

GOING THE DISTANCE

Build a car without any sensors that can drive as close as possible to a LEGO minifig a specified distance away without knocking it over



Think Like an Engineer:

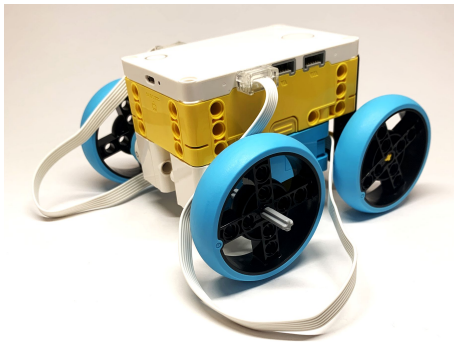
How can you design your car to produce repeatable data?



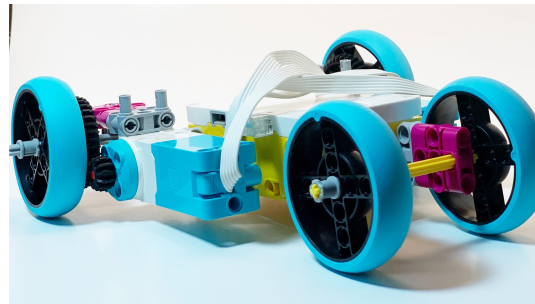
Think Like a Scientist:

How will you collect data in order to predict how far your car will drive in a set amount of time?

EXAMPLE IDEAS



Simple Car



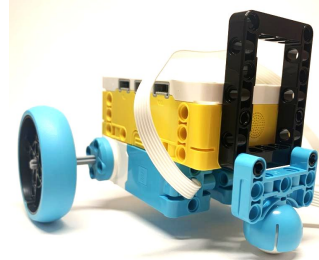
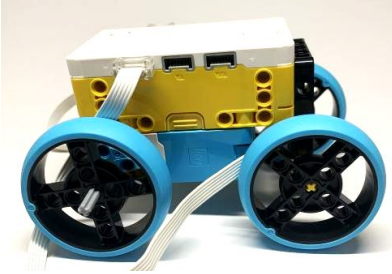
Geared Down Car



Car With a View

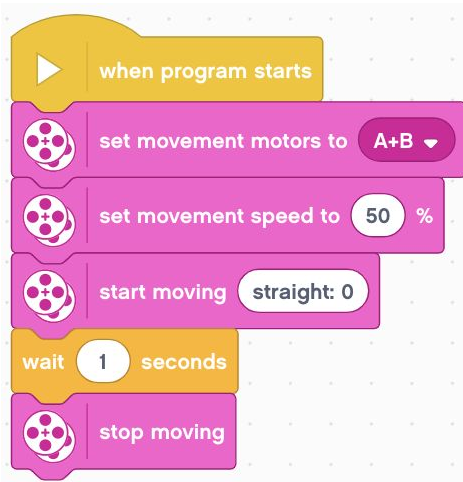
Flip over for more details!

BUILD IT!



- Think about where you want your wheels to be placed compared to the hub
- How can you build the robot so it could always move straightly and stop accurately

CODE IT!



Try using Python!

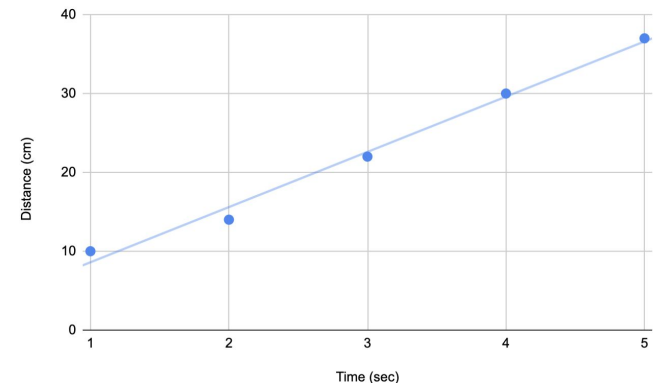
Try changing the speed, what impact does speed have on accuracy?

How can you adjust your code to help collect data?

You can only use time or rotation blocks for this challenge!

GRAPH IT!

Start collecting data. Measure how far your car goes in a set amount of time or rotation and graph it



How can you use this graph to help you?



CHALLENGE YOURSELF

Can you try to measure how far the robot moves with distance sensor?

