

Śmieszne spacery

Zbuduj robota, który porusza się do przodu, mimo że nie ma kół.



Myśl jak inżynier:

Jak wprawić w ruch robota, który nie ma kół?

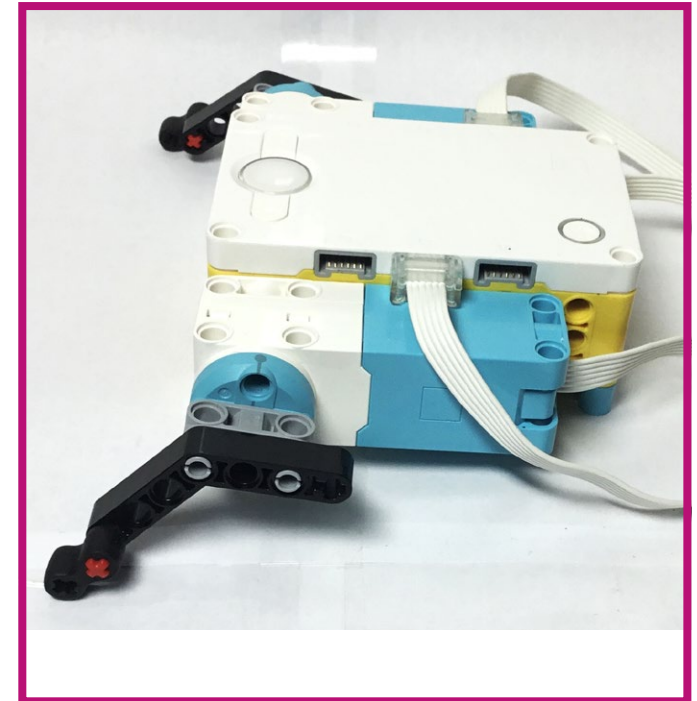
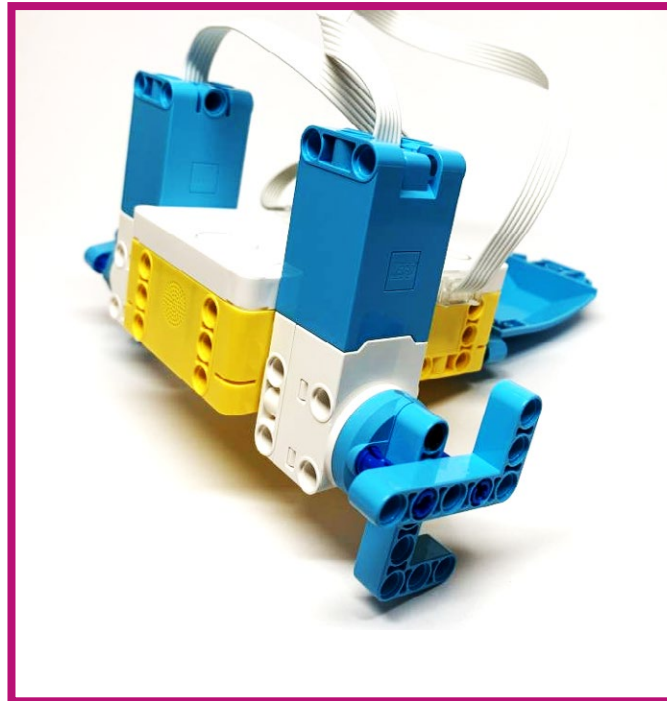
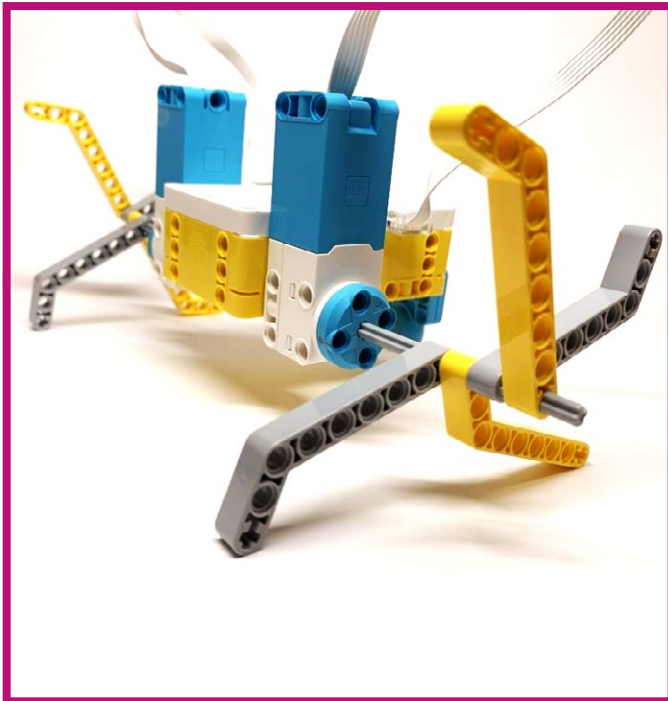
Myśl jak fizyk:

Jaki wpływ będzie miała długość nóg robota na jego ruchy?



Przykładowe pomysły

Na jakie różne sposoby można połączyć „nogi” robota z silnikiem?



Ciąg dalszy z drugiej strony!



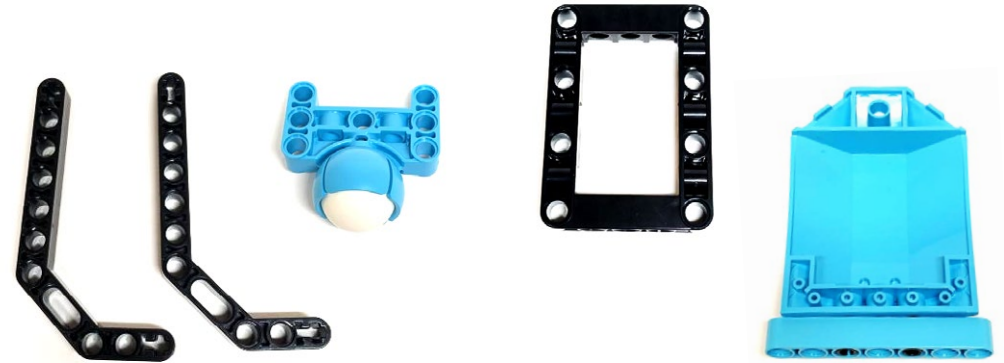
Czas na budowanie!

Elementy LEGO® mają różne zastosowania. Popisz się kreatywnością! Wybieraj nieoczywiste elementy i wymyślaj odważne projekty.

