



Poznaj logikę jeśli, wtedy, w innym wypadku!



Odkryj nowe praktyczne kompilacje i możliwości programowania, aby pogłębić swoją wiedzę na ten temat.

# Kompletny wygląd robota Clawbot IQ



Gotowy Clawbot IQ.

Ten robot jest zaprojektowany tak, aby można go było szybko zbudować i by mógł poruszać się autonomiczne lub za pomocą kontrolera.

# Instrukcja budowania - napęd + czujnik odległości

### Podsumowanie instrukcji

#### Instrukcje dotyczące napędu + czujnika odległości (19 kroków):

- Prawe koło: kroki 1 do 6
- $\circ$  Lewe koło: kroki: 7 do 12
- o Czujnik odległości: kroki 13 do 19

#### Wskazówki dotyczące budowania dla wszystkich kroków:

- W sekcji na górze każdego z kroków są przedstawione ważne informacje związane z budową robota. Pierwsza liczba pod obrazem części (1x, 2x, 4x itd.) To numer tego elementu, którego będziesz potrzebować w tym kroku.Kolejną informacja jest rozmiar (jeśli jest to potrzebne) oraz nazwa potrzebnego elementu.
- Gotowy krok przedstawiono w ramce w prawym dolnym rogu.

o Przyglądaj się uważnie zielonym liniom, gdyż pokazują jak należy połączyć części.







Krok 1: Policz wszystkie elementy przed rozpoczęciem budowy i miej je dostępne pod ręką. Każdy członek zespołu powinien znaleźć elementy do swojego kroku..



Krok 2: Podczas dodawania wału podziałowego 4x, przekręć wał podziałowy, aby sprawdzić napięcie podczas obracania. Jeśli obraca się swobodnie, oznacza to, że nie jest prawidłowo włożony do silnika.







Krok 5: Przed zablokowaniem belki na miejscu upewnij się, że koła zębate pasują do siebie.



Krok 6: Po zamocowaniu kół, przekręć koło, które ma wał wchodzący w silnik. Jeśli koło obraca się swobodnie i bez naprężenia, wał wypadł ze swojego miejsca.





Krok 8: Podczas dodawania wałka przekręć go, aby sprawdzić napięcie podczas obracania. Jeśli obraca się swobodnie, oznacza to, że nie jest prawidłowo włożony do silnika.



Krok11: Przed zablokowaniem belki na miejscu upewnij się, że koła zębate pasują do siebie.



Krok 12: Po zamocowaniu kół, przekręć koło, które ma wał wchodzący w silnik. Jeśli koło obraca się swobodnie i bez naprężenia, wał wypadł ze swojego miejsca.



Krok 14: Upewnij się, że żyroskop jest umieszczony w prawidłowy sposób, aby umożliwić dostęp do kabla.









Krok 18: Pomarańczowe strzałki wskazują, że należy obrócić element.



Krok 19: Podczas mocowania czujnika odległości nie wciskaj żadnego z dwóch otworów osłoniętych siatką. Spowoduje to uszkodzenie czujnika. Upewnij się, że czujnik jest umieszczony w prawidłowy sposób, aby umożliwić dostęp do kabla.

## Instrukcje budowania - rama robota

### Podsumowanie instrukcji

#### Instrukcje budowy ram robota (22 kroki):

- o Cargo Holder: kroki 20 do 28
- Podstawa ramienia: kroki 29 do 41

#### Wskazówki dotyczące budowania dla wszystkich kroków:

- W sekcji na górze każdego z kroków są przedstawione ważne informacje związane z budową robota. Pierwsza liczba pod obrazem części (1x, 2x, 4x itd.) To numer tego elementu, którego będziesz potrzebować w tym kroku. Kolejną informacja jest rozmiar (jeśli jest to potrzebne) oraz nazwa potrzebnego elementu.
- o Gotowy krok przedstawiono w ramce w prawym dolnym rogu.

o Przyglądaj się uważnie zielonym liniom, gdyż pokazują jak należy połączyć części.

























Krok 31: Pomarańczowe strzałki wskazują, aby obrócić robota.



Krok 33: Upewnij się, że przełącznik zderzaka jest umieszczony w prawidłowy sposób, aby umożliwić dostęp do kabla.



Krok 36: Upewnij się, że inteligentny silnik jest ustawiony we właściwym kierunku (otwór na wałek znajduje się na dole).



Krok 38: Upewnij się, że dotykowy czujnik LED jest umieszczony w prawidłowy sposób, aby umożliwić dostęp do kabla.



Krok 39: Pomarańczowe strzałki oznaczają obrót robota.



Krok 40: Zamiast pokazania pojedynczych części, utworzone z nich elementy są pokazane w sekcji u góry. Podczas dodawania zespołu Krok 37, przekręć wał podziałowy, aby sprawdzić napięcie podczas obracania. Jeśli obraca się swobodnie, oznacza to, że nie jest prawidłowo włożony do silnika.



Krok 41: Zamiast pokazania pojedynczych części, utworzone z nich elementy są pokazane w sekcji u góry. Pomarańczowe strzałki wskazują, że należy obrócić element.

## Instrukcje budowania - Ramię

## Podsumowanie instrukcji

#### Instrukcja budowy ramienia (19 kroków):

Ramię: kroki 42 do 60

#### Wskazówki dotyczące budowania:

W sekcji na górze każdego z kroków są przedstawione ważne informacje związane z budową robota. Pierwsza liczba pod zdjęciem części (1x, 2x, 4x, itd.) oznacza ile elementów danego rodzaju potrzebujesz w tym kroku. Kolejną informacja jest rozmiar (jeśli jest to potrzebne) oraz nazwa potrzebnego elementu.

Gotowy element przedstawiono w ramce w prawym dolnym rogu.

• Przyglądaj się uważnie zielonym liniom, gdyż pokazują jak należy połączyć części.

















Krok 48: Upewnij się, że koła zębate są dobrze dopasowane, zanim przejdziesz do następnego kroku.



Krok 49: Obróć jeden z czarnych wałów na środku koła zębatego, aby upewnić się, że pasują do siebie i oba obracają się w tym samym czasie przed dodaniem płyty 4x4











Krok 56: Zamiast pokazania pojedynczych części, utworzone z nich elementy są pokazane w sekcji u góry.



Krok 59: Zamiast pokazania pojedynczych części, utworzone z nich elementy są pokazane w sekcji u góry.