GOING THE DISTANCE

Build a car without any sensors that can drive as close as possible to a LEGO minifig a specified distance away without knocking it over



Think Like an Engineer:

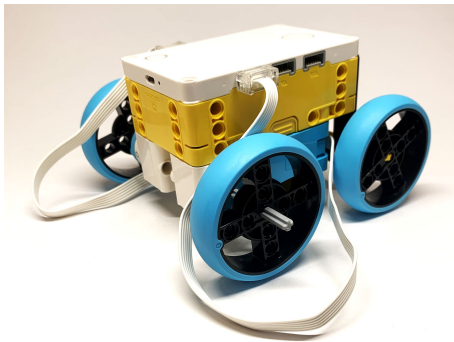
How can you design your car to produce repeatable data?



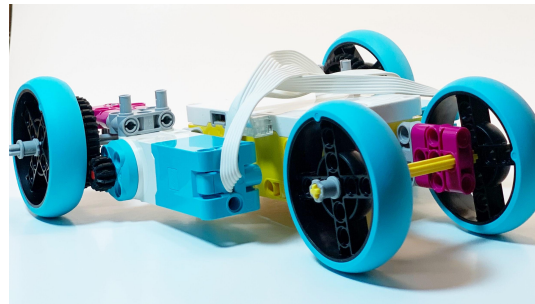
Think Like a Scientist:

How will you collect data in order to predict how far your car will drive in a set amount of time?

EXAMPLE IDEAS



Simple Car



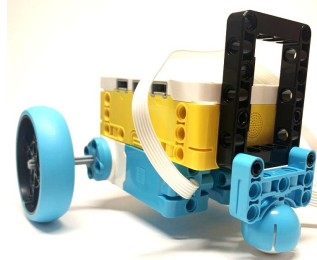
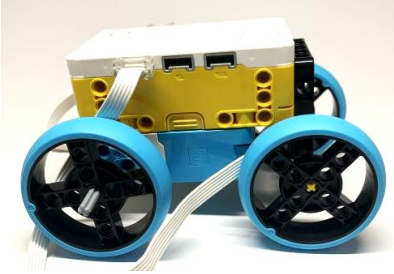
Geared Down Car



Car With a View

Flip over for more details!

BUILD IT!



- Think about where you want your wheels to be placed compared to the hub
- How can you build the robot so it could always move straight and stop accurately?

CODE IT!

How can you adjust your code to help collect data?

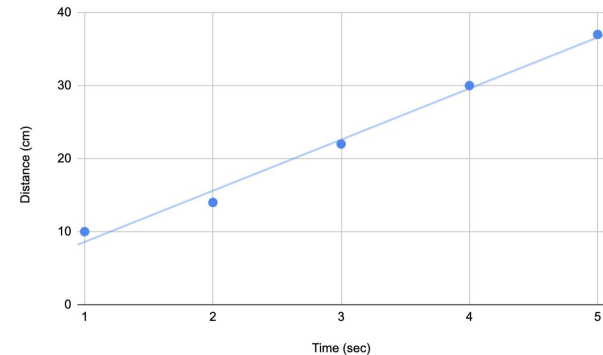
Try using the python “move” action!

```
14 motor_pair = MotorPair('B', 'A')
15 #start the car
16 motor_pair.start()
17 #wait for a designated amount of time
18 wait_for_seconds(3)
19 #stop the car
20 motor_pair.stop()
```

Remember, for this challenge you can't use any distance functions!

GRAPH IT!

Start collecting data. Measure how far your car goes in a set amount of time or rotation and graph it



How can you use this graph to help you?



CHALLENGE YOURSELF

Can you code with Python to generate the linear model automatically with two data points?

GOING THE DISTANCE: REINFORCEMENT LEARNING

Train a robot car with reinforcement learning so it can drive and stop at a specified distance