Jako nóg możesz użyć tych elementów:



Te sprawdzą się jako stabilizatory albo podpory:



Czas na programowanie!

Spróbuj zaprogramować robota tak, aby zatrzymywał się między obrotami. Jak szybko ma się poruszać Twój robot?

```
1 from hub import port
2 import runloop, motor_pair, time
3
4 motor_pair.unpair(motor_pair.PAIR_1)
5 motor_pair.pair(motor_pair.PAIR_1, port.A, port.B)
6
7 async def main():
8     for i in range(10):
9         await motor_pair.move_for_degrees(motor_pair.PAIR_1, 100, 0, velocity=500)
10        time.sleep_ms(1000)
11
12 runloop.run(main())
```

Modyfikacje:

- Dodaj czujnik, żeby Twój robot nie wszedł w ścianę.
- Czy chcesz, żeby silniki poruszały się razem, czy jeden po drugim?



Zadanie specjalne!

Spróbuj zbudować robota, który śmiesznie chodzi, używając tylko jednego silnika.

Czołem, Ziemiańskie

Zbuduj robota, który na powitanie macha do ludzi, przybija żółwika lub piątkę albo wykonuje inny gest.



Myśl jak inżynier:

Jak ustawić silnik, aby ruchy ramienia wyglądały bardziej naturalnie?

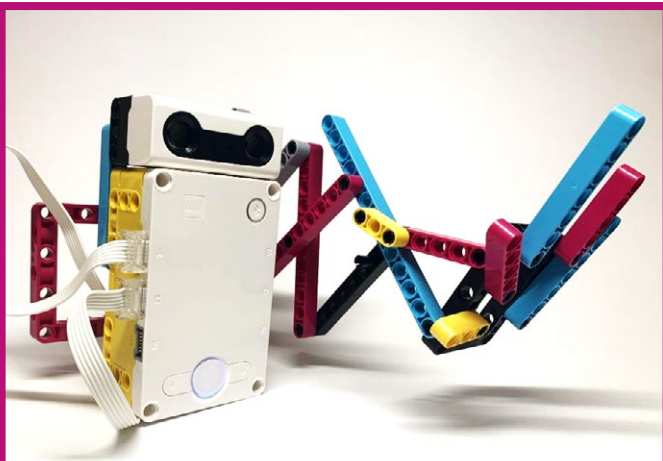
Myśl jak artysta:

Jak zbudować robota, który będzie wyglądał przyjaźnie i miło?



Przykładowe pomysły

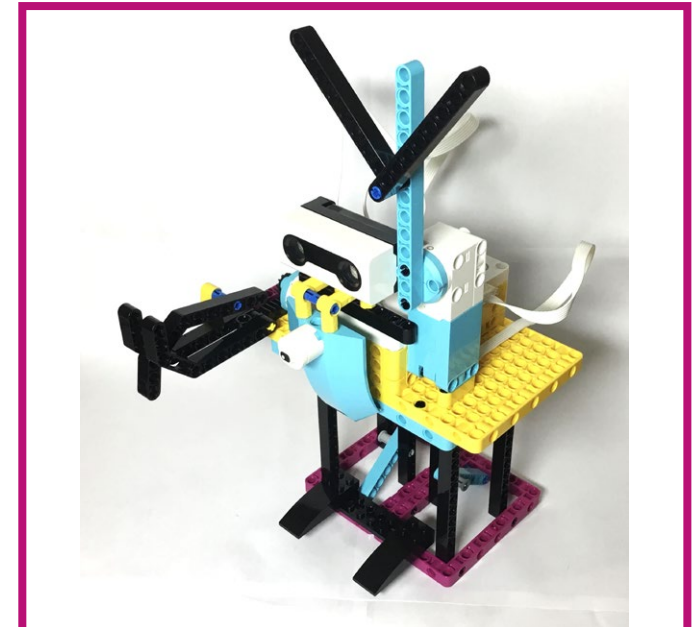
Jak robot ma się witać?



Robot przybijający piątkę



Robot machający



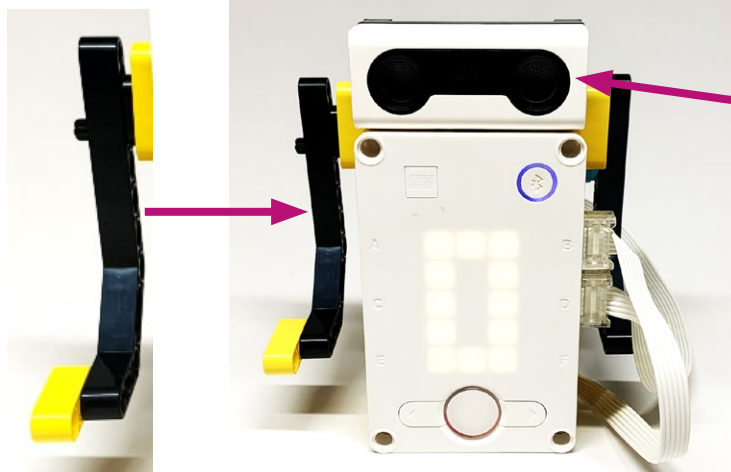
Robot przybijający żółwika

Ciąg dalszy z drugiej strony!



Czas na budowanie!

Z jakich elementów będzie się składał Twój robot?



Na końcu ramienia może mieć element, służący za dłoń.



Czujnik odległości wygląda jak oczy i pozwala na zautomatyzowanie ruchu.



Dłoń z „palcami” przypomina prawdziwą.



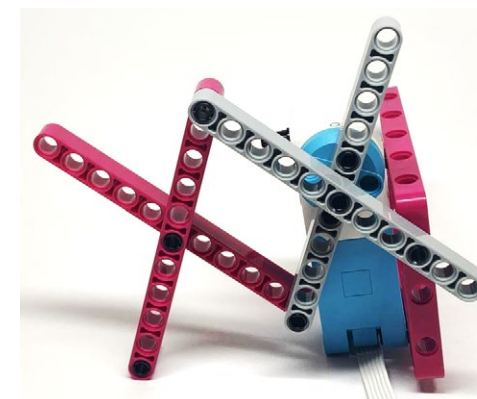
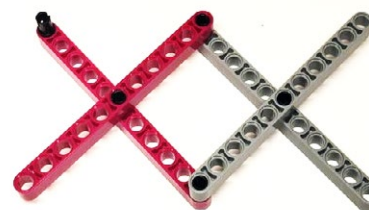
Czas na programowanie!