Krok 60: Upewnij się, że inteligentny silnik jest ustawiony we właściwym kierunku (otwór na wał znajduje się po prawej stronie). Po dodaniu silnika obróć jedno z kół zębatych, aby sprawdzić napięcie podczas obracania. Jeśli obraca się swobodnie, oznacza to, że nie jest prawidłowo włożony do silnika.
# Instrukcja budowy - kleszcze

# Podsumowanie instrukcji budowania

- Claw Building Instructions (22 steps):
  - $_{\odot}$  Kleszcze: kroki 61 do 82
- Wskazówki do wszystkich kroków:

 Krok 60: Upewnij się, że inteligentny silnik jest ustawiony we właściwym kierunku (otwór na wałek znajduje się po prawej stronie). Po dodaniu silnika obróć jedno z kół zębatych, aby sprawdzić napięcie podczas obracania. Jeśli obraca się swobodnie, oznacza to, że nie jest prawidłowo włożony do silnika..

o Gotowy element przedstawiono w ramce w prawym dolnym rogu.

Przyglądaj się uważnie zielonym liniom, gdyż pokazują jak należy połączyć części.

















Krok 67: Upewnij się, że obręcz 100 mm jest dobrze dopasowana do rowka łącznika narożnego o szerokości 2 x ½..



Krok 68: Pomarańczowe strzałki wskazują, że należy obrócić element.















Krok 75: Upewnij się, że obręcz 100 mm jest dobrze dopasowana do rowka łącznika narożnego o szerokości 2 x  $\frac{1}{2}$ .



Krok 76: Pomarańczowe strzałki wskazują, że należy obrócić element.



Krok 77: Zamiast pokazania pojedynczych części, utworzone z nich elementy są pokazane w sekcji u góry.



Krok 78: Zamiast pokazania pojedynczych części, utworzone z nich elementy są pokazane w sekcji u góry.



Krok 80: Podłączany czujnik to czujnik koloru. Upewnij się, że czujnik jest umieszczony w prawidłowy sposób, aby umożliwić dostęp do kabla.



Krok 81: Zamiast pokazania pojedynczych części, utworzone z nich elementy są pokazane w sekcji u góry.



# Instrukcja budowania – montaż i

# okablowanie

## Podsumowanie instrukcji budowy

- Montaż i okablowanie (11 kroków):
  - Montaż końcowy: kroki od 83 do 93
  - Grupa jest również odpowiedzialna za upewnienie się, że czujniki i silniki są podłączone do właściwych portów za pomocą wyznaczonych kabli..
    - Port 1: Lewe koło
    - Port 2: dotykowy czujnik LED
    - Port 3: czujnik koloru
    - Port 4: żyroskop
    - Port 6: prawe koło
    - Port 7: czujnik odległości
    - Port 8: Przełącznik zderzaka
    - Port 10: silnik ramienia
    - Port 11: silnik kleszczy
- Wskazówki dotyczące budowania:
  - W sekcji na górze każdego z kroków są przedstawione ważne informacje związane z budową robota. Pierwsza liczba pod zdjęciem części (1x, 2x, 4x, itd.) oznacza ile elementów danego rodzaju potrzebujesz w tym kroku. Kolejną informacja jest rozmiar (jeśli jest to potrzebne) oraz nazwa potrzebnego elementu.
  - o Gotowy element przedstawiono w ramce w prawym dolnym rogu.
  - o Przyglądaj się uważnie zielonym liniom, gdyż pokazują jak należy połączyć części.
- Wskazówki dotyczące kroków 87-89:

Liczby na zielonym tle oznaczają numer portu, do którego zostanie podłączony kabel. Obrysowana na zielono liczba wskazuje czujnik, do którego będzie podłączony kabel. Użyj wskazanego kabla dla każdego czujnika lub silnika. Podłączając inteligentne kable, upewnij się, że są schowane, aby nie blokować inteligentnych czujników ani nie zakłócać ruchu Clawbota.







Krok 83: Zamiast pokazania pojedynczych części, utworzone z nich elementy są pokazane w sekcji u góry.



Krok 84: Zamiast pokazania pojedynczych części, utworzone z nich elementy są pokazane w sekcji u góry.



Krok 85: Zamiast pokazania pojedynczych części, utworzone z nich elementy są pokazane w sekcji u góry.



Krok 86: Upewnij się, że Smart Radio jest dobrze wsunięte. Przed włożeniem upewnij się, że bateria robota jest ustawiona we właściwy sposób. Pomarańczowe strzałki wskazują, że należy obrócić element.



Krok 87: Kabel silnika ramienia można wsunąć pod Mózg i podłączyć do odpowiedniego portu (port 10).



Krok 89: Pomarańczowe strzałki wskazują, że należy obrócić element.



Krok 90: Zamiast pokazania pojedynczych części, utworzone z nich elementy są pokazane w sekcji u góry. Pomarańczowe strzałki wskazują, że należy obrócić element.



Krok 91: Zamiast pokazania pojedynczych części, utworzone z nich elementy są pokazane w sekcji u góry. Pomarańczowe strzałki wskazują, że należy obrócić element.



Krok 92: Podczas dodawania wału podziałowego 8x, przekręć wał podziałowy, aby sprawdzić napięcie podczas obracania. Jeśli obraca się swobodnie, nie jest prawidłowo włożony do kół zębatych.



Krok 93: Liczby na zielonym tle oznaczają numer portu, do którego zostanie podłączony kabel. Obrysowana na zielono liczba wskazuje czujnik, do którego będzie podłączony kabel. Użyj wskazanego kabla dla każdego czujnika lub silnika. Podłączając inteligentne kable, upewnij się, że są schowane, aby nie blokować inteligentnych czujników ani nie zakłócać ruchu Clawbota.

# Analiza

Po zakończeniu budowy sprawdź, co potrafi robot. Sprawdź to, a następnie odpowiedz na poniższe pytania w swoim notatniku technicznym.

Jak zmieniłaby się prędkość ramienia, gdyby 60 zębowe koła zębate w krokach 55 i 58 zostały zmienione na mniejsze 36 zębowe koła zębate?



- Aby uzyskać pomoc przy tym pytaniu, spójrz na prędkość obrotową (RPM) kół zębatych z 12 zębami (z Kroku 37) i porównaj ją z prędkością obrotową koła zębatego o 60 zębach, delikatnie przesuwając ramię w górę i w dół.
- Obserwuj, ile razy obraca się koło zębate 12-zębowe w porównaniu do koła zębatego 60-zębowego. Jak myślisz, jak to się zmieni, gdy 60-zębowe koło zębate zostanie zastąpione 36-zębowym?
- Pamiętaj, aby uzasadnić swoją odpowiedź danymi z obserwacji.



Przetestuj swojego robota, obserwuj, jak działa, i wzmocnij swoją logikę i umiejętności rozumowania poprzez pomysłową, kreatywną zabawę.

