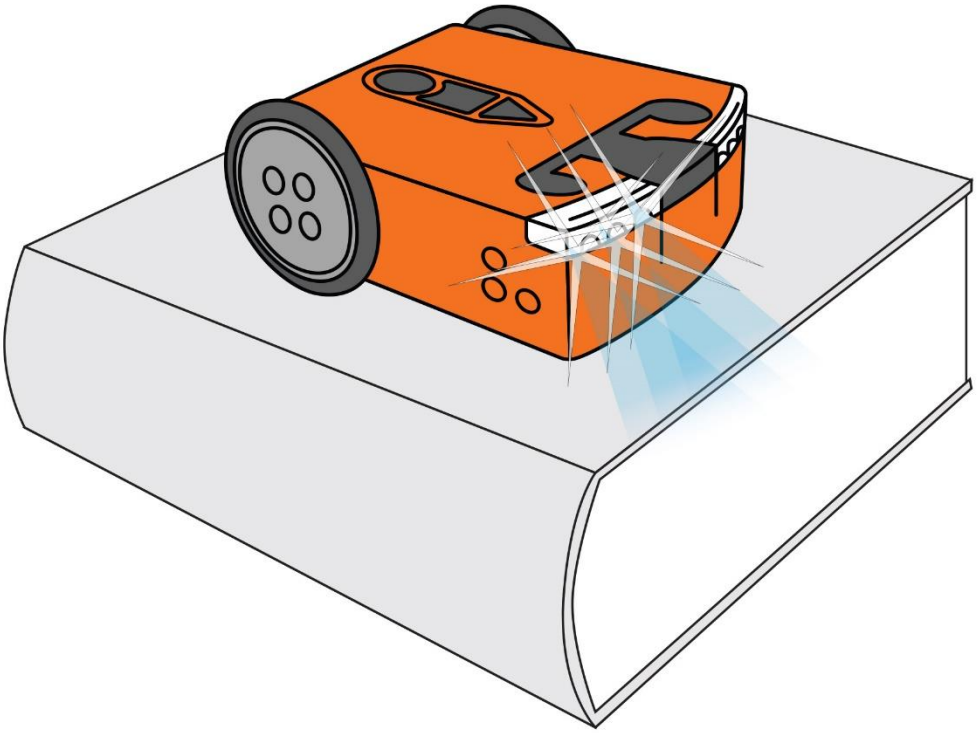
Write a program to turn Edison into a decider bot. Your main program should have Edison flash its LEDs on and off forever. You also need a subroutine that will interrupt the main program if the robot detects a button being pressed and tells Edison to wait for a few seconds so that you can see which LED is on and get your answer.

You also need to create some sort of physical set-up so that the decider bot's choice lights up the answer. That way you can write down two choices, and Edison will light one up!

Download your program to your Edison robot and test it out using your creation.

Hint!

You can use the decider bot program from activity U3-2.1 as a base for your program.



Name_____

U3-2.2 Let's explore comments in coding

Part of learning how to code is learning to speak another language: a computer programming language! The more coding you do in a programming language, the easier it is to understand programs written in that language. You can look at a program, follow each command in order, and start to figure out what the program will do when you run it.

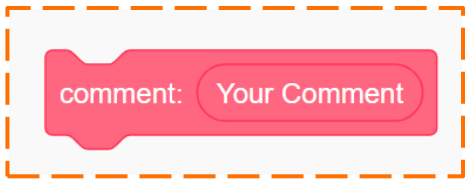
There is also a tool to help make it easier for us to read programs called **comments**.

Jargon buster

In programming, **comments** are notes that the programmer adds to help keep track of things. Comments are messages to document what's happening in the program and clarify things so that people can understand the program.

What comments look like depends on the programming language. Sometimes comments can look a bit like code, but comments are not actually code. When a computer runs a program that has comments in it, the computer will ignore the comments. Comments are just for people!

In EdScratch, you can add a comment to your program by using the special block in the Comments category. Here is what the comment block looks like:



The comment block looks similar to other blocks in EdScratch, but it works a bit differently. See where it says 'Your Comment' in the block? That's where you can write in your note. Remember, this block isn't code for Edison. When Edison sees a comment block in a program, it simply skips the block and moves on to the next command. You can think of the message you write in a comment block as that block's input parameter, but instead of being information for Edison, it is a note for a person.

How do you use comments?

In many ways, how you use comments in your program is up to you. Comments do not need to follow syntax, so there isn't a specific way you must write your comments. You can phrase things in your comments however you think makes the most sense. You can also add comments to your code wherever you think you need a note to help explain what comes next.