Przemyśl, jaki zakres ruchu ma mieć Twój robot. Czy ruch chcesz odmierzać w czasie czy w obrotach?

```

1 from hub import port
2 import runloop, motor
3
4 async def main():
5     for i in range(10):
6         await motor.run_for_degrees(port.A, 180, 500)
7         await motor.run_for_degrees(port.A, -180, 500)
8
9 runloop.run(main())

```



Połącz ze sobą elementy w kształcie litery „X”, a powstaną „nożyce” wydłużające ramię.



Zadanie specjalne!

Żeby robot poruszał się tylko, kiedy wykryje kogoś przed sobą, można mu zamontować czujnik odległości.

Przedstawienie kukiełkowe

Stwórz dowolną kukiełkę, która będzie się poruszać dzięki systemowi SPIKE™ Prime.



Myśl jak inżynier:

Jak przetestujesz swój projekt kukiełki i jakie ulepszenia wprowadzisz?

Myśl jak biolog:

Jakie stworzenia możesz zbudować i jak będą się poruszać?

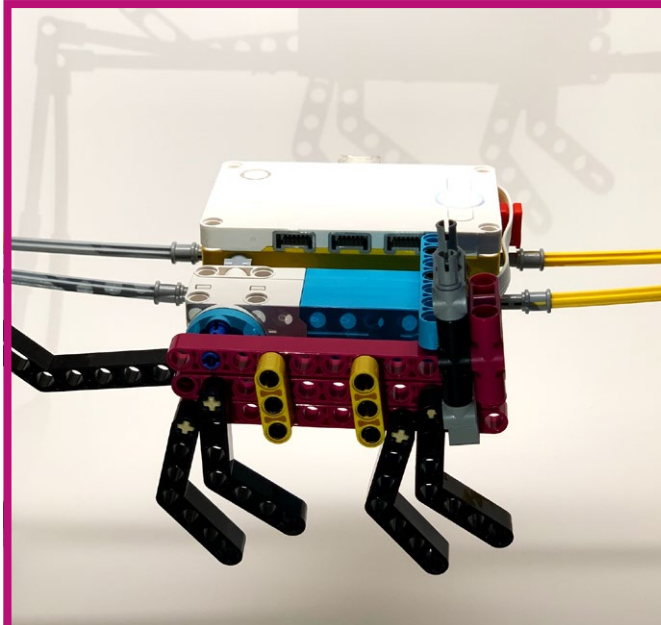


Przykładowe pomysły

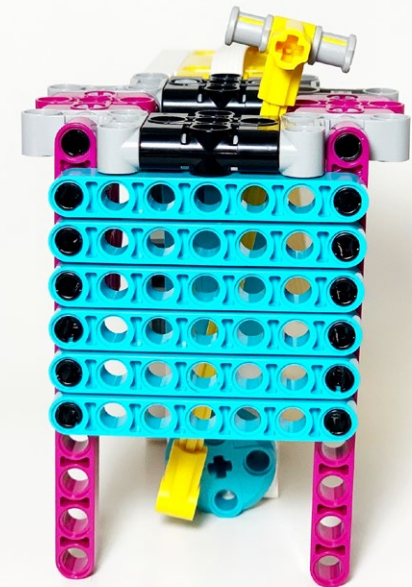
Poniżej jest kilka pomysłów na kukiełki. Jakie jeszcze przyjdą Ci do głowy?



Marionetka w kształcie człowieka



Lalka cieniowa w kształcie kota



Marionetka w kształcie węża w koszu

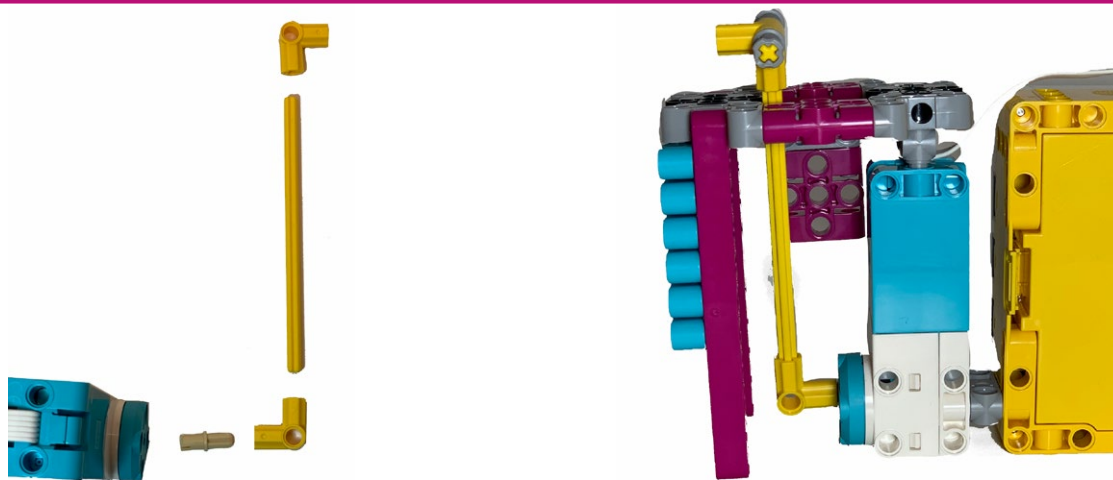
ciąg dalszy z drugiej strony!





Czas na budowanie!

Zastanów się, jak można zmienić ruch okrężny na liniowy.
Spróbuj stworzyć swoje ulubione zwierzę.



Czas na programowanie!

Fragment kodu do tańczącej marionetki:

```
1 from hub import port
2 import runloop, motor
3
4 async def main():
5     count = 0
6
7     while count < 10:
8         await motor.run_for_time(port.D, 500, -300)
9         await motor.run_for_time(port.D, 1000, 300)
10        await motor.run_for_time(port.D, 1000, 300)
11
12        count += 1
13
14 runloop.run(main())
```

Modyfikacje:

- Czy potrafisz tak zaprogramować swoją marionetkę, aby wykonywała inne ruchy po naciśnięciu różnych przycisków?
- Czy Twoja zabawka może w tym samym czasie poruszać się i wydawać dźwięki?
- Jak zmiana prędkości wpływa na ruch marionetki?



Zadanie specjalne!

Czy potrafisz zbudować marionetkę, która będzie się poruszać w rytm muzyki?

Ogród

Jak się ma Twój ogród? Pokaż nam, co masz w ogródku: kwitnące kwiatki, rosnące warzywa, a może robotyczne grabie?