# Podejmowanie decyzji



Drogi: prawda i fałsz

# Podejmowanie decyzji

Na najbardziej podstawowym poziomie programy są pisane w celu wykonywania prostych sekwencji zachowań. Na przykład możesz chcieć, aby robot jechał do przodu, a także wykonał kilka skrętów, aby dotrzeć do celu. Ale co, jeśli chcesz, aby Twój robot zaczekał na odpowiedni moment, aby rozpocząć jazdę do przodu i ukończyć trasę? To wymagałoby programowania z instrukcjami warunkowymi. Aby zdefiniować "właściwy czas na rozpoczęcie" projektu, należy użyć instrukcji warunkowej. Może "właściwy czas" jest po naciśnięciu przycisku lub gdy czujnik wykryje określony poziom i wtedy rusza. Kiedy obserwujesz zachowanie robota, wydaje się, że decyduje on, kiedy rozpocząć jazdę, ale dzieje się tak, ponieważ ustawiłeś warunek, w którym jazda powinna się rozpocząć.

Instrukcje warunkowe to instrukcje, które używają warunku logicznego (PRAWDA lub FAŁSZ) i umożliwiają tworzenie projektów, w których robot zachowuje się inaczej w zależności od tego, co wyczuwa.

W poniższym przykładzie, jeśli przycisk Brain Up zostanie naciśnięty (TRUE), robot będzie jechał do przodu. Jeśli przycisk Brain Up nie zostanie naciśnięty (FALSE), robot zatrzyma się. Pokazuje, że robot jedzie do przodu tylko po naciśnięciu przycisku Brain Up, w przeciwnym razie robot się zatrzyma.

whe	n star	rted				
fore	ver					
	$\overline{\langle}$	Brain	Up 🔻	button pres	ssed?	then
	drive	forwa	ard 🔻			
els	se					
	stop	driving				
	$\sim$					
	_					٦

# Programming with Conditionals in VEXcode IQ Blocks

Hardware/Software Required:

Amount	Hardware/Software
1	VEX IQ Super Kit (with up-to-date firmware)
1	VEXcode IQ Blocks (latest version, Windows, macOS, Chromebook, iPad)
1	Engineering Notebook
1	Clawbot (Drivetrain) Template

Clawbot jest gotowy do podejmowania decyzji!

To ćwiczenie dostarczy Ci umiejętności do zaprogramowania robota z zachowaniami warunkowymi. Głównym celem działania są bloki *if then* i *if then else*, ale używane są również bloki operujące i czujników.



Możesz skorzystać z informacji pomocy zawartych w blokach VEXcode V5, aby dowiedzieć się więcej o blokach. Wskazówki dotyczące korzystania z funkcji pomocy można znaleźć w samouczku *Korzystanie z pomocy.* 



## 1. Zacznijmy od zrozumienia pętli i instrukcji warunkowych.

Zanim zaczniesz programować z warunkami, najpierw obejrzyj samouczki wideo dotyczące używania pętli i Jeśli-to-w innym przypadku w blokach VEXcode IQ.

## 2. Programowanie warunkowe.

Chcemy zbudować projekt, który podniesie ramię, jeśli warunek naciśnięcia przycisku Brain Up jest prawdziwy.



Jeśli warunek jest fałszywy, silnik ramienia zatrzyma się. Przyjrzyjmy się budowaniu projektu, który używa bloku *na zawsze* i bloku warunkowego *jeśli to* do zaprogramowania ramienia.

• Otwórz szablon **Clawbot (Drivetrain).** Aby uzyskać pomoc, obejrzyj samouczek wideo dotyczący korzystania z przykładowych projektów i szablonów.



• Utwórz poniższy projekt.

forever			
if	Brain Up 💌	button pre	ssed? the
spin	ArmMotor 👻	up 🔻	
stop	ArmMotor 💌		

 Zapisz projekt jako ArmUp. Aby uzyskać pomoc, obejrzyj film instruktażowy nazywanie i zapisywanie projektu (Windows, Mac, iPad, Chrome) w VEXcode IQ Blocks.



- Sprawdź, czy nazwa projektu ArmUp znajduje się teraz w oknie na środku paska narzędzi.
- Przewiduj, co projekt zrobi Clawbot w twoim notatniku inżyniera. Wyjaśnij zachowanie użytkownika i Clawbota.
- Sprawdź, czy twoje przewidywania dotyczące tego, co ma zrobić Clawbot, są poprawne.
- Pobierz projekt do Slotu 1 na Clawbocie, a następnie uruchom go.
- Aby uzyskać pomoc dotyczącą pobierania i uruchamiania projektu, zobacz samouczek dotyczący bloków VEXcode IQ, który wyjaśnia, jak pobrać i uruchomić projekt.
- Sprawdź swoje wyjaśnienia dotyczące projektu i dodaj notatki, aby w razie potrzeby je poprawić.

#### 3. Zrozumienie bloku czekaj do.

W poprzednim kroku, stworzony projekt nie pozwolił na podniesienie ramienia. Zobacz poniższy schemat, który wyjaśnia przebieg tego projektu. Zauważ, że po naciśnięciu przycisku Brain Up przepływ projektu przesuwa się tak szybko, że projekt przesunie się dalej, do następnego bloku, którym jest blok *zatrzymaj silnik*.



W związku z tym projekt wymaga bloku *czekaj do*, który "każe" silnikowi ramienia obracać się do momentu zwolnienia przycisku Brain Up.



Blok *czekaj do* jest konieczny ze względu na szybkość przepływu projektu. Gdyby go tam nie było, projekt przesunąłby się do następnego bloku, zanim silnik ramienia zdążyłby zareagować. W ten sposób bloki spływałyby w dół do bloku *zatrzymaj silnik,* a następnie

projekt zaczynałyby się z powrotem na szczycie stosu z powodu bloku na zawsze, który powtarza wszystkie zawarte w nim bloki.

Przyjrzyjmy jak zmieni się projekt, gdy dodamy *blok czekaj do.* Silnik ramienia będzie teraz obracał się do momentu zwolnienia przycisku Brain Up. Po zwolnieniu przycisku Brain Up projekt będzie kontynuowany do następnego bloku, którym jest blok *zatrzymaj silnik*.

Projekt najpierw sprawdzi warunek, czy przycisk Brain Up jest wciśnięty. Jeśli przycisk Brain Up zostanie naciśnięty (TRUE), ramię będzie się obracać, aż przycisk Brain Up zostanie zwolniony. Po zwolnieniu przycisku Brain Up projekt przesunie się do bloku *zatrzymaj silnik*, a następnie wróci na szczyt stosu, aby rozpocząć projekt od nowa z powodu *bloku na zawsze*.

Jeśli przycisk Brain Up nie zostanie naciśnięty (FALSE), projekt przesunie się do bloku *zatrzymaj silnik*, zanim powróci na szczyt stosu, aby rozpocząć od nowa projekt z powodu bloku *na zawsze,* a ramię nigdy się nie obróci.