Why is that?

Because comments are just written for people, you don't have to worry about the computer understanding what you mean. That's why comments don't need to be written in the syntax of the programming language.

Name_____

Adding comments to a program makes it easier for other people to read your program, but the person that is most likely to read your comments is you in the future! This is because comments are a helpful tool for debugging your programs.

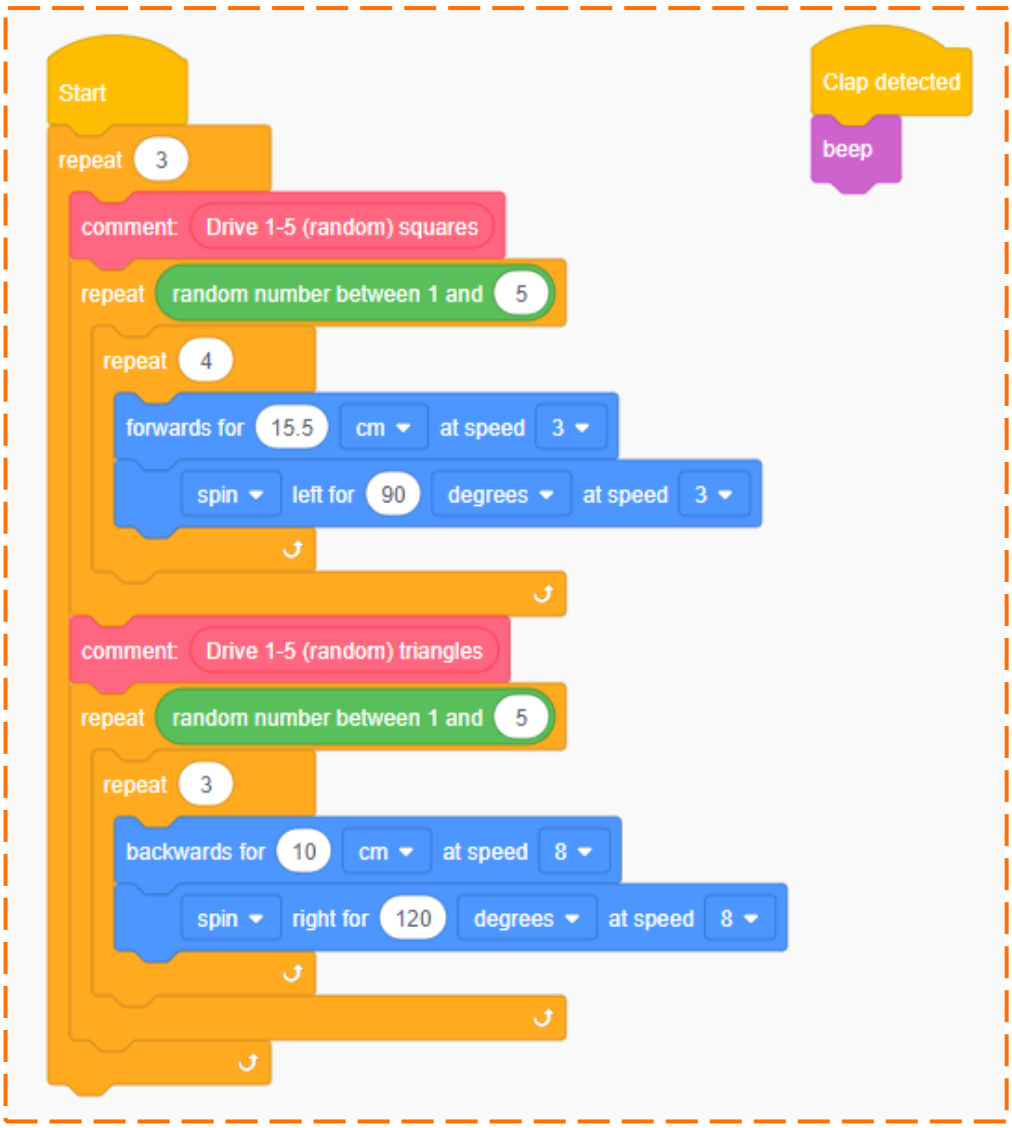
By adding a comment, you can organise your thinking and keep track of what it is you are trying to do. That way, if something in your program doesn't work the way you intended, it is easier to go back and see where the issue might be.

Don't forget

Finding and fixing problems in a computer program is called debugging.

Try it out!

Look at this program:



The programmer has added some comments to the code to help make the program easier to read. Use this picture to answer the following questions.

Name_____

1. What do you think Edison will do if you program the robot with the code in the picture?

2. Do the comments make it easier for you to understand what the programmer wants the program to do? Why or why not?

Now try programming your Edison with the program in the picture.

3. Did the program work the way you expected? Describe anything that happened that you didn't expect.

When the programmer ran this program, something unexpected happened:

"I only wanted Edison to beep when I clapped, but the robot keeps beeping repeatedly the whole time the program is running. Why is that?"

4. Why is Edison beeping? *Hint:* Is there a clue in the EdScratch environment?

Name_____

U3-2.2a Challenge up: Create and comment

Using comments is a good way to organise your thinking as you write code. Adding comments lets you leave a little message to yourself or someone else about what the code is doing. This is really helpful if you need to come back to a program later on, especially if you need to do some debugging. Reading the comments in a program is a quick way to know what is meant to be going on!

Don't forget

The person that is most likely to read your comments is you in the future!

What to do

In this activity, you need to design a program in EdScratch to run with your Edison robot. What your program does is up to you, but it needs to include the following things:

- ☐ a main program
- ☐ at least one subroutine that is triggered by either a button press event or a clap event
- ☐ at least one loop
- ☐ at least two types of outputs

Design and write your program in EdScratch. Add comments as you write your program to help keep track of what you are trying to do and to help someone else read your program.

1. What do you want your program to do? Describe what your program should do when you run it in Edison.

_____

_____

_____

_____

2. Where in your program did you include comments? What did you use them to say? Write down at least one example of a comment you included in your program. Explain where you put the comment block, why you put it there, and what the comment says.

_____

_____

_____

_____

_____

Name_____

Test your program with your Edison robot. Does it work like you expected? Is there anything happening you don't want to happen? Is your program doing everything you want it to do and is each action happening in the order you want?

3. Describe any issue you had when you first tested your program. What did you do to fix the issue?

_____

_____

_____

_____

Once you have your program working just how you want, go back and look at your comments. Do they all still make sense? Do you need to add any new comments or get rid of any of the ones you put in originally? Refine your comments so they work with your final program.

4. What does your finished program look like? Write your program below. Be sure to include the input parameters you used and your comments.

Name_____

U3-2.2b Challenge up: Share your comments

Comments help make it easier for other people to read your code and understand what your program is meant to do. That is, so long as other people understand what your comments mean!

A lot of the time, people work together on coding projects, and comments make it easier to collaborate. Because there aren't strict rules about how to write comments, however, different people write comments in different ways. The result is that sometimes comments are more confusing than the code!

What to do

You will need at least one partner for this activity.

Design and write a program in EdScratch. Keep your program a secret from your partner!

What your program does is up to you. Be sure to include comments to help keep track of what it is you are trying to do and to help someone else read your program.

Hint!

You can use activity U3-2.2a as inspiration for what to include in your program.

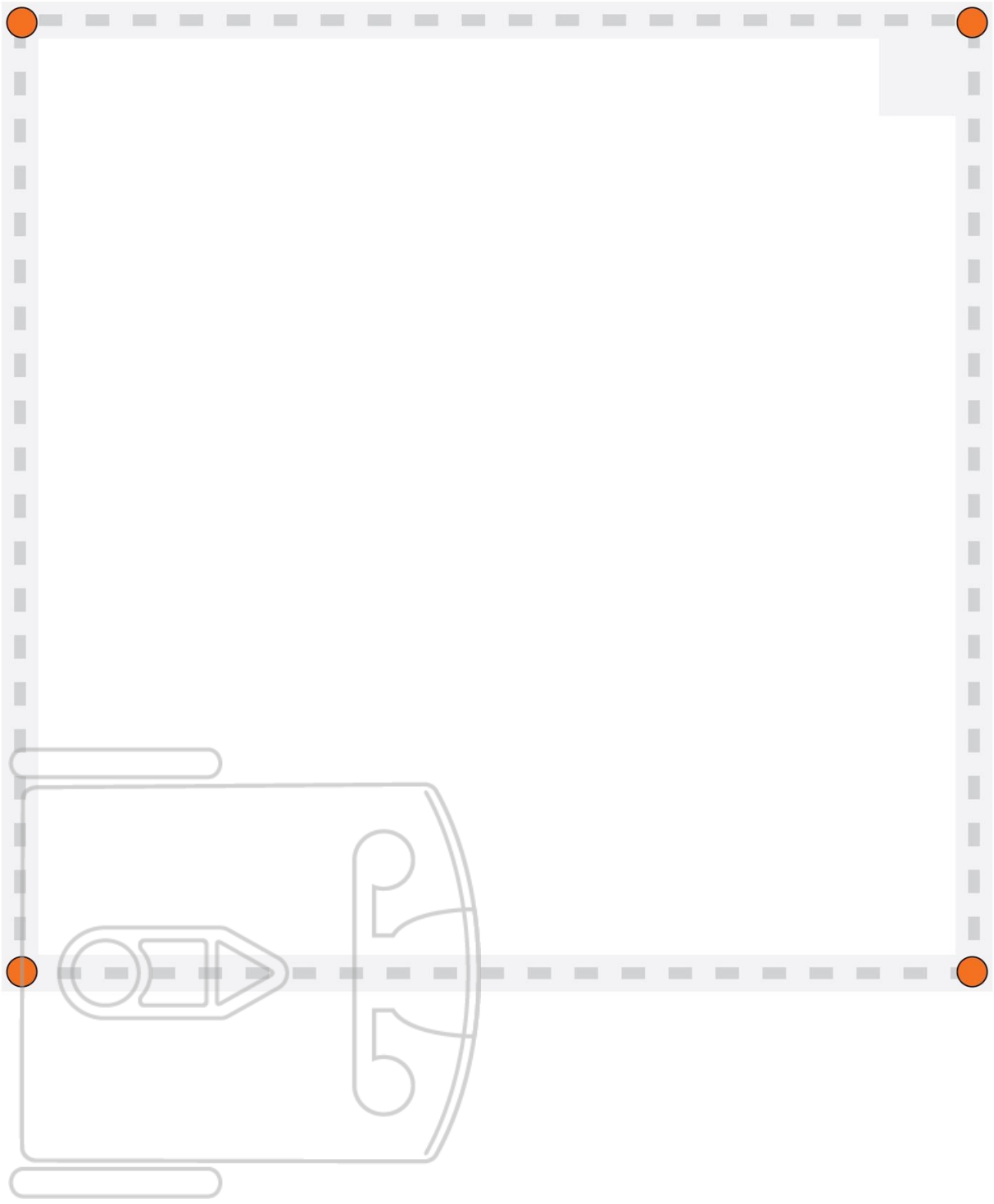
Once both you and your partner have finished writing your own programs, swap with each other.

Without running the program in Edison, can you understand your partner's program by reading their code and comments? Can your partner understand your program and comments?

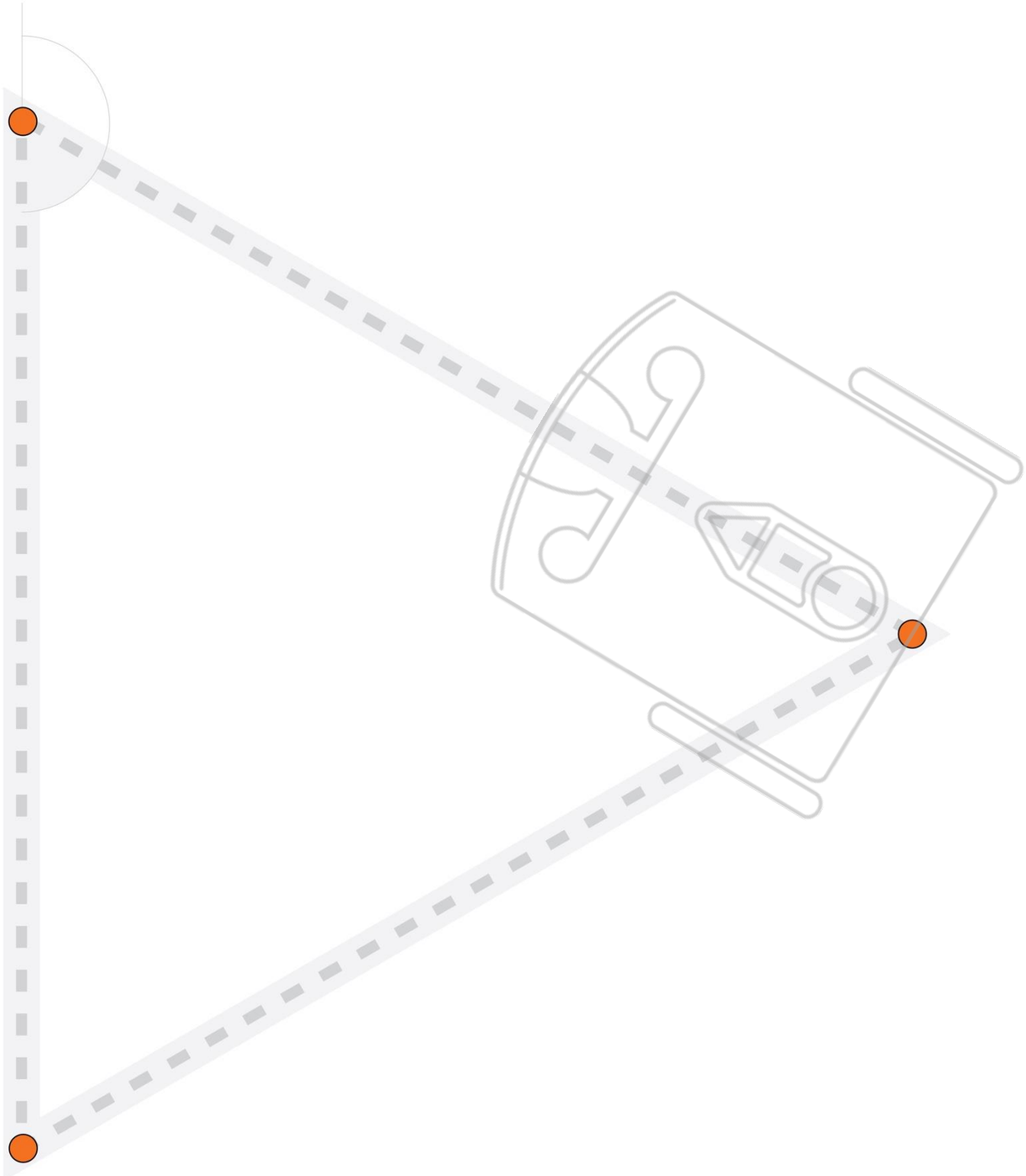
1. Pretend that you are in charge of making the rules for how everyone should use comments in EdScratch. What rules should everyone follow when they use comments so that the comments are helpful and easy to understand? Work together with your partner to decide the new rules for using comments.

Name_____

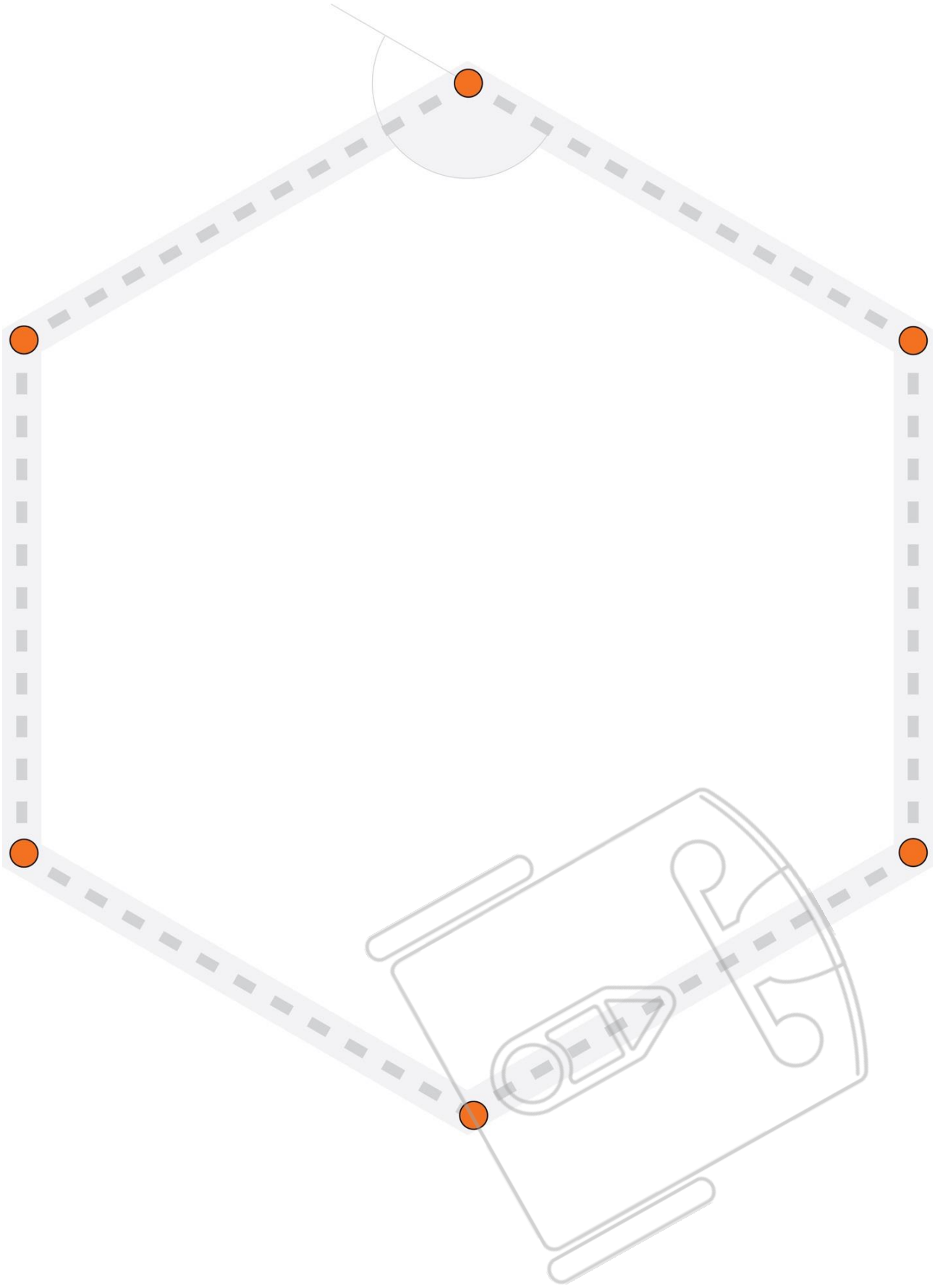
Activity sheet U3-1: Drive a square



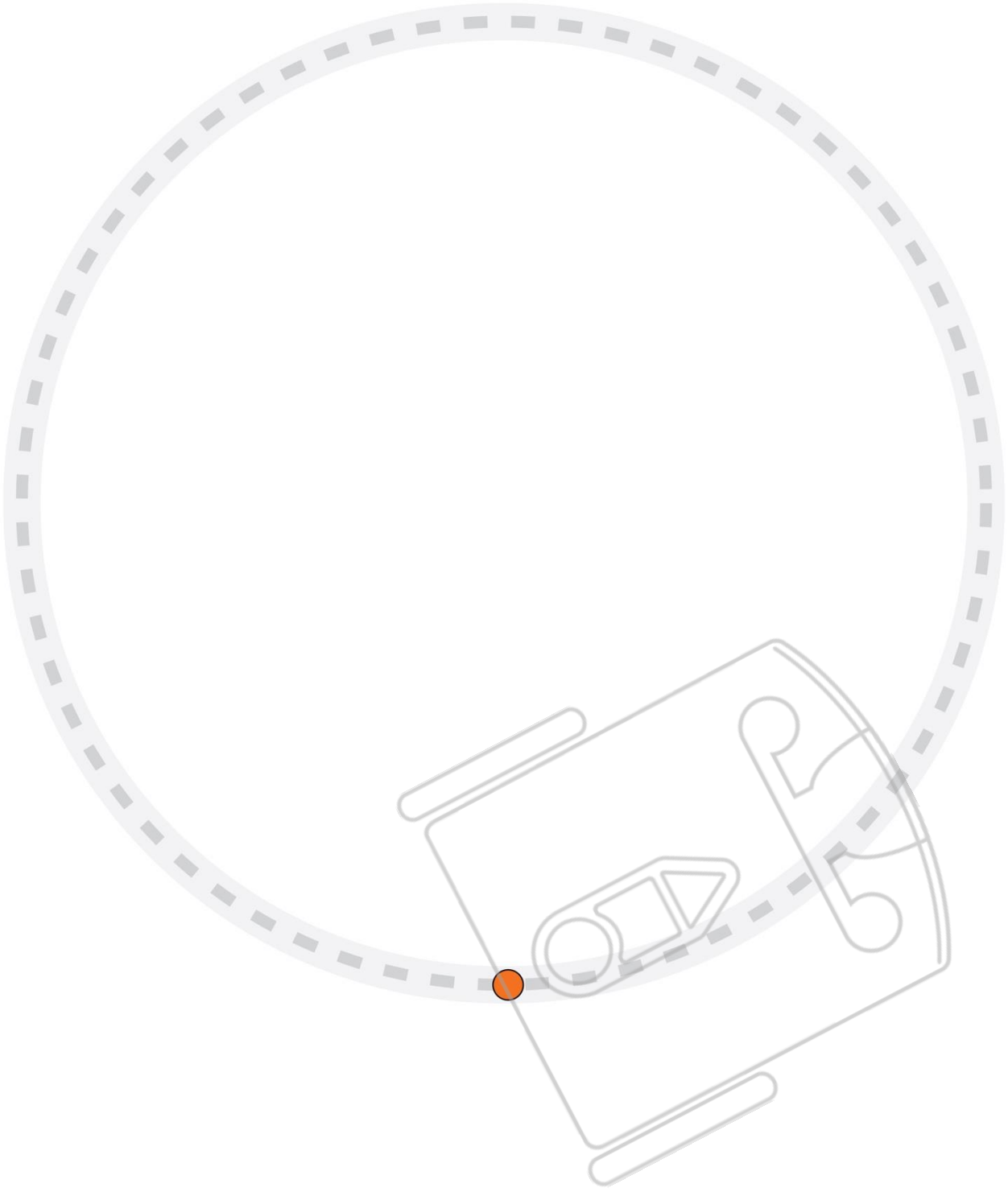
Activity sheet U3-2: Drive a triangle



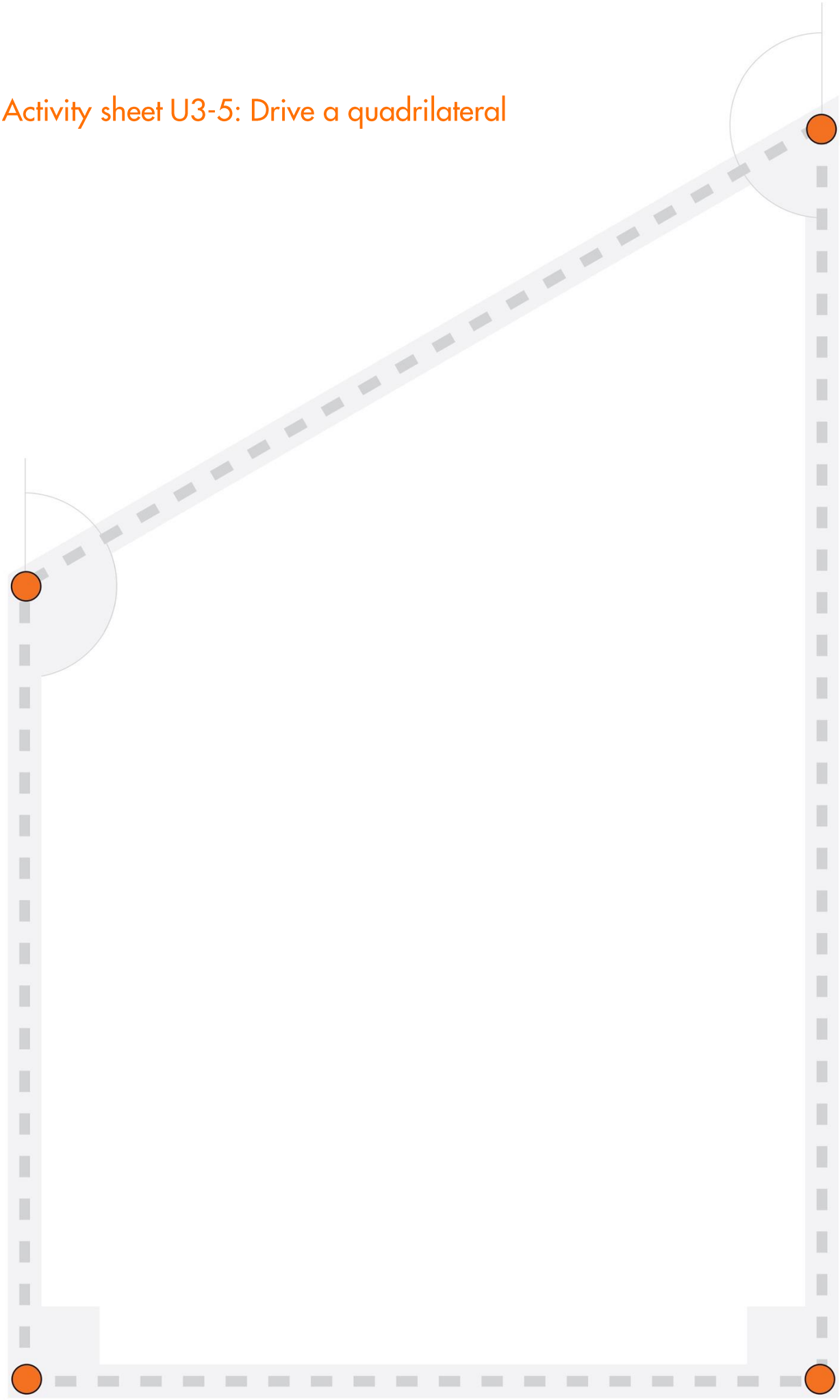
Activity sheet U3-3: Drive a hexagon



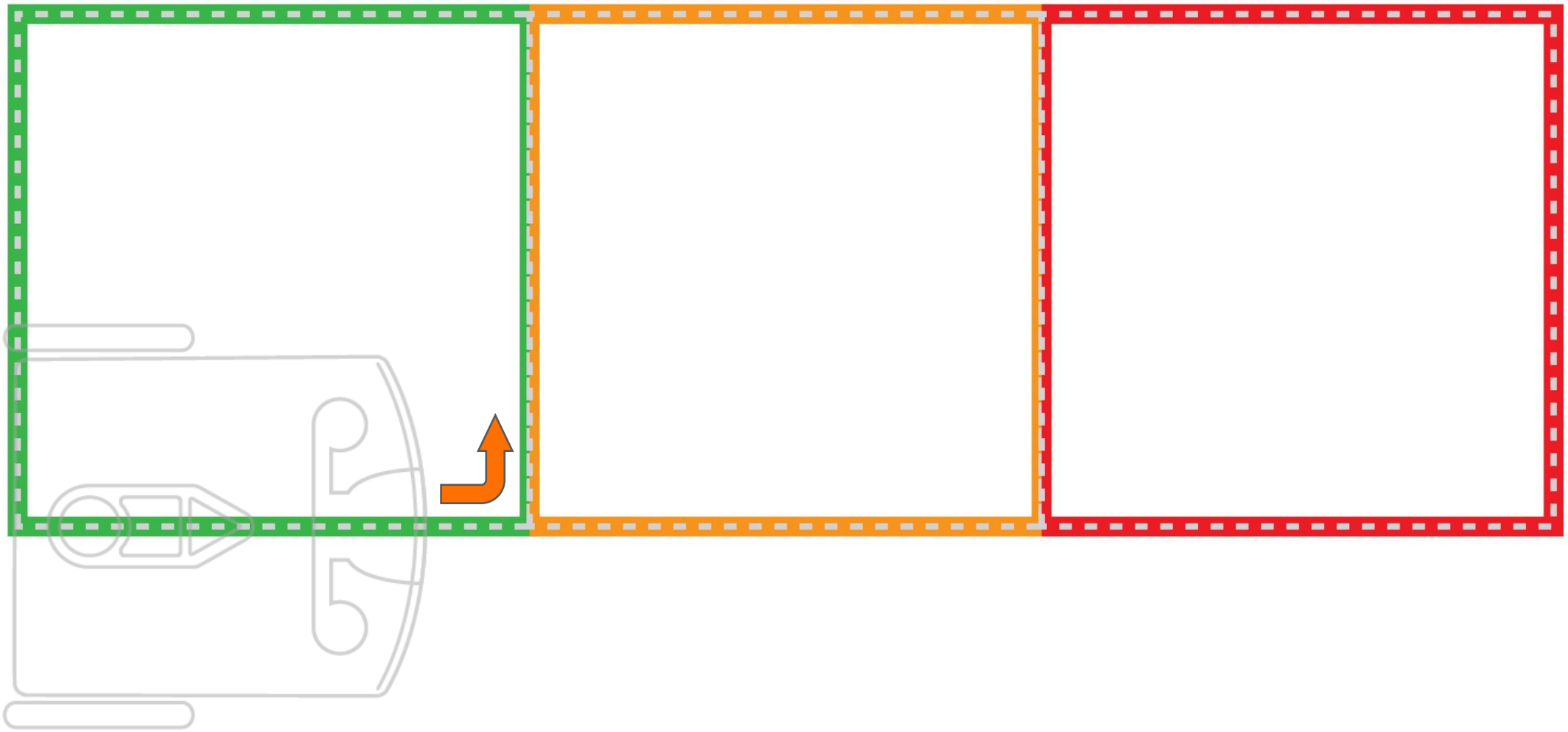
Activity sheet U3-4: Drive a circle



Activity sheet U3-5: Drive a quadrilateral



Activity sheet U3-6: Repeating squares



Activity sheet U3-7: Driving designs