Myśl jak ogrodnik:

Jakie rośliny i przedmioty można znaleźć w ogrodzie?

Myśl jak inżynier:

Jak za pomocą kilku silników możesz wprowadzić ruch do swojego ogrodu?



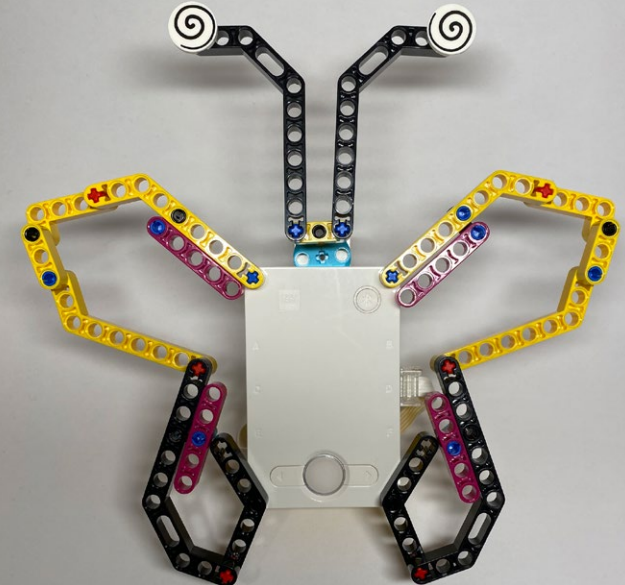
Przykładowe pomysły



Kwitnące kwiaty



Urocza budka dla ptaków



Urzekający motyl

Ciąg dalszy z drugiej strony!



Czas na budowanie!

Zbuduj realistyczną roślinę lub przedmiot, czerpiąc inspirację z kształtów klockowych elementów.



← Płatki słonecznika są zbudowane z prostych i zakrzywionych żółtych elementów →



Czas na programowanie!

Fragment kodu do czułek motyla:

```
1 from hub import port
2 import runloop, motor
3
4 async def main():
5     initial_power = 100
6     rotation = 0.5
7     for i in range(10):
8         await motor.run_for_degrees(port.A, int(rotation * 360), initial_power)
9         await motor.run_for_degrees(port.A, int(rotation * -360), initial_power)
10
11 runloop.run(main())
```

Modyfikacje:

- Każdy model jest inny i potrzebuje innych wartości obrotów silnika.
- Jeśli ustawisz różne prędkości silnika (na początku albo we fragmentach kodu), Twoje rośliny i przedmioty będą się poruszać bardziej dynamicznie.
- Zmieniaj wzór ze świateł na swoim klocku.



Zadanie specjalne!

Czy potrafisz sprawić, żeby Twój ogrodowy model poruszał się po zadziałaniu czujnika?

Prosty samochód SPIKETM

Zbuduj solidny samochód z jak najmniejszej liczby elementów. Twój robot musi umieć jechać w przód i w tył przez dwie sekundy oraz przez taki sam czas kręcić się w prawo i w lewo.



Myśl jak inżynier:

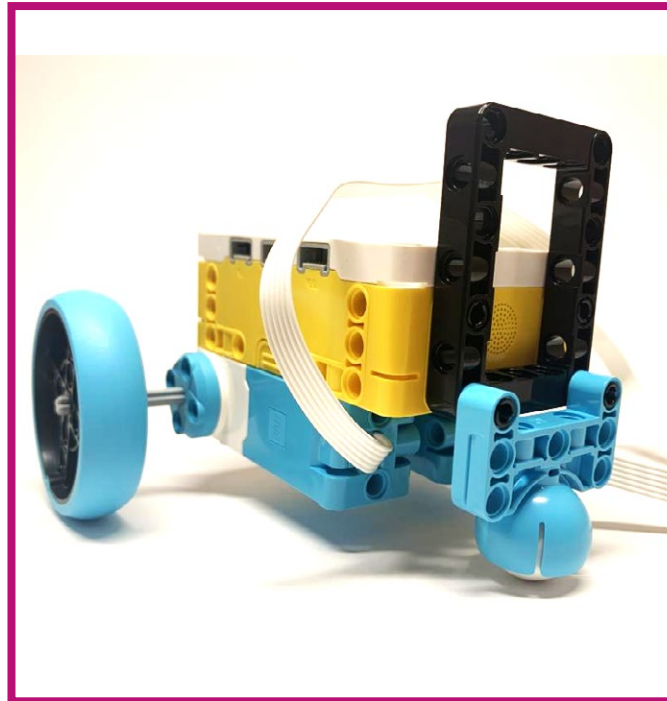
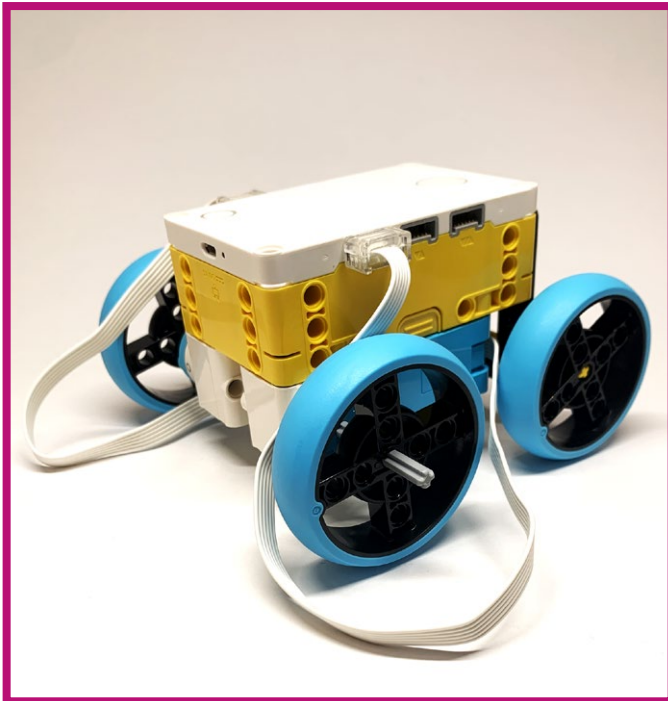
Eksperymentuj do woli, sprawdzając, co działa a co nie!

Myśl jak architekt:

Czy grupę elementów można zastąpić ich mniejszą liczbą?