## 4. Dodanie bloku czekaj do

Dodajmy blok czekaj do:

• Dodaj blok czekaj do do swojego projektu ArmUp, aby wyglądał następująco:

if	Brain Up  button pressed?	
spin	ArmMotor 🔹 up 💌	
wait	until not Brain Up - button pressed?	

• Zapisz projekt jako ArmUp2. Aby uzyskać pomoc, obejrzyj film instruktażowy *nazywanie i zapisywanie projektu* (Windows, Mac, iPad, Chrome) w VEXcode IQ Blocks.



- Pobierz projekt do slotu 2 na Clawbocie, a następnie uruchom go.
- Aby uzyskać pomoc dotyczącą pobierania i uruchamiania projektu, zobacz samouczek dotyczący bloków VEXcode IQ, który wyjaśnia, jak pobrać i uruchomić projekt.
- Sprawdź, czy ramię się teraz obróci po naciśnięciu przycisku Brain Up.
- Sprawdź, czy gdy przycisk Brain Up nie jest wciśnięty (zwolniony), silnik ramienia zatrzymuje się.
- Zapisz swoje obserwacje, jak zachowywał się Clawbot przed i po dodaniu bloku czekaj do do projektu w swoim notatniku inżyniera.

# **Programming Two Buttons**

#### Hardware/Software Required:

Amount	Hardware/Software
1	VEX IQ Super Kit (with up-to-date firmware)
1	VEXcode IQ Blocks (latest version, Windows, macOS, Chromebook, iPad)
1	Engineering Notebook
1	Clawbot (Drivetrain) Template

The Clawbot arm is ready to move up and down!

This activity will give you the tools to program your robot with conditional behaviors. The *if then else* block is the main focus within the activity but other Sensing, Control, and Operator blocks are also used.



You can use the Help information inside of VEXcode IQ Blocks to learn about the blocks. For guidance in using the Help feature, see the Using Help tutorial.



## 1. Programowanie opuszczania ramienia.

Na poprzedniej stronie zaprogramowałeś ramię Clawbota tak, aby obracało się po naciśnięciu przycisku Brain Up. Ale co z obniżeniem ramienia? Wróćmy najpierw do poprzedniego projektu ArmUp2.

if	Brain Up  button pressed?	
$\sim$		
spin	ArmMotor • up •	
wait u	Intil not Brain Up - button pressed?	
~		

Czy byłoby możliwe użycie tego samego bloku *jeżeli to*, aby zablokować obracanie ramienia w górę i obrócić je w dół? Spróbujmy! Zbuduj projekt poniżej, edytując już istniejący projekt ArmUp2.

	_ E	Brain	Up 🔻	button pre	ssed?		
s	pin	ArmN	lotor 🔻	up 🔻			
		ntil 🧹	not CE	Brain Up	button p	ressed?	
	F	Brain	Down •	- button	pressed?		
s	pin	Arm№	lotor 🗢	down 🖣			
	vait ur	vtil 🤇	not E	Brain Dow	n 🔻 buttor	pressed?	

Zapisz projekt jako ArmUpDown. Aby uzyskać pomoc, obejrzyj film instruktażowy *nazywanie i zapisywanie projektu* (Windows, Mac, iPad, Chrome) w VEXcode IQ Blocks.



- Sprawdź, czy nazwa projektu ArmUpDown znajduje się teraz w oknie pośrodku paska narzędzi.
- W twoim notatniku inżynieryjnym zapisz przewidywania, co zrobi Clawbot . Wyjaśnij zachowanie użytkownika i Clawbota.
- Sprawdź, czy twoje przewidywania, są poprawne.
- Pobierz projekt na Slot 3 na Clawbocie, a następnie uruchom go.
- Aby uzyskać pomoc dotyczącą pobierania i uruchamiania projektu, zobacz samouczek dotyczący bloków VEXcode IQ, który wyjaśnia, jak pobrać i uruchomić projekt.
- Sprawdź swoje wyjaśnienia dotyczące projektu i dodaj notatki, aby w razie potrzeby je poprawić.

## 2. Zrozumienie przebiegu projektu.

W poprzednim kroku projekt podniósł i opuścił ramię. Jednakże, warunek naciśnięcia przycisku Brain Up występuje jako pierwszy, jeśli przycisk ten zostanie przytrzymany, a następnie naciśnięty zostanie również Brain Down, ramię będzie się dalej obracać, ponieważ czynność ta nie zostanie zatrzymana do momentu zwolnienia przycisku Brain Up. Prześledź następujący schemat blokowy, który wyjaśnia przebieg projektu.



W ten sposób projekt może zastąpić bloki *jeżeli to* blokami *jeżeli to w przeciwnym razie*, tak aby tylko jedno zdarzenie mogło być prawdziwe w danym momencie.

Obejrzyj poniższy film instruktażowy na temat, bloku jeżeli to w przeciwnym razie:

Użycie bloku *jeżeli to w przeciwnym razie* wyeliminuje potrzebę użycia bloku *czekaj do*, ponieważ ramię będzie się obracać do momentu zwolnienia przycisku Brain Up. Dzieje się tak, ponieważ element *"else - w przeciwnym razie"* nie zostaje spełniony, dopóki warunek przycisku Brain Up nie jest fałszywy (zwolniony).

f Brain Up	button pressed?	
spin ArmMotor	▼ up ▼	
else		
if Brain D	own ▼ button pressed? t	hen
spin ArmMot	or 👻 down 👻	
else		
stop ArmMot	or 🔻	_
Stop Anniviot		

Blok na zawsze pozwala na ciągłe sprawdzanie pierwszego warunku.

Przyjrzyjmy się zmianom w projekcie, dodając bloki *jeżeli to w przeciwnym wypadku*. Silnik ramienia będzie się teraz obracał, dopóki przycisk Brain Up będzie wciśnięty (nie będzie zwolniony). Ze względu na blok *na zawsze* warunek ten będzie nadal sprawdzany, dopóki nie będzie fałszywy.

Gdy stan przycisku Brain Up jest fałszywy, projekt będzie kontynuowany do następnego bloku, który ma za zadanie sprawdzić, czy przycisk Brain Down jest naciśnięty. Jeśli warunek ten jest prawdziwy, ramię zacznie się obracać. Ponownie, ze względu na blok *na zawsze*, warunek ten będzie nadal sprawdzany, dopóki nie będzie fałszywy.

Gdy stan przycisku Brain Up jest fałszywy, projekt będzie kontynuowany do następnego bloku, którym jest blok *zatrzymaj silnik.* Tak więc silnik ramienia zatrzyma się tylko wtedy, gdy oba warunki są fałszywe (żaden przycisk nie jest naciśnięty).