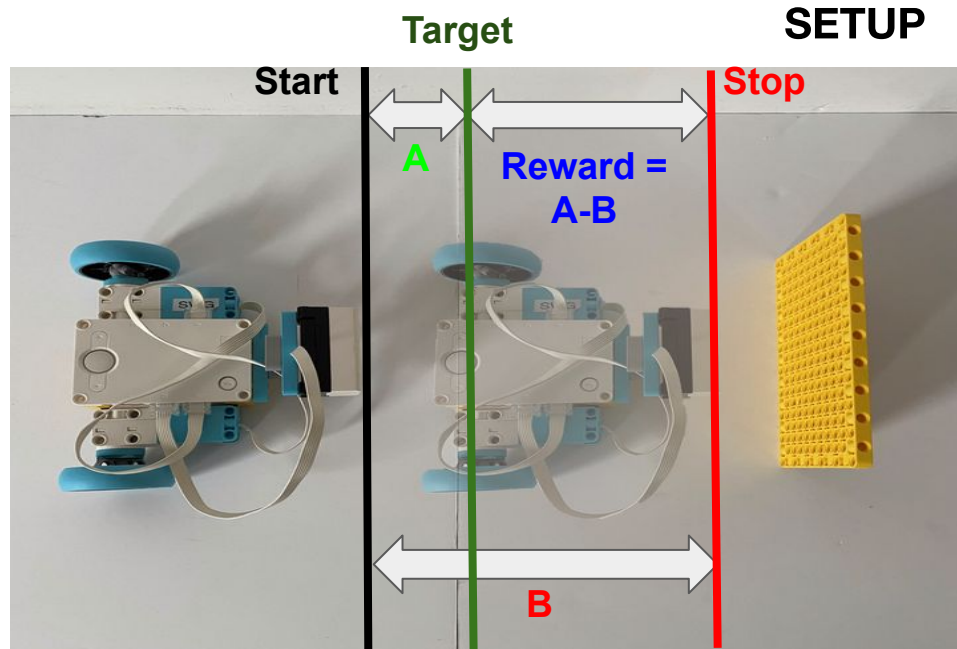
Think Like an Engineer:

How can you build your car so it could always move in a straight line?



Think Like a Scientist:

How can you interpret human feedback to a language that robot could understand?



Big picture idea:

Pick a random time, drive forward for that time, how close were you to the target?

Estimate a reward and try again by driving for a new time adjusted by the reward:

Flip over for more details!

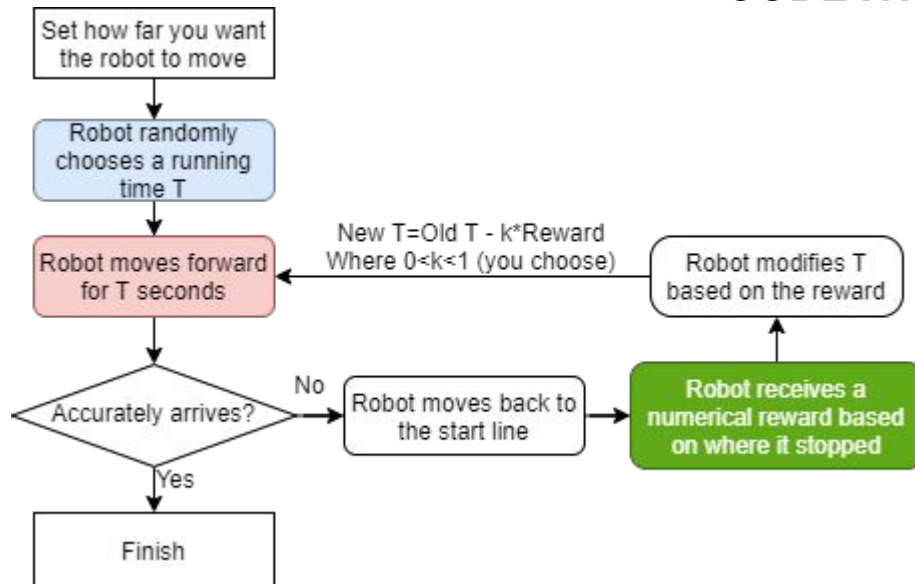
HELPFUL DEFINITIONS

Reinforcement learning: An AI training method for teaching a robot to do a task using a reward system

Action: What the robot does in one attempt to achieve the goal (e.g. move T seconds to try and get to the target position)

Reward: Feedback from a sensor after an action (attempt) to help teach the robot to make better attempts in the future"

CODE IT!



```
'''Randomly choose an initial running time T'''
import random
# Randomly choose an initial T within range 1-3
T = random.randrange(1, 3)
```

```
'''Drive forward for 2 sec'''
drive_base = MotorPair('A','B')
# Negative 'speed' could move the car backwards
drive_base.move(time=1, unit='seconds', speed=50)
```

```
'''Measure reward (assume you want move 20 cm)'''
ds = DistanceSensor('E')
# Read distance sensor value (cm) at beginning
start_pos = ds.get_distance_cm()
/robot moves forward/
# Read sensor value after makes a try
stop_pos = ds.get_distance_cm()
reward = start_pos - 20 - stop_pos
```



CHALLENGE YOURSELF

Tweak the variable k, see how it affects the training result.