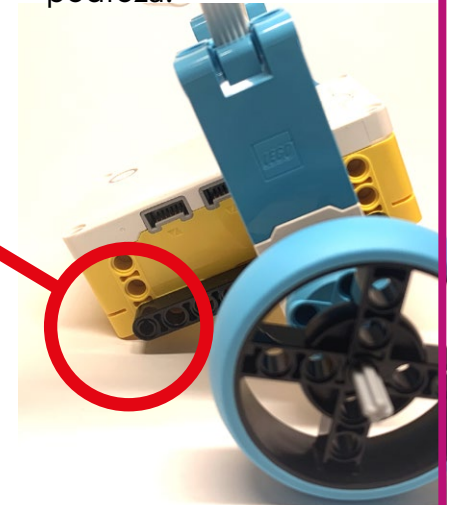
Przykładowe pomysły



Zadbaj o równowagę

Pamiętaj, że robot musi utrzymywać równowagę i tylko jego koła mogą dotykać podłoża.

Tego nie rób!

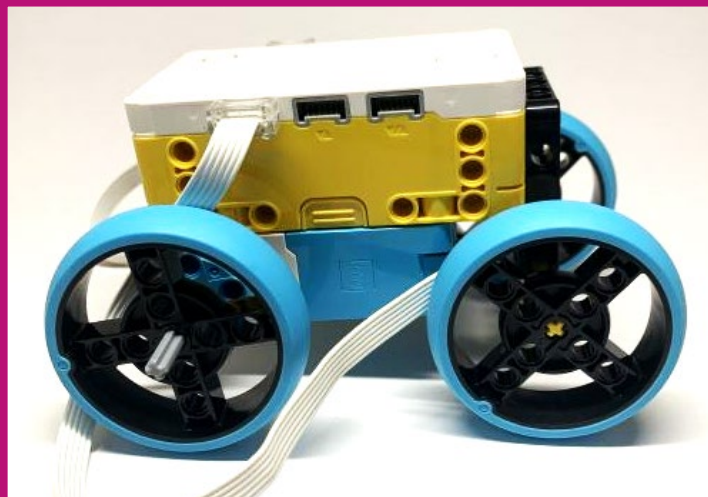


Ciąg dalszy z drugiej strony!

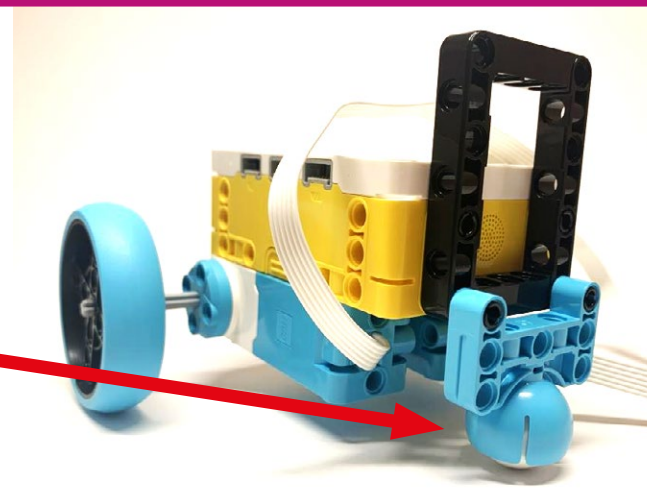


Czas na budowanie!

Na jakie różne sposoby mogą być zbudowane Twoje auta?



Dzięki tej kulce jezdnej pojazd może skręcać w dowolnym kierunku.



Czas na programowanie!

To jest fragment kodu poruszający dwa silniki w przód:



```
1 from hub import port
2 import runloop, motor_pair
3
4 motor_pair.unpair(motor_pair.PAIR_1)
5 motor_pair.pair(motor_pair.PAIR_1, port.A, port.B)
6
7 async def main():
8     await motor_pair.move_tank_for_time(motor_pair.PAIR_1, 500, 500, 2000)
9
10 runloop.run(main())
```

Modyfikacje:

Jak można go zmienić, żeby robot jechał do tyłu i skręcał?



Zadanie specjalne!

Zaprogramuj robota tak, żeby kreślił ósemki.

Perkusyjny popis

Bam-bam, ram-tam-tam! Zbuduj instrument perkusyjny dla zespołu. Perkusja, cymbałki, ksylofon albo dzwonki – każdy instrument perkusyjny jest mile widziany.



Myśl jak inżynier:

Jak ulepszysz swój projekt, aby dźwięk był jeszcze głośniejszy?

Myśl jak muzyk:

Jakie znasz instrumenty perkusyjne?



Przykładowe pomysły

Może do generowania dźwięku użyjesz innych elementów niż silniki?



Grzechotka



Bęben



Ksylofon

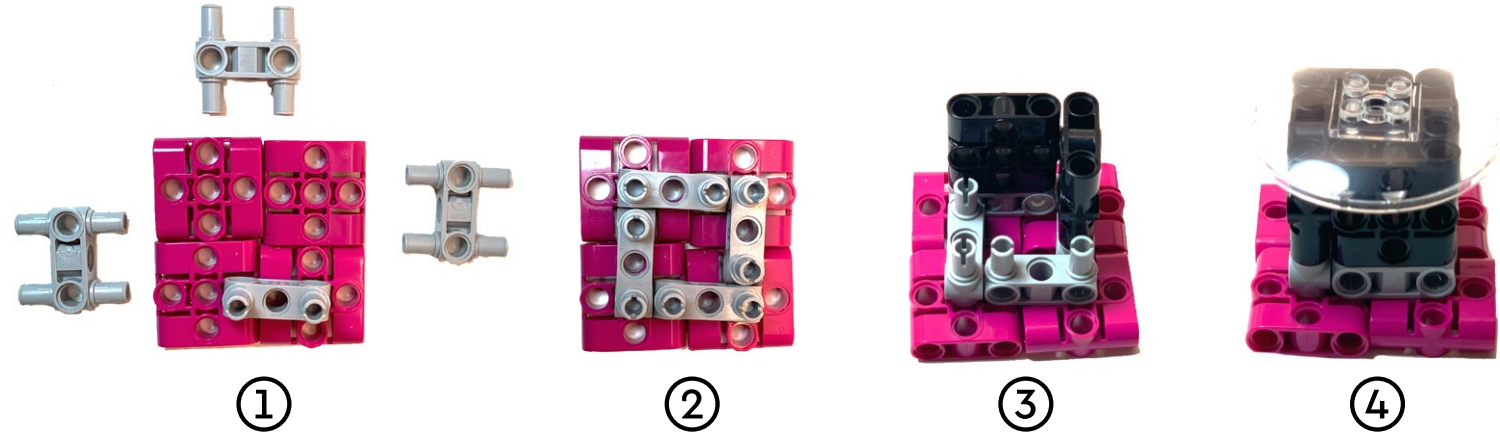
Ciąg dalszy z drugiej strony!