## 3. Programowanie z blokiem jeżeli to w przeciwnym razie.

Użyjmy bloku jeżeli to w przeciwnym razie:

Dodaj blok *jeżeli to w przeciwnym razie* do projektu ArmUpDown, aby wyglądał następująco:

f	Brain Up 🔹 button pressed?	then
spin	ArmMotor	
else		
if	Brain Down - button pressed? then	
spin	ArmMotor 👻 down 👻	
else		
stop	ArmMotor 🗢	

• Zapisz projekt jako ArmUpDown2. Aby uzyskać pomoc, obejrzyj film instruktażowy *nazywanie i zapisywanie projektu* (Windows, Mac, iPad, Chrome) w VEXcode IQ Blocks.



- Zapisz projekt w Slocie 4 na Clawbocie, a następnie uruchom go.
- Aby uzyskać pomoc dotyczącą pobierania i uruchamiania projektu, zobacz samouczek dotyczący bloków VEXcode IQ, który wyjaśnia, jak pobrać i uruchomić projekt.
- Sprawdź, czy ramię zacznie się teraz obracać po naciśnięciu przycisku "Brain Up" i w dół, gdy zostanie naciśnięty przycisk "Brain Down".
- Sprawdź, czy gdy przyciski Brain Up i Down nie są wciśnięte (są zwolnione), silnik ramienia zatrzymuje się.
- Zapisz swoje obserwacje, jak zachowywał się Clawbot przed i po dodaniu bloku jeżeli to w przeciwnym razie do projektu w notatniku inżyniera.

#### 4. Programowanie kleszczy

W poprzednim kroku projekt został zoptymalizowany, aby działał wydajniej, używając bloku jeżeli to w przeciwnym razie. W poprzednim przykładzie manipulowano ramieniem za pomocą przycisków Brain Up i Down.

Używając dokładnie tego samego zarysu projektu, możemy również manipulować kleszczami.

 Przejrzyj projekt ArmUpDown2 i zmień w bloku obróć ArmMotor na ClawMotor za pomocą menu rozwijanego. Upewnij się, że przycisk "Brain Up" powoduje "otwarcie" Kleszczy, a przycisk "Brain Down" obraca je "zamyka" za pomocą menu rozwijanych.

if Brain Up   button pressed?	
spin ClawMotor ▼ open ▼	
else	
if Brain Down ▼ button pressed? then	
spin ClawMotor ▼	
else	
stop ArmMotor 👻	
ClawMotor	

• Zapisz projekt jako ClawUpDown. Aby uzyskać pomoc, obejrzyj film instruktażowy nazywanie i zapisywanie projektu (Windows, Mac, iPad, Chrome) w VEXcode IQ Blocks.



- Zapisz projekt w Slocie 1 na Clawbocie, a następnie uruchom go.
- Aby uzyskać pomoc dotyczącą pobierania i uruchamiania projektu, zobacz samouczek dotyczący bloków VEXcode IQ, który wyjaśnia, jak pobrać i uruchomić projekt.
- Sprawdź, czy czy kleszcze otworzą się po naciśnięciu przycisku Brain Up i zamkną po naciśnięciu przycisku Brain Down.
- Sprawdź, czy gdy przyciski Brain Up i Down nie są wciśnięte (są zwolnione), silnik kleszczy zatrzymuje się.
- Zapisz swoje obserwacje, jak zachowywał się Clawbot przed i po dodaniu bloku *jeżeli to w* przeciwnym razie do projektu w notatniku inżyniera.notebook.



Rozwiąż inne problemy XXI wieku, stosując podstawowe umiejętności i koncepcje, których się nauczyłeś.
## Interfejs użytkownika



#### Interakcja z systemami komputerowymi

Przyciski użyte w mózgu są początkiem podstawowego interfejsu użytkownika (UI). Interfejs użytkownika to przestrzeń, która umożliwia użytkownikowi interakcję z systemem komputerowym (lub maszyną). Kiedy zaprogramowałeś przyciski w mózgu, dałeś użytkownikom możliwość interakcji z Clawbotem, aby mogli podnosić i opuszczać ramię. Istnieją inne typy interfejsów użytkownika (UI), w tym graficzne interfejsy użytkownika (GUI), takie jak ekrany dotykowe w samochodach i na smartfonach. Kiedy wchodzisz w interakcję z ekranem dotykowym na jednym ze swoich urządzeń (tablet, smartfon, smartwatch), ekrany te są często jedynym posiadanym interfejsem tych urządzeń. Być może Twoje urządzenie ma również przyciski głośności lub zasilania, ale głównie wchodzisz w interakcję z ekranem.

Na pilocie telewizora znajdują się przyciski zaprogramowane do wyłączania telewizora lub zwiększania głośności po ich naciśnięciu. Niektóre przykłady interfejsów użytkownika obejmują przyciski na kontrolerze gier wideo lub przyciski na kuchence mikrofalowej. Sposób projektowania tych interfejsów użytkownika zależy od sposobu działania urządzenia i interakcji użytkowników z nim. Te zasady projektowania stanowią podstawę doświadczenia użytkownika (UX) podczas korzystania z interfejsu użytkownika.

Wrażenia użytkownika to to, jak dobrze interfejs pozwala mi jako użytkownikowi robić to, co próbuję zrobić. Czy interfejs działa tak, jak tego oczekuję? Czy reaguje na to, co próbuję zakomunikować naciskając na przyciski? Czy jest dobrze zorganizowany, czy też przyszłe wersje interfejsu użytkownika powinny przenosić przyciski, w inne miejsce, aby korzystanie z nich było to łatwiejsze? Jak ogólnie wygląda interfejs? Czy przyjemnie się na niego patrzy i czy sprawia, że chcę go częściej używać?

Gdy interfejs użytkownika jest nadal rozwijany i przechodzi iteracje, programiści zbierają dane o tym, co działa zgodnie z planem, a co należy naprawić lub ulepszyć. Dane te informują, że należy przeprowadzić kolejne projektowanie iteracyjne. Niektóre z zalecanych zmian UX mają miejsce przed wydaniem urządzenia. Ale urządzenie może być również sprzedawane w stanie, w jakim jest, a zmiany te są wprowadzane później, zanim kolejna wersja zostanie zaoferowana konsumentowi.

## Kontroler jako interfejs użytkownika



Przykładowy projekt Controller Buttons z VEXcode IQ Blocks

## Zdalne sterowanie robotem

Do interakcji z naszymi telewizorami najczęściej używamy pilotów. Naciskamy przyciski, które powodują, że telewizor wyświetla żądany kanał lub ekran informacyjny / dostępowy. Technicznie rzecz biorąc, pilot do telewizora to interfejs użytkownika. Jest jednak znacznie mniej wyrafinowany niż ten, z którego korzysta Twój smartfon.