Czas na budowanie!

Zacznij od zbudowania prostego modelu (na przykład pudełka), a potem dodawaj do niego kolejne elementy (na przykład umieść w środku części LEGO, żeby powstała grzechotka).

Nawet prosty model można zbudować na bardzo wiele sposobów!



Czas na programowanie!

Kod, który sprawia, że grzechotka się trzęsie:

```
1 from hub import port, sound
2 import runloop, color_sensor, color
3
4 async def main():
5     while True:
6         colorSeen = color_sensor.color(port.E)
7
8         if colorSeen == color.BLACK:
9             await sound.beep(800, 500, 50)
10
11        if colorSeen == color.AZURE:
12            await sound.beep(1200, 500, 50)
13
14        if colorSeen == color.YELLOW:
15            await sound.beep(1400, 500, 50)
16
17 runloop.run(main())
```

Zastanów się!

- Spróbuj tak ustawić przerwy między potrząśnięciami, aby grzechotka grała do rytmu Twojej ulubionej piosenki.
- Jaki efekt będzie miała zmiana prędkości silnika?
- Jak można zmodyfikować kod, żeby grzechotka trzęsła się trochę dłużej lub krócej?



Zadanie specjalne!

Fajnie się śpiewa do rytmu, ale czy potrafisz zbudować instrument, który będzie grał konkretną melodię?

Odkrywanie kosmosu

Wyrusz na kosmiczną przygodę! Zbuduj statek kosmiczny, narzędzie dla astronauty albo nawet kosmitę – ta kosmiczna misja nie ma żadnych ograniczeń.



Myśl jak wynalazca:

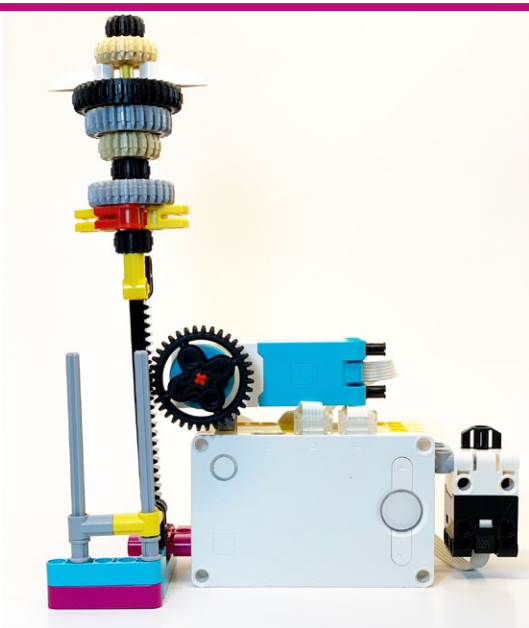
Jakie narzędzia przydałyby Ci się najbardziej w kosmosie?

Myśl jak odkrywca:

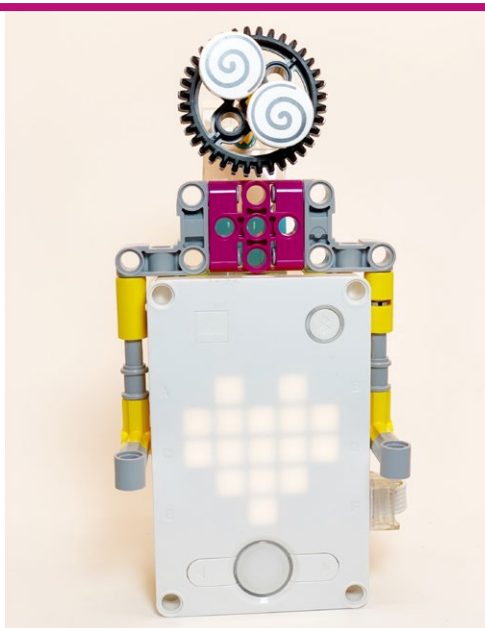
Na jakie stworzenia i środowiska można się natknąć na różnych planetach?



Przykładowe pomysły



Startująca rakietka kosmiczna



Kosmita z wirującą głową



Krążące planety

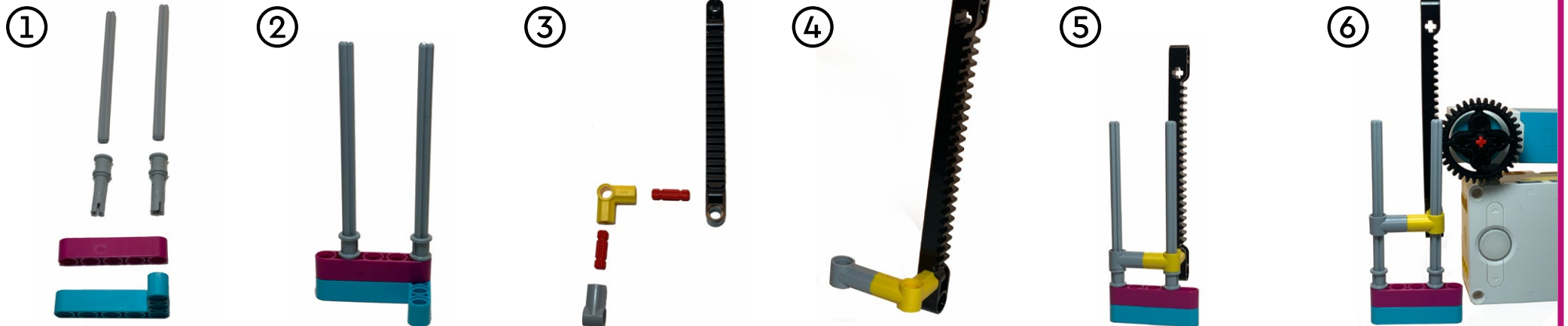
Ciąg dalszy z drugiej strony!





Czas na budowanie!

Za pomocą listwy zębatej (podłużny element) i koła zębatego (okrągły element) można zmienić ruch obrotowy w liniowy, tak jak w przykładowej rakiiecie kosmicznej. Koło zębate toczy się po listwie zębatej, poruszając nią w przód i w tył lub w górę i w dół.



Oto przykładowa podstawa, dzięki której listwa zębata gładko porusza się w górę i w dół.



Czas na programowanie!

Fragment kodu do poruszania kosmitą:

```
1 from hub import button, port, light_matrix
2 import runloop, motor
3
4 async def main():
5     while True:
6         if button.pressed(button.LEFT):
7             light_matrix.write("Blast Off!", 100, 250)
8             await motor.run_for_degrees(port.A, -90, 100)
9
10 runloop.run(main())
```

Modyfikacje:

- Czy potrafisz sprawić, żeby hub wydawał kosmiczne dźwięki?
- Czy potrafisz sprawić, żeby hub świecił?
- Spróbuj wykorzystać w swoim projekcie przycisk i go zakoduj.



Zadanie specjalne!

Czy potrafisz uzupełnić swój projekt o czujnik?

Przysłowia w stylu SPIKE™ Prime

Sowa huczy, choć nikt jej nie uczy. Co dwie głowy, to nie jedna. Zilustruj przysłowie dzięki systemowi SPIKE Prime. Wybierz swoje ulubione polskie przysłowie albo powiedzenie z innej części świata.



Myśl jak inżynier:

Jak dzięki silnikom i kołom zębatym można wykorzystać ruch do przedstawienia przysłowia?

Myśl jak pisarz:

Jakie przysłowie dobrze ilustruje jakąś sytuację z Twojego życia?



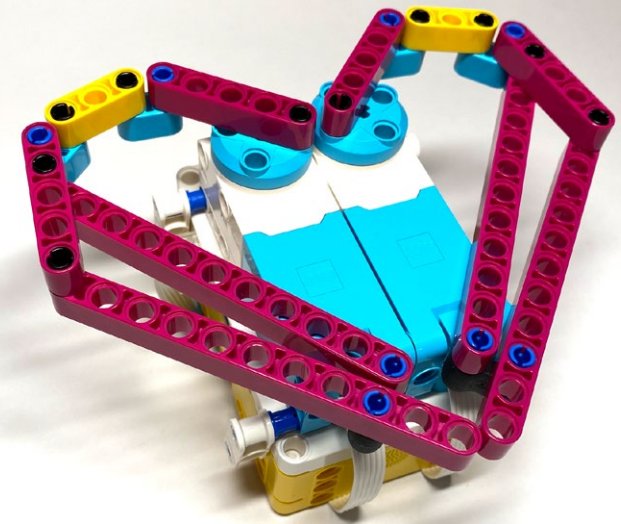
Przykładowe pomysły



Sowa huczy, choć nikt jej nie uczy.



Nie oceniaj książki po okładce.



Co z oczu, to z serca.

Ciąg dalszy z drugiej strony!





Czas na budowanie!

Zacznij od przejrzania dostępnych elementów i zastanów się, jak można ich użyć do zilustrowania przysłowia.



Czas na programowanie!

Fragment kodu do otwierania i zamykania książki:

```
1 from hub import port, light_matrix
2 import runloop, motor
3
4 async def main():
5     for i in range(10):
6         light_matrix.show_image(3)
7         await runloop.sleep_ms(2000)
8
9         await motor.run_for_degrees(port.A, int(0.15 * 360), 100)
10
11        light_matrix.show_image(1)
12        await runloop.sleep_ms(2000)
13
14        await motor.run_for_degrees(port.A, int(-0.15 * 360), 100)
15        light_matrix.show_image(3)
16
17 runloop.run(main())
```

Modyfikacje:

- Jak można użyć świateł na hubie do zilustrowania przysłowia?
- Czy w Twoim kodzie można użyć pętli warunkowej (polecenia „repeat”) albo instrukcji warunkowej (polecenia „if-then”) ? A może można użyć kilku pętli?



Zadanie specjalne!

Spróbuj tak zaprogramować swoje przysłowie SPIKETM Prime, żeby jego model wchodził w interakcję z użytkownikiem.

Wymiatacz

Każdy fan klocków LEGO® wie, że jakimś cudem te małe kawałki plastiku zawsze lądują na podłodze! Zbuduj zamiatarkę, która posprząta z podłogi klocki LEGO lub inne małe elementy LEGO. Wybór należy do Ciebie – może to być miotła, zgarniarka albo odkurzacz. Grunt, żeby urządzenie posprzątało podłogę.



Myśl jak inżynier:

Co możesz zmienić w swoim urządzeniu, żeby sprzątało mniejsze/większe powierzchnie?

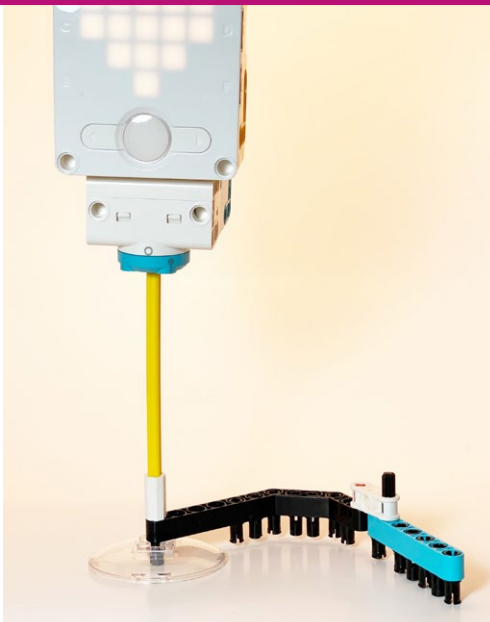
Myśl jak przedsiębiorca:

Czy potrafisz stworzyć urządzenie, które będzie chciał mieć każdy fan LEGO?