Programowanie kontrolera IQ jest znacznie bardziej wyrafinowane. Podczas rozgrywek kontrolowanych przez kierowcę chcesz, aby Twój kierowca / zespół miał jak najwięcej korzyści. Możesz zaprogramować przyciski i joysticki, aby wykonywały więcej niż jedno proste polecenie, i możesz zaprogramować je tak, aby wykonywały złożone polecenia, gdy przyciski / joysticki są używane w danej kombinacji - podobnie jak działają niektóre kontrolery do gier. Jako programista kontrolera, zastanawiasz się, które przyciski użyć w kombinacji oraz w jaki sposób powinny być umieszczone twoje palce i dłonie, aby dotrzeć do wszystkich przycisków.

Powyższy obrazek przedstawia projekt z przykładowego projektu Controller Buttons z VEXcode IQ Blocks. Czy widzisz, jak pętla *na zawsze* ma dwa zagnieżdżone w niej bloki *jeżeli to w przeciwnym razie*? Sprawia to, że projekt sprawdza, czy przyciski R Up lub R Down są wciśnięte, a robot obraca silnik kleszczy w kierunku zamykania lub otwierania w zależności od naciśnięcia. Ta pętla jest bardzo ważna. To sprawia, że kontroler IQ wielokrotnie sprawdza, które przyciski są naciskane, tak aby robot wykonywał odpowiednie zadanie.

Weź pod uwagę, że możesz umieścić dwa bloki *naciśnij kontroler* oraz blok operatora *i* . Dzięki temu projekt sprawdziłby, czy naciśnięto dwa przyciski jednocześnie. Możesz również wstawić blok *i* wewnątrz innego bloku *i* i sprawdzić trzy warunki przed wykonaniem danej akcji. Umożliwiłoby to zaprogramowanie sekwencji zachowań, które można uruchomić po prostu naciskając przyciski na kontrolerze.

Zastanów się, więc o ile więcej kombinacji różnych warunków możesz utworzyć za pomocą wszystkich przycisków i ich kombinacji na kontrolerze. Oczywiście, w miarę programowania bardziej złożonych zachowań w funkcjonowaniu Kontrolera, projekt zbliża się do autonomii robota. Zespół musi dowiedzieć się, jakie zachowania najlepiej zaprogramować w swoich kontrolerach jako złożone sekwencje, a które lepiej rozłożyć na wiele mniejszych części, aby kontroler pozwolił kierowcy (użytkownikowi) na większą kontrolę nad szybkością i dokładnością zachowania robota.



Czy istnieje skuteczniejszy sposób, aby dojść do tego samego wniosku? Zastanów się nad tym czego się nauczyłeś i spróbuj to ulepszyć.

# Przygotuj się na wyzwanie związane z interfejsem użytkownika



## Interfejs do chwytania i podnoszenia!

W wyzwaniu związanym z interfejsem użytkownika musisz zaprogramować robota tak, aby użytkownik mógł używać przycisków Sprawdź, W górę i W dół w mózgu, aby podnosić różne obiekty.

Twój interfejs użytkownika do kontrolowania Clawbota będzie wymagał:

- Przycisk lub przyciski do otwierania kleszczy
- Przycisk lub guziki do zamykania kleszczy
- Przycisk lub przyciski do podnoszenia ramienia
- Przycisk lub przyciski do opuszczania ramienia

Aby ukończyć wyzwanie, będziesz potrzebować:

- Clawbot
- Przedmioty do podnoszenia: pusta puszka lub butelka wody, kostka VEX, nieużywany element z zestawu VEX lub cokolwiek innego, co może dostarczyć nauczyciel

# Projektuj, rozwijaj i iteruj swój projekt

W tej części użyjesz projektów, które zbudowałeś wcześniej, do stworzenia programu, który może manipulować zarówno ramieniem, jak i kleszczami.

when started	when started
forever	forever
if Brain Up - button pressed? then	if Brain Up  button pressed? then
spin ArmMotor 👻 up 👻	spin ClawMotor ▼ open ▼
else	else
if Brain Down - button pressed? then	if Brain Down - button pressed? then
spin ArmMotor ▼ down ▼	spin ClawMotor ▼ close ▼
else	else
stop ArmMotor 👻	stop ClawMotor 🗸
٠	و
ArmUpDown2	ClawUpDown

Przypomnij sobie projekty ArmUpDown2 i ClawUpDown.

Chcemy w jakiś sposób włączyć oba te projekty do tego samego programu. Jest jednak tylko jeden przycisk Brain Up i jeden przycisk Brain Down.

Dlatego potrzebujemy przycisku, który będzie działał jako "przełącznik" między ramieniem a kleszczami.

Skorzystaj z następującego schematu bloków, które pomogą Ci zbudować projekt:

when started forever	if Brain check  button pressed? then else
if Brain Up ← button pressed? spin ArmMotor ← up ← else	then if Brain Up  button pressed? then spin ClawMotor  open  else
if Brain Down ▼ button pressed? then spin ArmMotor ▼ down ▼ else stop ArmMotor ▼	if Brain Down ▼ button pressed? then spin ClawMotor ▼ close ▼ else stop ClawMotor ▼

Podczas planowania projektu odpowiedz na następujące pytania w notatniku inżyniera:

- Do czego chcesz zaprogramować robota? Wyjaśnij szczegółowo.
- Ile warunków będzie musiał sprawdzić Twój projekt w blokach *jeżeli to w przeciwnym razie*?

*Wskazówka:* użyj przycisku Brain Check jako "przełącznika" między ramieniem a kleszczami. Tak więc, jeśli przycisk Brain Check jest wciśnięty i przytrzymany, ramię jest kontrolowane za pomocą przycisków Brain Up i Down. Jeśli przycisk Brain Check zostanie zwolniony, kleszcze są kontrolowane za pomocą przycisków Brain Up i Down.

Podczas tworzenia projektu wykonaj poniższe czynności:

- Zaplanuj warunki, które musi sprawdzić twój program, używając rysunków i pseudokodu.
- Użyj pseudokodu, który stworzyłeś, aby rozwinąć stworzyć projekt.
- Często testuj projekt i iteruj go, korzystając z tego, czego nauczyłeś się podczas testów.
- Co możesz dodać do swojego projektu, aby lepiej kontrolować silniki kleszczy i ramion? Wyjaśnij szczegółowo.
- Podziel się końcowym projektem z nauczycielem.