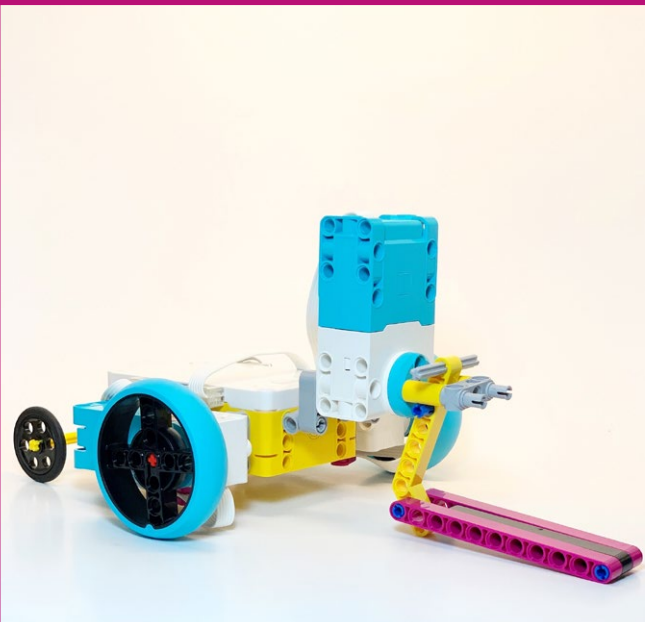


Przykładowe pomysły

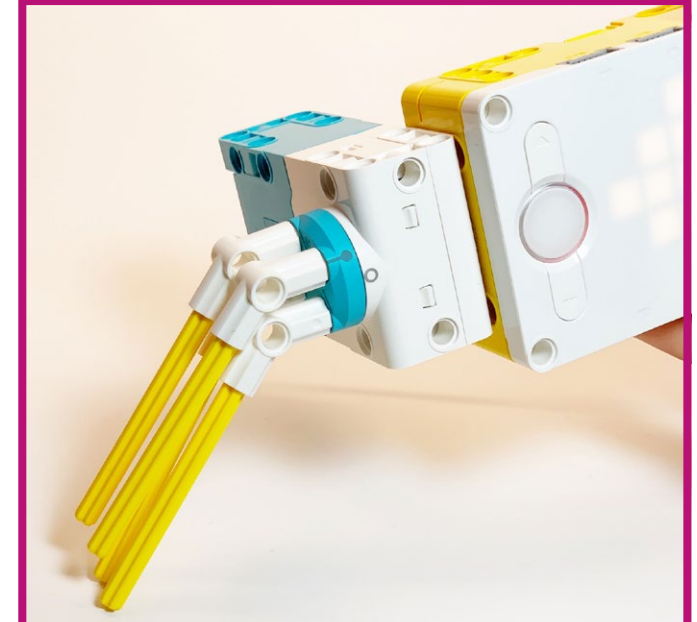
Czy urządzenie będzie trzeba trzymać w ręce, czy będzie samojezdne?



Miotła rotacyjna



Zamiatarka samojezdna



Miotła z napędem

Ciąg dalszy z drugiej strony!





Czas na budowanie!

Twój robot musi być stabilny.

①

Zadbaj o pewny element środkowy albo punkt podparcia, wokół którego robot będzie krążył.



②



③



Czas na programowanie!

Fragment kodu dla miotły rotacyjnej:

```
1 from hub import port, light_matrix
2 import runloop, motor
3
4 async def main():
5     await light_matrix.write("Beginning to Clean!", 100, 500)
6
7     while True:
8         await motor.run_for_time(port.A, 2000, 100)
9         await motor.run_for_time(port.A, 2000, -100)
10        await light_matrix.write("Still Cleaning!", 100, 500)
11
12 runloop.run(main())
```

Modyfikacje:

- Czy potrafisz zrobić tak, żeby robot zaczynał działać po wciśnięciu przycisku?
- Niech robot mówi do Ciebie podczas sprzątania.
- Jak prędkość obrotów silnika wpływa na jakość sprzątania?
- Co jeszcze może mówić/robić hub, żeby zasignalizować, że sprzątanie jest zakończone?



Zadanie specjalne!

Zmiatanie plastikowych części to jedno, ale czy potrafisz tak zaprogramować robota, żeby również zbierał plastikowe elementy?

Miotacz kulek

Zaprojektuj maszynę, która będzie jak najdalej wyrzucać małą plastikową kulkę.



Myśl jak sportowiec:

Jak rzucasz piłkę? Jak można wykorzystać tę wiedzę do zbudowania robota?

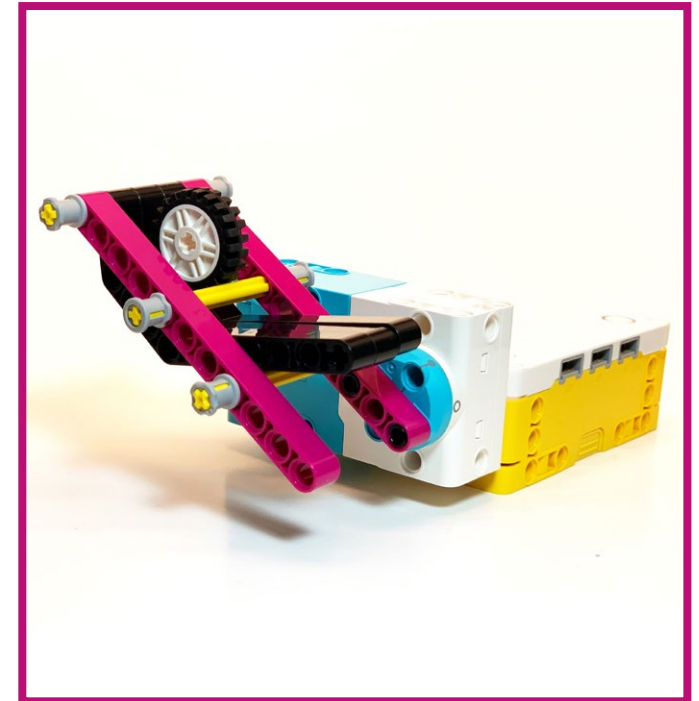
Myśl jak fizyk:

Jak wykorzystać dźwignię, aby kulka leciała jeszcze dalej?



Przykładowe pomysły

Możesz się wzorować na rzeczywistych urządzeniach, które miotają kulami (np. proca albo katapulta).



Ciąg dalszy z drugiej strony!





Czas na budowanie!

Oto przykład modelu, który działa jak ludzka ręka – przegub służy za „nadgarstek”.



Czas na programowanie!

Fragment kodu, dzięki któremu „ręka” wyrzuca kulkę:



```
1 from hub import port
2 import runloop, motor
3
4 async def main():
5     while True:
6         motor_A = port.A
7         await runloop.sleep_ms(2000)
8         await motor.run_to_absolute_position(port.A, 260, 300, direction=motor.SHORTEST_PATH)
9         await runloop.sleep_ms(1000)
10        await motor.run_for_degrees(motor_A, 72, 300)
11        await motor.run_to_absolute_position(motor_A, 260, 300, direction=motor.SHORTEST_PATH)
12
13 runloop.run(main())
```

Modyfikacje:

- Poeksperymentuj z prędkością ręki i sprawdź, co się stanie.
- Co się stanie, jeśli zwiększysz/zmniejszysz obroty?
- Zmień położenie wyjściowe „ręki” i sprawdź, jaki ma to wpływ na odległość wyrzucania kulki.



Zadanie specjalne!

Postaraj się, aby Twój robot miotał kulką na odległość równą Twojemu wzrostowi.