Jeśli masz problemy z rozpoczęciem, zapoznaj się z poniższymi informacjami o blokach VEXcode IQ:

- Filmy instruktażowe jeżeli to w przeciwnym razie lub Używanie pętli
- Korzystanie z samouczka
- Poprzednie wersje Twojego projektu (ArmUpDown2 lub ClawUpDown)



## Wyzwanie dotyczące interfejsu użytkownika



## Wyzwanie dotyczące interfejsu użytkownika

W wyzwaniu interfejsu użytkownika zaprogramujesz Clawbota tak, aby użytkownik mógł naciskać przyciski Sprawdź, Góra i Dół w Mózgu, aby sterować silnikami ramion i kleszczy. Następnie przyciski te będą używane do podnoszenia i zastępowania dziesięciu różnych obiektów. To wyzwanie nie wymaga, aby Clawbot jeździł i skręcał. Przedmioty są podnoszone, a następnie umieszczane w tym samym miejscu na stole lub podłodze.

#### Zasady:

- Jeden przycisk (przycisk W górę) lub jedna kombinacja przycisków (przycisk Sprawdź i przycisk W górę) może wykonywać tylko jedną z czterech czynności: otworzyć kleszcze, zamknąć kleszcze, unieść ramię lub je opuścić.
- Korzystanie z Kontrolera jest niedozwolone.
- Każdy Clawbot będzie musiał podnieść i wymieniać jak najwięcej obiektów w ciągu jednej minuty bez upuszczania ich. Zalecane jest podnoszenie i wymiana jednego przedmiotu na raz.
- Runda jednominutowa kończy się po upływie 1 minuty lub jeśli jakiś przedmiot zostanie upuszczony - nawet jeśli runda trwa zaledwie kilka sekund. Upuszczenie przedmiotu dyskwalifikuje drużynę z pełnej minuty rundy, ale wszelkie punkty zdobyte przed upadkiem są liczone.
- Jeśli wszystkie obiekty zostały podniesione przed zakończeniem jednej minuty rundy, obiekty można ponownie wykorzystać do czasu zakończenia rundy.

- Przedmiot należy podnieść wyżej niż silnik ramienia, zanim zostanie umieszczony na stole.
- Każdy przedmiot, który udało się złapać i podnieść, a następnie opuścić i odłożyć, jest wart jeden punkt.
- Pomiędzy rundami role można zmieniać, ale tylko jeden użytkownik, przypadający na Clawbota, może grać w każdej rundzie.
- Grupa z największą liczbą punktów na koniec wszystkich rund wygrywa!

Round Number	Clawbot ID	Number of Items Successfully Lifted and Replaced

#### Role w wyzwaniu:

- W każdej rundzie powinien być tylko jeden użytkownik (podnośnik) podnoszący przedmioty, ale grupy mogą zamieniać użytkowników między rundami.
- Powinien istnieć wyznaczony sekretarz, który ocenia, ile obiektów jest w stanie podnieść każdy Clawbot. Każdy pomyślnie wymieniony obiekt jest wart jeden punkt. Po upuszczeniu obiektu tura Clawbota się kończy. Można rozegrać kilka rund. Tabelę punktacji znajduje się powyżej, ale można dodać wiersze dla dodatkowych grup i rund. Każda grupa może korzystać z własnej tabeli punktacji lub każdy może korzystać z tej samej. Jeśli wiele grup używa tej samej tabeli wyników, zapisz identyfikator Clawbota lub nazwę grupy dla każdej z nich w drugiej kolumnie. Ponadto sekretarz powinien zsumować punkty z wszystkich rund na koniec wyzwania, aby uzyskać łączny wynik.

- Powinna być wyznaczona osoba do zmiany obiektów na stole: Zamieniacz.
   Ponieważ Clawbot nie skręca ani nie jedzie, podniesiony przedmiot będzie musiał zostać usunięty i zastąpiony nowym po każdej próbie.
- Powinna być wyznaczona osoba do śledzenia czasu: **Osoba mierzący czas**. Każda runda trwa jedną minutę.
- Nauczyciel powinien dostarczyć przedmioty do tego wyzwania, przed jego rozpoczęciem.



Zrozum podstawowe pojęcia i dowiedz się, jak zastosować je w różnych sytuacjach. Ten proces powtórki będzie motywował do nauki.

## Powtórzenie

#### 1. Dlaczego programista miałby używać warunków w projekcie?

- Aby zapętlić zachowanie robota
- Aby robot obracał kolejno swoje silniki
- Aby robot podejmował decyzje
- Aby robot odtwarzał dźwięki
- 2. W tym projekcie (rysunek na kolejnej stronie) , zgodnie z jego przebiegiem, jaki jest następny blok, który program będzie wykonywał, jeśli pierwszy warunek naciśnięcia przycisku Brain Up jest fałszywy (przycisk jest zwolniony)?
  - o Drugi warunek naciśnięcia przycisku Brain Down zostanie zaznaczony.
  - o Blok obróć.
  - o Blok czekaj do.
  - Blok zatrzymaj silnik.



3. Ten projekt jest uruchomiony i robot skręcił w prawo. Jaki warunek jest PRAWDZIWY?

forever				
if	Brain	check -	button pressed?	the
turn	right	-		
else	/			
driv	e forwa	ard 🗢		

- Wciśnięty jest przycisk Sprawdź.
- Przycisk W górę jest wciśnięty.
- Przycisk Sprawdź NIE jest wciśnięty.
- Przycisk W górę NIE jest wciśnięty.

#### 4. Który z poniższych, jest najlepszym pseudokodem dla tego projektu?

						then
sp	oin	ArmMo				
else						
	<				d? the	n
1	spin	Arm	Motor 🚽	n 🕶		
el	se					
	stop	Arm	Motor 🖣			

- Jeśli przycisk Brain Up jest wciśnięty (prawda) i przycisk Brain Down jest wciśnięty (prawda), opuść ramię. Jeśli przycisk "Brain Up" jest wciśnięty (prawda) i przycisk "Down" jest zwolniony (fałsz), podnieś ramię. Jeśli przyciski Brain Up lub Down są wciśnięte (true), zatrzymaj silnik ramienia.
- Podnieś ramię po naciśnięciu dwóch przycisków. Opuść ramię po naciśnięciu jednego przycisku. Skręć w prawo, gdy żaden przycisk nie jest wciśnięty
- Jeśli przycisk Brain Up jest wciśnięty (prawda), obróć ramię do góry. Jeśli jest fałszywy (zwolniony), sprawdź, czy przycisk Brain Down jest wciśnięty. Jeśli przycisk.
   Brain Down jest wciśnięty (prawda), obróć ramię w dół. Jeśli jest fałszywy (zwolniony), zatrzymaj ramię.
- Użyj przycisków Sprawdź, W górę i W dół jako interfejsu użytkownika.

#### 5. Który z poniższych NIE jest typem interfejsu użytkownika (UI)?

- Kontroler IQ
- o Klamka do drzwi
- Ekran dotykowy smartfona
- Przyciski na kuchence mikrofalowej

### APPENDIX

Dodatkowe informacje, zasoby i materiały.

## Artykuły w bazie wiedzy

#### Linki do artykułów bazy wiedzy VEX Robotics dla tego laboratorium STEM:

- How to Turn On/Off a VEX IQ Robot Brain https://kb.vex.com/hc/en-us/articles/360035952571-How-to-Turn-On-Off-a-VEX-IQ-Robot-Brain
- How to Read Indicator Lights on the VEX IQ Robot Brain https://kb.vex.com/hc/en-us/articles/360035590672-How-to-Read-Indicator-Lightson-the-VEX-IQ-Robot-Brain
- How to Navigate the VEX IQ Robot Brain https://kb.vex.com/hc/en-us/articles/360035952331-How-to-Navigate-the-VEX-IQ-Robot-Brain
- How to Connect VEX IQ Devices to Smart Ports https://kb.vex.com/hc/en-us/articles/360035952151-How-to-Connect-VEX-IQ-Devices-to-Smart-Ports
- How to Install or Remove the VEX IQ Robot Battery https://kb.vex.com/hc/en-us/articles/360035951991-How-to-Install-or-Remove-the-VEX-IQ-Robot-Battery
- How to Charge the VEX IQ Robot Battery https://kb.vex.com/hc/en-us/articles/360035955011-How-to-Charge-the-VEX-IQ-Robot-Battery
- How to Use the Autopilot Program in the Demos Folder https://kb.vex.com/hc/en-us/articles/360035952031-How-to-Use-the-Autopilot-Program-in-the-Demos-Folder
- Best Practices for Preserving the VEX IQ Robot Battery's Life https://kb.vex.com/hc/en-us/articles/360035953671-Best-Practices-for-Preservingthe-VEX-IQ-Robot-Battery-s-Life
- Ideas for Organizing the VEX IQ Super Kit https://kb.vex.com/hc/en-us/articles/360035590332-Ideas-for-Organizing-the-VEX-IQ-Super-Kit
- VEX IQ Brain Status (USB Cable) https://kb.vex.com/hc/en-us/articles/360035955411-How-to-Understand-the-VEX-IQ-Brain-Status-Icon-USB-VEXcode-IQ-Blocks

#### Linki do artykułów bazy wiedzy VEXCode IQ Blocks dla tego laboratorium STEM:

- How to Begin a New Project in VEXcode IQ Blocks https://kb.vex.com/hc/en-us/articles/360035954551-How-to-Begin-a-New-Project-VEXcode-IQ-Blocks
- How to Download and Run a Project https://kb.vex.com/hc/enus/articles/360035591232-How-to-Download-and-Run-a-ProjectVEXcode-IQ-Blocks

- How to Save a Project on Windows https://kb.vex.com/hc/en-us/articles/360035954531-How-to-Save-a-Project-on-WindowsVEXcode-IQ-Blocks
- How to Save a Project on macOS https://kb.vex.com/hc/en-us/articles/360035954511-How-to-Save-a-Project-on-macOSVEXcode-IQ-Blocks
- How to Save a Project on Chromebook https://kb.vex.com/hc/enus/articles/360035955351-How-to-Save-on-a-ChromebookVEXcode-IQ-Blocks
- How to Download to a Selected Slot on the Brain https://kb.vex.com/hc/enus/articles/360035591292-How-to-Download-to-a-Selected-Sloton-the-Brain-VEXcode-IQ-Blocks

## Identyfikacja belek kątowych



### Jak rozpoznać różne kąty belek kątowych

Istnieją cztery różne typy belek, które mają kąty: 30 °, 45 °, 60 ° i 90 °. Istnieją również trzy typy belek pod kątem prostym: 3x5, 2x3 i typu offset. Najlepszym sposobem na określenie, które z belek posiadaj jaki kąt, jest ułożenie ich jedna na drugiej. Następnie możesz porównać, jak wyglądają. Możesz również użyć kątomierza do pomiaru.

# Instalowanie gumowych kołnierzy wału



Użyj ręki by podnieść temperaturę kołnierza

## Kołnierz mięknie gdy robi się ciepły

Przytrzymaj gumowe kołnierze wałka w dłoni przez 15-30 sekund, zanim wsuniesz je na wałek. Trzymanie gumowego kołnierza w dłoni rozgrzeje i zmiękczy gumę, ułatwiając wsuwanie się na wałek.

## Usuwanie łączników z belek i płyt



Użycie wału podziałowego do usunięcia

### Jak łatwo usunąć złącza

Możesz łatwo usunąć łączniki narożne z belek lub płyt, umieszczając metalowy trzon w jednym z otworów łącznika narożnego i pociągając na zewnątrz, przytrzymując jednocześnie belkę lub płytę.

## Usuwanie pinów z belek i płyt VEX IQ



Usuwanie pinu / kołka płytki za pomocą belki

### Jak łatwo usunąć kołki / piny z belek i płyt

Możesz szybko usunąć łączniki z belek lub płyt, dociskając belkę do tylnej części pinu / kołka, co częściowo go wypycha, dzięki czemu można go wyjąć palcami. Możesz użyć tej techniki, aby łatwiej usunąć je z poszczególnych płyt i belek lub z konstrukcji.

# Usuwanie elementów dystansowych ze złączek dystansowych mini



Usunięcie elementu dystansowego ze złącza dystansowego mini

### Jak łatwo usunąć części ze złączy dystansowych mini

Złącza dystansowe i mini dystansowe można rozdzielić, przepychając wałek przez złącze. Tej samej techniki można użyć do części z podobnymi końcami w złączach Mini, np.: za pomocą pinów.

# Zabezpieczenie wałów za pomocą gumowych kołnierzy



Zabezpieczanie wału gumowym kołnierzem

### Jak zabezpieczyć wały za pomocą gumowych kołnierzy

Wały mogą bardzo łatwo wypaść z miejsca lub przechylić się, jeśli nie są odpowiednio zabezpieczone. Możesz zabezpieczyć go przed wypadnięciem, zakładając gumowy kołnierz na jego końcu. Następnie można połączyć wał z konstrukcją wsporczą za pomocą kołnierza umieszczonego na nim. Umożliwi to obracanie się wału, ale zapobiegnie jego chybotaniu lub wypadaniu.

# Zabezpeczenie wału za pomocą tulei



Zabezpieczanie wałów za pomocą tulei

## Jak zabezpieczyć wały za pomocą tulei

Wały mogą bardzo łatwo wypaść lub przechylić się, jeśli nie są odpowiednio podparte. Możesz zabezpieczyć go przed wypadnięciem, umieszczając na jego końcu tuleję.

# Rozwiązania zadań STEM Labs

# Pseudokod wyzwania interfejsu użytkownika

Pseudokod wyzwania interfejsu użytkownika:



Kod blokowy wyzwania interfejsu użytkownika:

