

Clawbot z kontrolerem



Zaprogramuj kontroler VEX, aby kierował Clawbotem przez kilka angażujących wyzwań przy użyciu koncepcji pętli.



Odkryj nowe praktyczne konstrukcje i możliwości programowania, aby zrozumieć omawiany temat.

Gotowy Clawbot V5



Gotowy Clawbot V5

VEX V5 Clawbot to rozszerzenie Speedbot'a VEX, które można zaprogramować do poruszania się i interakcji z obiektami.

Lista elementów: część 1

Możesz go zbudować z:

• VEX V5 Classroom Starter Kit



Lista elementów: część 2



Instrukcja budowania









Zielona ikona wskazuje, że konstrukcję należy odwrócić (do góry nogami).





Obecnie używany jest tylko jeden z dwóch podzespołów wykonanych w tym kroku. Drugi zostanie użyty później w kroku 9.



Upewnij się, że inteligentne silniki są ustawione we właściwym kierunku (otwory na śruby skierowane na zewnątrz konstrukcji, a otwór wału do wewnątrz).







Upewnij się, że inteligentne silniki są ustawione we właściwym kierunku (otwory na śruby skierowane na zewnątrz konstrukcji, a otwór wału do wewnątrz).

















Zielona ikona wskazuje, że konstrukcję należy obrócić (o 180 stopni).





Obrazki w niebieskich kwadratach (objaśnienia) pokazują, jaka powinna być orientacja mózgu robota, gdyby konstrukcja została odwrócona prawą stroną do góry. Upewnij się, że 3przewodowe porty w mózgu robota są skierowane w stronę radia V5!





Zielone elementy na obrazku wskazują, do którego portu w Mózgu robota należy podłączyć każde urządzenie za pomocą odpowiedniego kabla.















Upewnij się, że w tym kroku dodałes dwa elementy!



Element z tego kroku dodajemy do dwóch elementów z kroku 29.



Upewnij się, że dodałeś ten element tylko do jednego z dwóch podzespołów, które właśnie utworzyłeś.
















Upewnij się, że 12-zębowe koło zębate jest zainstalowane po prawej stronie kleszczy.





Upewnij się, że port na silniku Smart Motor jest skierowany w prawą stronę robota, gdy kleszcze są zainstalowane (po tej samej stronie co radio V5).



Wskazówki

Sprawdź Dodatek, aby uzyskać informacje na temat używania elementów Hex Nut Retainers.

Analiza

Po zakończeniu budowy sprawdź, co potrafi robot. Poznaj swoją konstrukcję, a następnie odpowiedz na poniższe pytania w swoim notatniku technicznym.

Zwróć uwagę, jak przednie koła Clawbota są napędzane silnikami. Pomyśl o rozłożeniu ciężaru na Clawbocie.

- A gdyby tak zwiększyć prędkość lewego i prawego silnika, a robot był sterowany za pomocą kontrolera?
- Gdyby robot został poproszony o szybką jazdę wstecz, co mogłoby się stać i dlaczego? zastanów się, gdzie na robocie znajduje się ciężar ramienia, kleszczy, baterii i Mózgu.

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

Pętle: upraszczanie powtarzalnych czynności – bloki VEXcode V5

Uprość projekty dzięki pętlom

My, jako ludzie, powtarzamy wiele zachowań w naszym codziennym życiu. Od jedzenia i spania po mycie zębów i spacery z psami - większość tego, co robimy każdego dnia, jest powtarzalne. Na lekcjach matematyki wiemy, że pomnożenie liczby przez zero zawsze będzie równe zero lub że pomnożenie liczby przez jeden będzie zawsze równe liczbie, niezależnie od tego, ile razy to zrobimy. Chociaż mamy tendencję do powtarzania naszych zachowań, czasami nasze instrukcje mogą być uproszczone. Na przykład, jeśli korzystasz z przepisu na upieczenie ciasta, nie znajdziesz tam: "dodaj 1 szklankę cukru, 1 szklankę cukru, 1 szklankę cukru". Zamiast tego znajdziesz instrukcję, abyś dodał cztery szklanki cukru.

Pętle pomagają nam uprościć nasze projekty dla robotów. Zamiast na przykład dodawać ten sam blok cztery razy, możemy użyć pętli, aby nakazać robotowi wykonanie tego samego zachowania cztery razy, oszczędzając czas i miejsce podczas tworzenia projektu. Wyobraź sobie zadanie, które mógłby wykonać robot, a które wymagałoby powtórzenia konkretnych zachowań. Te zachowania, wraz z blokiem pętli z kategorii Blok sterowania, są tym, co musisz zaprojektować, aby wykonać zadanie.

Controller: Tank Drive Exploration-VEXcode V5 Blocks

Hardware/Software Required:

Quantity	Hardware/Other Items
1	VEX V5 Classroom Starter Kit (with up-to-date firmware)
1	VEXcode V5 Blocks (latest version, Windows, MacOS, Chromebook)
1	Engineering Notebook
1	Using Loops (Tutorial)
1	Tank Drive example project
2 or more	Classroom items to be used as obstacles

This activity will give you the tools to program your Controller.

You can use the Help information inside of VEXcode V5 Blocks to learn about the blocks. For guidance in using the Help feature, see the Using Help tutorial.

1. Let's start programming the Controller.

- Start by watching the **Using Loops** tutorial video.
- Open the Tank Drive example project.

							,
set	LeftMotor -	velocity to	Controller1 👻	3 - position) % -	Using a Clawbot - The Left up/down Controller	Axis (3) will drive the
	RightMotor -	velocity to	Controller1 -	2 - position	n % 🗸	- The Right left/right Controller turning the robot to the left or	Axis (2) will control right.
	LeftMotor 👻						
	RightMotor -	forward 🗢					

Wykonaj następujące czynności w swoim notatniku inżynieryjnym:

Zastanów się, co projekt zrobi z Clawbotem. Wyjaśnij więcej niż fakt, że projekt wykorzystuje kontroler.

W jaki sposób joysticki poruszają robotem? Co robi Clawbot?

• Zapisz, pobierz i uruchom przykładowy projekt Tank Drive.

 Aby uzyskać pomoc, zobacz samouczek dotyczący bloków VEXcode V5, który wyjaśnia, jak pobrać i uruchomić projekt.

<u>∖</u> 5 ⊕ -	File	🔆 Tutorials 🗲	
----------------	------	---------------	--

 Sprawdź wyjaśnienia projektu w swoim notatniku inżyniera i dodaj notatki, aby w razie potrzeby poprawić swoje przemyślenia.

2. Kontroler: Tank Drive

Jakie są zalety korzystania z bloku zawsze?

Oto nasz projekt bez bloku zawsze:

set	LeftMotor 🔻	velocity to	Controller1 -	position	% 🗢
set	RightMotor 🗢	velocity to	Controller1 -	position	% -
spin	LeftMotor 🔻	forward 🔻			

Jak myślisz, co by się stało, gdyby ten program został uruchomiony? Omów w grupie. Zapisz swoje przewidywania w notatniku technicznym.

3. Nawiguj w slalomie!

Slalomy to trasy, po których uczestnik musi poruszać się wokół ustawionych flag lub markerów. Slalomy narciarskie są popularnym sportem zimowym i są częścią Zimowych Igrzysk Olimpijskich.

Teraz, gdy kontroler jest sparowany i projekt został pobrany, możesz przenieść i poruszyć Clawbota za pomocą kontrolera!

- Powinieneś odebrać od nauczyciela cztery przedmioty z klasy, których użyjesz jako flag w slalomie.
- Powinieneś współpracować z innymi, aby je ustawić, tak by Kierowca mógł poruszać Clawbotem, zgodnie ze schematem slalomu.
- Powinieneś pobrać projekt *Tank Drive*. Jeśli są jakieś pytania dotyczące pobierania projektu, zapoznaj się z samouczkiem *Pobierz i uruchom projekt*.

4. Robo-Slalom

Użyj kontrolera, aby przesunąć swojego Clawbota po zewnętrznej stronie każdej "flagi". Ścieżka robota musi uniemożliwiać mu dotknięcie flagi i pozwalać na przekroczenie linii mety.

- Powinieneś uruchomić projekt i poprowadzić Clawbota do przodu i do tyłu oraz skręcić w lewo i w prawo używając obu joysticków.
- Powinieneś określić, ile czasu zajmie Clawbotowi ukończenie jazdy. Dokumentuj czasy w notatniku.
- Jak szybko możesz przeprowadzić Clawbota przez tor?

Pętle: upraszczanie powtarzalnych czynności – programowanie tekstowe VEXcode V5

Uprość projekty dzięki pętlom

My, jako ludzie, powtarzamy wiele zachowań w naszym codziennym życiu. Od jedzenia i spania po mycie zębów i spacery z psem - większość tego, co robimy każdego dnia, jest powtarzalne. Na lekcjach matematyki wiemy, że pomnożenie liczby przez zero zawsze będzie równe zero lub że pomnożenie liczby przez jeden będzie zawsze równe liczbie, niezależnie od tego, ile razy to zrobimy. Chociaż mamy tendencję do powtarzania naszych zachowań, czasami instrukcje ich wykonania mogą być uproszczone. Na przykład, jeśli korzystasz z przepisu na ciasto, nie znajdziesz tam: "dodaj 1 szklankę cukru, 1 szklankę cukru, 1 szklankę cukru". Zamiast tego znajdziesz instrukcję, abyś dodał 4 szklanki cukru.

Pętle pomagają nam uprościć projekty dla robotów. Zamiast, np.: dodawać ten sam blok 4 razy, możemy użyć pętli, aby nakazać robotowi wykonanie tego samego zachowania 4 razy, oszczędzając czas i miejsce podczas tworzenia projektu. Wyobraź sobie zadanie, które mógłby wykonać robot, a które wymagałoby powtórzenia konkretnych zachowań. Te zachowani, wraz z blokiem pętli, są tym, co musisz zaprojektować, aby wykonać zadanie.

Kontroler: Analiza *Tank Drive* - programowanie tekstowe

Wymagany sprzęt / oprogramowanie:

llość	Sprzęt / inne przedmioty
1	VEX V5 Classroom Starter Kit (z aktualnym oprogramowaniem)
1	VEXcode V5 Text (najnowsza wersja)
1	Notes inżyniera
1	Samouczek korzystania z pętli
1	Przykładowy projekt Tank Drive
2 lub więcej	Przedmioty z klasy jako przeszkody

To ćwiczenie dostarczy Ci umiejętności do programowania kontrolera.

1. Zacznijmy programować Kontroler.

• Otwórz przykładowy projekt Tank Drive.

```
20
21
     #include "vex.h"
22
23
     using namespace vex;
24
25
     int main() {
26
       // Initializing Robot Configuration. DO NOT REMOVE!
27
       vexcodeInit();
28
29
       // Deadband stops the motors when Axis values are close to zero.
30
       int deadband = 5;
31
32
       while (true) {
33
         // Get the velocity percentage of the left motor. (Axis3)
34
         int leftMotorSpeed = Controller1.Axis3.position();
35
         // Get the velocity percentage of the right motor. (Axis2)
         int rightMotorSpeed = Controller1.Axis2.position();
36
37
38
         // Set the speed of the left motor. If the value is less than the deadband,
39
         // set it to zero.
40
         if (abs(leftMotorSpeed) < deadband) {</pre>
41
           // Set the speed to zero.
42
          LeftMotor.setVelocity(0, percent);
43
         } else {
44
           // Set the speed to leftMotorSpeed
45
           LeftMotor.setVelocity(leftMotorSpeed, percent);
         }
46
47
48
         // Set the speed of the right motor. If the value is less than the deadband,
49
         // set it to zero.
50
         if (abs(rightMotorSpeed) < deadband) {
51
           // Set the speed to zero
52
           RightMotor.setVelocity(0, percent);
53
         } else {
54
           // Set the speed to rightMotorSpeed
55
           RightMotor.setVelocity(rightMotorSpeed, percent);
56
         }
57
58
         // Spin both motors in the forward direction.
59
         LeftMotor.spin(forward);
60
         RightMotor.spin(forward);
61
62
         wait(25, msec);
       }
63
64
     }
65
```

Wykonaj następujące czynności w swoim notatniku inżyniera:

Zastanów się, co projekt każe zrobić Clawbot'owi. Wyjaśnij więcej niż fakt, że projekt wykorzystuje kontroler.

W jaki sposób joysticki poruszają robotem? Co robi Clawbot?

• Zapisz, pobierz i uruchom przykładowy projekt Tank Drive.

 Sprawdź wyjaśnienia projektu w swoim notatniku i dodaj komentarze, aby w razie potrzeby poprawić swoje przemyślenia.

2. Kontroler: Tank Drive

Jakie są korzyści z używania instrukcji zawsze?

Oto nasz projekt bez instrukcji zawsze:

```
20
     #include "vex.h"
21
22
23
     using namespace vex;
24
     int main() {
25
        // Initializing Robot Configuration. DO NOT REMOVE!
26
27
        vexcodeInit();
28
29
        int deadband = 5;
30
31
32
          int leftMotorSpeed = Controller1.Axis3.position();
33
          int rightMotorSpeed = Controller1.Axis2.position();
34
          if (abs(leftMotorSpeed) < deadband) {
35
           LeftMotor.setVelocity(0, percent);
36
          } else {
37
           LeftMotor.setVelocity(leftMotorSpeed, percent);
38
39
          }
40
41
          if (abs(rightMotorSpeed) < deadband) {</pre>
42
            RightMotor.setVelocity(0, percent);
          } else {
43
44
           RightMotor.setVelocity(rightMotorSpeed, percent);
45
          }
46
47
          LeftMotor.spin(forward);
          RightMotor.spin(forward);
48
49
50
          wait(25, msec);
51
52
     }
53
```

Jak myślisz, co by się stało, gdyby ten program został uruchomiony? Omów w grupie. Zapisz swoje przewidywania w notatniku technicznym.

3. Nawiguj w slalomie!

Slalomy to trasy, po których uczestnik musi poruszać się wokół ustawionych flag lub markerów. Slalomy narciarskie są popularnym sportem zimowym i są częścią Zimowych Igrzysk Olimpijskich.

Teraz, gdy kontroler jest sparowany i projekt został pobrany, możesz przenieść i poruszyć Clawbota za pomocą kontrolera!

- Powinieneś odebrać od nauczyciela cztery przedmioty z klasy, których użyjesz jako flag w slalomie.
- Powinieneś współpracować z innymi, aby je ustawić, aby Kierowca mógł poruszać Clawbotem, zgodnie ze schematem slalomu.
- Powinieneś pobrać projekt Tank Drive. Jeśli są jakieś pytania dotyczące pobierania projektu, zapoznaj się z samouczkiem *Pobierz i uruchom projekt.*

4. Robo-Slalom

Użyj kontrolera, aby przesunąć swojego Clawbota po zewnętrznej stronie każdej "flagi". Ścieżka robota musi uniemożliwiać mu dotknięcie flagi i pozwalać na przekroczenie linii mety.

- Powinieneś uruchomić projekt i poprowadzić Clawbota do przodu i do tyłu oraz skręcić w lewo i w prawo używając obu joysticków.
- Powinieneś określić, ile czasu zajmie Clawbotowi ukończenie jazdy. Dokumentuj czasy w notatniku.
- Jak szybko możesz przeprowadzić Clawbota przez tor?

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

Używanie pętli do robienia słodyczy

Taśma produkcyjna przesuwa czekoladki wzdłuż mechanicznej linii montażowej.

Pętle w produkcji

Roboty są w stanie wykonywać w kółko to samo zadanie, korzystając z pętli. Posiadanie robotów wykonujących powtarzalne zadania ma wiele zalet. Nie męczą się i nie potrzebują przerw (o ile mają stałą moc). Z tego powodu odegrały one kluczową rolę w produkcji, w której mogą stale wykonywać zadania, które byłyby bardziej wymagające a nawet niebezpieczne dla ludzi.

Przykładem branży, która skorzystała na robotach wykonujących powtarzalne zadania, jest przemysł cukierniczy. Roboty, takie jak Flexpicker firmy ABB, mogą używać przystawki próżniowej do zbierania setek cukierków na minutę. Roboty można zaprogramować tak, aby wywierały odpowiednią siłę na bardzo delikatne cukierki, aby nie zostały zmiażdżone. Roboty na linii montażowej mogą również używać czujników wizyjnych do identyfikowania zniekształconych cukierków, by ich nie podnosić Programowanie robotów z pętlami może zwiększyć wydajność produkcji, na przykład przemysłu cukierniczego.

Nawiązanie do konkurencji: kontrola kierowcy

Gra VRC 2019-202- - Boisko Tower Takeover

Tower Takeover

Używanie pętli do sterowania Clawbotem za pomocą kontrolera pozwoliło kierowcy na prowadzenie go po slalomie. Ćwiczenie prowadzenia Clawbota to przydatne zadanie, które pomoże ci przygotować się do jednego z wyzwań w mistrzostwach świata robotów VEX.

Gra VRC 2019-2020 (VEX Robotics Competition) nazywała się Tower Takeover. W tej grze drużyny muszą za pomocą robota podnieść i przesunąć określone kolorowe kostki do celów lub wież. Zespoły rozpoczną od 15-sekundowego okresu autonomii, w którym robot nie może otrzymać żadnej pomocy od kierowcy. W tym czasie robot jest zmuszony do zdobycia punktu po swojej stronie boiska bez pomocy kierowcy.

Po okresie autonomii zespoły przechodzą do 1-minutowego i 45-sekundowego czasu kontroli kierowcy, podczas którego manipulują robotem za pomocą kontrolera. Drużyny starają się zdobyć jak najwięcej punktów w tym czasie.

Możliwość obsługi i programowania kontrolera jest niezwykle ważnym czynnikiem wpływającym na dobre wyniki w zawodach.

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

Programowanie oparte na zdarzeniach: komunikacja między blokami

Programowanie oparte na zdarzeniach

Jeśli Twój pies przynosi Ci smycz lub siada przy drzwiach, daje Ci znać, że musi wyjść na zewnątrz. W szkole, kiedy nauczyciel zadaje pytanie i widzi, jak podnosisz rękę, wie, że uważasz, że znasz odpowiedź i chciałbyś odpowiedzieć na pytanie. Te zachowania są również nazywane "wyzwalaczami".

Twój pies wie, że przyniesienie smyczy lub siedzenie przy drzwiach jest wyzwalaczem, który daje ci znać, że musi wyjść na zewnątrz. Kiedy więc widzisz go siedzącego przy drzwiach ze smyczą, reagujesz, wyprowadzając go na zewnątrz. Podniesienie ręki jest wyzwalaczem, który informuje nauczyciela, że chcesz odpowiedzieć na jego pytanie. Następnie nauczyciel reaguje na wyzwalacz, wzywając cię do odpowiedzi.

Programowanie oparte na zdarzeniach w robotyce ma miejsce, gdy pewne zachowania robota powodują, że robot wykonuje określone czynności lub reaguje na określone wyzwalacze.

Kontroler: Clawbot kontrola

Teraz jesteś gotowy, aby pobrać przykładowy projekt i używać kontrolera do jednoczesnej obsługi Clawbota, jego ramienia i kleszczy!

Upewnij się, że masz wymagany sprzęt i notes inżynieryjny. Otwórz bloki VEXcode V5.

Hardware/Software Required:

Quantity	Hardware/Other Items
1	Clawbot
1	Charged Robot Battery
1	VEX V5 Radio
1	Controller
1	Tether Cable
1	VEXcode V5 Blocks
1	USB Cable (if using a computer)
1	Engineering Notebook

Before you begin the activity...

Do you have each of these items ready? The Builder should check each of the following:

- Are all the motors and sensors plugged into the correct port?
- Are the smart cables fully inserted into all of the motors?
- Is the battery fully charged?
- Is the Controller paired with the Robot Brain?

Przygotowanie do wyzwań – Bloki VEXcode V5

Przed rozpoczęciem projektu, musisz wybrać właściwy przykładowy projekt. Przykładowy projekt Clawbot Control zawiera konfigurację silników i czujników Clawbot. Jeśli szablon nie jest używany, twój robot nie uruchomi projektu poprawnie.

- Przejdź do menu plik, Otwórz Przykłady, a następnie wybierz przykładowy projekt *Clawbot Control.*
- Zapisz projekt.

• Sprawdź, czy nazwa projektu *Clawbot Control* znajduje się teraz w oknie "pośrodku paska narzędzi. Clawbot jest teraz poprawnie skonfigurowany, a projekt *Clawbot Control* jest gotowy do użycia.

Teraz spójrz, jak bloki są używane w tym projekcie. W swoich notatnikach inżyniera dokonaj następujących prognoz:

- Co się dzieje, gdy uruchomisz ten projekt? Co Clawbot będzie w stanie zrobić?
- Co by się stało, gdybyśmy nie użyli bloku "nie" w tym projekcie?

Wyzwania: Kontrola Clawbota – Bloki VEXcode V5

Chwyć przedmiot!

Celem tego działania jest złapanie i uwolnienie obiektu za pomocą kontrolera.

Oto kilka kroków, które poprowadzą Twój zespół:

- Umieść robota swojej grupy na podłodze i upewnij się, że Clawbot ma wystarczająco dużo miejsca, aby się poruszać bez przeszkadzania innym grupom.
- Wypisz kroki, jakie Clawbot będzie musiał wykonać, aby złapać obiekt. Pamiętaj, aby podać przyciski, których będziesz używać do wykonania tego zadania!
- Zapisz czynności wymienione w liście kierowcy w notatniku technicznym.
- Kliknij przycisk Pobierz na pasku narzędzi, aby pobrać projekt Clawbot Control do Mózgu.

- Sprawdź, czy Twój projekt został pobrany do mózgu Clawbota, patrząc na ekran robota. Nazwa projektu, Clawbot Control, powinna być wymieniona w slocie 1.
- Uruchom projekt na Clawbocie.
- Chwyć i uwolnij obiekt za pomocą kontrolera.

Gratulacje! Złapałeś przedmiot swoim Clawbotem za pomocą kontrolera!

Czy były jakieś różnice między twoimi przewidywaniami a działaniami, które wykonałeś podczas ćwiczenia? Jeśli tak, dodaj je do swojego notatnika technicznego.
Kolorowe klejnoty

Celem tej aktywności jest użycie umiejętności Clawbot do zebrania kilku obiektów, po kolei i położenia ich w wybranym miejscu szybciej niż inne grupy w twojej klasie. Powodzenia!



Oto kilka kroków, które poprowadzą Twój zespół:

- Przenieś obiekty swojej grupy na obszar *pobierania obiektów*, który wyznaczył twój nauczyciel i upewnij się, że Clawbot ma przestrzeń do poruszania się bez przeszkadzania innym grupom.
- Wymień kroki, jakie Clawbot będzie musiał wykonać, aby złapać każdy obiekt i zwrócić go. Pamiętaj, aby podać przyciski, których będziesz używać do wykonania tego zadania!
- Zapisz czynności wymienione w liście sterowników w notatniku technicznym..
- Korzystając z zegara w klasie lub zegarka, zmierz czas i zgłoś go Protokolantowi.
- Odzyskaj każdy obiekt tak szybko, jak to możliwe.

Gratulacje! Zebrałeś, swoim Clawbotem, wszystkie trzy obiekty i zwróciłeś je do bazy domowej za pomocą kontrolera!

Sztafeta!

Celem tego ćwiczenia jest wykorzystanie Clawbota i umiejętności pracy zespołowej w sztafecie. Grupa będzie odpowiedzialna za przesuwanie obiektu po trzymetrowej trasie w jak najkrótszym czasie.

Gracz 1: Wybierz przedmiot i przenieś go na linię jednego metra. Upuść przedmiot.

Gracz 2: Podnieś przedmiot i przenieś go z jedno- na dwumetrową linię. Upuść przedmiot.

Gracz 3: Podnieś przedmiot i przenieś z dwumetrowej linii do mety. Upuść przedmiot w polu bramkowym.

Gracz 4: Monitoruj czas i upewnij się, że Clawbot nie przeszkadza innym kierowcom ani uczniom. Upewnij się, że robot umieścił obiekt wystarczająco daleko za linią.



Dodatkowe pytania - bloki VEXcode V5

Po wykonaniu trzech czynności odpowiedz na poniższe pytania w swoim notatniku inżyniera.

- Blok *ustaw zatrzymanie silnika* jest ustawiony na "wstrzymanie" zarówno dla silnika ramienia, jak i silnika kleszczy. Co by się stało, gdyby te bloki zostały usunięte?
- Bloki obrót i zatrzymanie, które kontrolują silniki ramion i kleszczy, są prawie identyczne. Gdybyś miał stworzyć ten projekt samodzielnie, jak mógłbyś zaoszczędzić czas i uniknąć ciągłego przeciągania każdego bloku do obszaru roboczego?

Programowanie oparte na zdarzeniach: komunikacja między instrukcjami



Programowanie oparte na zdarzeniach

Jeśli Twój pies przynosi Ci smycz lub siada przy drzwiach, daje Ci znać, że musi wyjść na zewnątrz. W szkole, kiedy nauczyciel zadaje pytanie i widzi, jak podnosisz rękę, wie, że uważasz, że znasz odpowiedź i chciałbyś odpowiedzieć na pytanie. Te zachowania są również nazywane "wyzwalaczami".

Twój pies wie, że przyniesienie smyczy lub siedzenie przy drzwiach jest wyzwalaczem, który daje ci znać, że musi wyjść na zewnątrz. Kiedy więc widzisz go siedzącego przy drzwiach ze smyczą, reagujesz, wyprowadzając go na zewnątrz. Podniesienie ręki jest wyzwalaczem, który informuje nauczyciela, że chcesz odpowiedzieć na jego pytanie. Następnie nauczyciel reaguje na wyzwalacz, wzywając cię do odpowiedzi.

Programowanie oparte na zdarzeniach w robotyce ma miejsce, gdy pewne zachowania robota powodują, że robot wykonuje określone czynności lub reaguje na określone wyzwalacze.

Kontroler: Clawbot kontrola

Teraz jesteś gotowy, aby pobrać przykładowy projekt i używać kontrolera do jednoczesnej obsługi Clawbota, jego ramienia i kleszczy!

Upewnij się, że masz wymagany sprzęt i notes inżynieryjny. Otwórz VEXcode V5 Text.

Wymagany sprzęt / oprogramowanie:

llość	Sprzęt / inne przedmioty
1	Clawbot
1	Naładowana bateria robota
1	VEX V5 Radio
1	Kontroler
1	Kabel Tether
1	VEXcode V5 Text
1	Kabel USB (jeśli jest używany komputer)
1	Notes inżyniera

Zanim rozpoczniesz ćwiczenie...

Czy masz gotowy każdy z tych elementów? Konstruktor powinien sprawdzić każdą z następujących kwestii:

- Czy wszystkie silniki i czujniki są podłączone do właściwego portu?
- Czy inteligentne kable są odpowiednio włożone do wszystkich silników?
- Czy bateria jest w pełni naładowana?
- Czy kontroler jest sparowany z mózgiem robota?

Przygotuj się do wyzwań programowanie tekstowe VEXcode V5

Przed rozpoczęciem projektu, musisz wybrać właściwy przykładowy projekt. Przykładowy projekt Clawbot Control zawiera konfigurację silników i czujników Clawbot. Jeśli szablon nie jest używany, twój robot nie uruchomi projektu poprawnie.

- Przejdź do menu plików, Otwórz Przykłady, a następnie wybierz przykładowy projekt *Clawbot Control.*
- Zapisz projekt



• Sprawdź, czy nazwa projektu *Clawbot Control* znajduje się teraz w oknie "pośrodku paska narzędzi. Clawbot jest teraz poprawnie skonfigurowany, a projekt Clawbot Control jest gotowy do użycia.

Teraz spójrz, jak bloki są używane w tym projekcie. W swoich notatnikach inżyniera dokonaj następujących prognoz:

- Co się dzieje, gdy uruchomisz ten projekt? Co Clawbot będzie w stanie zrobić?
- Co by się stało, gdybyśmy nie użyli bloku "nie" w tym projekcie?

Wyzwania: Kontrola Clawbota – programowanie tekstowe

Chwyć przedmiot!

Celem tego działania jest złapanie i uwolnienie obiektu za pomocą kontrolera.

Oto kilka kroków, które poprowadzą Twój zespół:

- Umieść robota swojej grupy na podłodze i upewnij się, że Clawbot ma wystarczająco dużo miejsca, aby się poruszać bez przeszkadzania innym grupom.
- Wypisz kroki, jakie Clawbot będzie musiał wykonać, aby złapać obiekt. Pamiętaj, aby podać przyciski, których będziesz używać do wykonania tego zadania!
- Zapisz czynności wymienione w liście kierowcy w notatniku technicznym.
- Kliknij przycisk Pobierz na pasku narzędzi, aby pobrać projekt Clawbot Control do Mózgu.



- Sprawdź, czy Twój projekt został pobrany do mózgu Clawbota, patrząc na ekran robota. Nazwa projektu, Clawbot Control, powinna być wymieniona w slocie 1.
- Uruchom projekt na Clawbocie.
- Chwyć i uwolnij obiekt za pomocą kontrolera.

Gratulacje! Złapałeś przedmiot swoim Clawbotem za pomocą kontrolera!

Czy były jakieś różnice między twoimi przewidywaniami a działaniami, które wykonałeś podczas ćwiczenia? Jeśli tak, dodaj je do swojego notatnika technicznego.

Kolorowe klejnoty

Celem tej aktywności jest użycie umiejętności Clawbot do zebrania kilku obiektów, jeden po drugim i położenie ich w wybrane miejsce szybciej niż inne grupy w twojej klasie. Powodzenia!



Oto kilka kroków, które poprowadzą Twój zespół:

- Przenieś obiekty swojej grupy na obszar pobierania obiektów, który wyznaczył twój nauczyciel i upewnij się, że Clawbot ma przestrzeń do poruszania się bez przeszkadzania innym grupom.
- Wymień kroki, jakie Clawbot będzie musiał wykonać, aby złapać każdy obiekt i zwrócić go. Pamiętaj, aby podać przyciski, których będziesz używać do wykonania tego zadania!
- Zapisz czynności wymienione w liście sterowników w notatniku technicznym.
- Korzystając z zegara w klasie lub zegarka, zmierz czas i zgłoś go Protokolantowi.
- Odzyskaj każdy obiekt tak szybko, jak to możliwe.

Gratulacje! Zebrałeś, swoim Clawbotem, wszystkie trzy obiekty i zwróciłeś je do bazy domowej za pomocą kontrolera!

Sztafeta!

Celem tego ćwiczenia jest wykorzystanie Clawbota i umiejętności pracy zespołowej w sztafecie. Grupa będzie odpowiedzialna za przesuwanie obiektu po trzymetrowej trasie w jak najkrótszym czasie.

Gracz 1: Wybierz przedmiot i przenieś go na linię jednego metra. Upuść przedmiot.

Gracz 2: Podnieś przedmiot i przenieś go z jedno- na dwumetrową linię. Upuść przedmiot.

Gracz 3: Podnieś przedmiot i przenieś z dwumetrowej linii do mety. Upuść przedmiot w polu bramkowym.

Gracz 4: Monitoruj czas i upewnij się, że Clawbot nie przeszkadza innym kierowcom ani uczniom. Upewnij się, że robot umieścił obiekt wystarczająco daleko za linią.



Dodatkowe pytania programowanie tekstowe

Po wykonaniu trzech czynności odpowiedz na poniższe pytania w swoim notatniku inżyniera.

- Instrukcja *ArmMotor.stop()* jest ustawiony na "wstrzymanie". Co by się stało, gdyby ta instrukcja została usunięta?
- Instrukcje obrót i zatrzymanie, które kontrolują silniki ramion i kleszczy, są prawie identyczne. Gdybyś miał stworzyć ten projekt samodzielnie, jak mógłbyś zaoszczędzić czas i uniknąć ciągłego wpisywania instrukcji w obszarze roboczym?



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

Powtórzenie - bloki VEXcode V5

Wiele osiągnąłeś w tym laboratorium STEM! Poniższe pytania pomogą ci przemyśleć wszystko, czego się nauczyłeś. Możesz odpowiedzieć tylko raz, więc dobrze się zastanów, zanim podasz odpowiedź!

- 1. Które z poniższych stwierdzeń dotyczących bloku zawsze jest prawdziwe?
 - Cały czas powtarza działania robota.
 - Powtarza działania robota określoną liczbę razy.
 - Zatrzyma się po upływie określonego czasu.
 - o Można go używać tylko podczas programowania kontrolera.
- 2. Michał chce zaprogramować swojego Clawbota do ciągłego poruszania się dookoła kwadratowego pokoju bez zatrzymywania się. Którego bloku najlepiej użyć, aby to osiągnąć?
 - Blok poczekaj do
 - o Blok powtórz
 - o Blok zawsze
 - Blok powtarzaj do
- 3. Asia stworzyła projekt, aby jej Clawbot podjechał do przodu, podniósł i opuścił ramię, a następnie skręcił w prawo. Chce czterokrotnie powtórzyć tę serię zachowań. Ile łącznie milimetrów przebył Clawbot po zakończeniu pętli?



- 104100
- o **400**
- o **90**

- 4. Tosia używa bloku *powtórz* zawierającego 4 bloki w środku. Blok *powtórz* ma powtarzać się 11 razy. Jaka jest prawidłowa notacja matematyczna, która opisuje, ile bloków robot przejdzie w projekcie, łącznie z blokiem *powtórz*?
 - 1 + (4 x 11)
 4 x 11
 4 + 11
 1 + 4 + 11
- 5. Które z poniższych słów najlepiej uzupełnia poniższe zdanie: Wszystkie przyciski Joystick, L i R są _____ przeznaczonymi do programowania Clawbot'a poprzez sterowanie zdarzeniami w przykładowym projekcie Clawbot Control.
 - o Wyzwalaczami
 - o Komentarzami
 - o **Pętlami**
 - o Stosami
- 6. Jaką funkcję w poniższym projekcie spełniają ustawione bloki zatrzymujące silnik?



- Zatrzymują ruch silników po zakończeniu pętli.
- Nie pozwalają kontrolerowi poruszać ramieniem i kleszczami.
- Umożliwiają swobodne obracanie się silników.
- Zapobiegają upuszczeniu ramienia i / lub zamknięciu kleszczy, gdy przyciski sterujące ich silnikami są zwolnione.

Powtórzenie – programowanie

tekstowe

Wiele osiągnąłeś w tym laboratorium STEM! Poniższe pytania pomogą ci przemyśleć wszystko, czego się nauczyłeś. Możesz odpowiedzieć tylko raz, więc dobrze się zastanów, zanim podasz odpowiedź!

7. Które z poniższych stwierdzeń dotyczących komendy *zawsze* jest prawdziwe?

- o Cały czas powtarza działania robota.
- Powtarza działania robota określoną liczbę razy.
- o Zatrzyma się po upływie określonego czasu.
- o Można go używać tylko podczas programowania kontrolera.
- 8. Michał chce zaprogramować swojego Clawbota do ciągłego poruszania się dookoła kwadratowego pokoju bez zatrzymywania się. Której komendy najlepiej użyć, aby to osiągnąć?
 - o Komendy poczekaj do
 - o Komendy *powtórz*
 - o Komendy zawsze
 - o Komendy powtarzaj do
- 9. Asia stworzyła projekt, aby jej Clawbot podjechał do przodu, podniósł i opuścił ramię, a następnie skręcił w prawo. Chce czterokrotnie powtórzyć tę serię zachowań. Ile łącznie milimetrów przebył Clawbot po zakończeniu pętli?

21	
22	#include "vex.h"
23	
24	using namespace vex;
25	
26	<pre>int main() {</pre>
27	<pre>// Initializing Robot Configuration. D0 NOT REMOVE!</pre>
28	<pre>vexcodeInit();</pre>
29	
30	repeat(4) {
31	Drivetrain.driveFor(forward, 100, mm);
32	ArmMotor.spinFor(forward, 90, degrees);
33	<pre>ArmMotor.spinFor(reverse, 90, degrees);</pre>
34	<pre>Drivetrain.turnFor(right, 90, degrees);</pre>
35	}
36	}

- o **104**
- o **100**
- o **400**
- o **90**

10. Tosia używa komendy *powtórz* zawierającej 4 komendy w środku. Komenda *powtórz* ma powtarzać się 11 razy. Jaka jest prawidłowa notacja matematyczna, która opisuje, ile komend robot wykona w projekcie, łącznie z komendą powtórz?

```
○ 1 + (4 x 11)
○ 4 x 11
```

- o 4 + 11
- 1+4+11
- 11.Które z poniższych słów najlepiej uzupełnia poniższe zdanie: Wszystkie przyciski Joystick, L i R są _____ przeznaczonymi do programowania Clawbot'a poprzez sterowanie zdarzeniami w przykładowym projekcie Clawbot Control.
 - o Wyzwalaczami
 - o Komentarzami
 - o **Pętlami**
 - o Stosami

12. Jaką funkcję w poniższym projekcie spełniają komendy zatrzymujące silnik?

```
int main() {
   // Initializing Robot Configuration. DO NOT REMOVE!
    vexcodeInit();
    Controller1.ButtonL1.pressed(whenControllerL1Pressed);
    Controller1.ButtonL2.pressed(whenControllerL2Pressed);
    Controller1.ButtonR1.pressed(whenControllerR1Pressed);
    Controller1.ButtonR2.pressed(whenControllerR2Pressed);
    ArmMotor.setStopping(hold);
    ClawMotor.setStopping(hold);
    ArmMotor.setVelocity(60, percent);
    ClawMotor.setVelocity(30, percent);
while (true) {
     LeftMotor.setVelocity(Controller1.Axis3.position(), percent);
     RightMotor.setVelocity(Controller1.Axis2.position(), percent);
     LeftMotor.spin(forward);
     RightMotor.spin(forward);
    }
  }
```

- Zatrzymują ruch silników po zakończeniu pętli.
- Nie pozwalają kontrolerowi poruszać ramieniem i kleszczami.
- Umożliwiają swobodne obracanie się silników.
- Zapobiegają upuszczeniu ramienia i / lub zamknięciu kleszczy, gdy przyciski sterujące ich silnikami są zwolnione.

APPENDIX

Dodatkowe informacje, zasoby i materiały.

Używanie 1 Post Hex Nut Retainer w/ Bearing Flat



1 Post Hex Nut Retainer w/ Bearing Flat

Używanie the 1 Post Hex Nut Retainer w/ Bearing Flat

1 Post Hex Nut Retainer w/ Bearing Flat umożliwia płynne obracanie się wałów przez otwory w elementach konstrukcyjnych. Po zamontowaniu zapewnia dwa punkty styku na elementach konstrukcyjnych w celu zapewnienia stabilności. Na jednym końcu elementu znajduje się słupek o wymiarach umożliwiających bezpieczne dopasowanie do kwadratowego otworu elementu konstrukcyjnego. Środkowy otwór elementu ma taki rozmiar i szczelinę, aby bezpiecznie dopasować nakrętkę sześciokątną, umożliwiając łatwe dokręcenie śruby 8-32 bez użycia klucza lub kombinerek. Otwór na końcu elementu jest przeznaczony do przechodzenia wałków lub śrub.

Aby skorzystać z retainera:

- Wyrównaj go na elemencie konstrukcyjnym VEX tak, aby otwór końcowy znajdował się w żądanym miejscu, a sekcja środkowa i końcowa również były podparte przez element konstrukcyjny.
- Włóż kwadratowy słupek wystający z elementu do elementu konstrukcyjnego, aby pomóc go utrzymać na miejscu.
- Włożyć nakrętkę sześciokątną w środkową część retainera tak aby zrównała się z resztą elementu.
- W stosownych przypadkach wyrównaj wszelkie dodatkowe elementy konstrukcyjne z tyłu głównego elementu konstrukcyjnego.
- Użyj śruby 8-32 odpowiedniej długości, aby przymocować elementy konstrukcyjne do retainera przez środkowy otwór i nakrętkę sześciokątną.

Używanie 4 Post Hex Nut Retainer



4 Post Hex Nut Retainer

Używanie the 4 Post Hex Nut Retainer

The 4 Post Hex Nut Retainer zapewnia pięć punktów styku do tworzenia wytrzymałego połączenia między dwoma elementami konstrukcyjnymi za pomocą jednej śruby i nakrętki. Każdy narożnik retainera zawiera słupek o rozmiarze umożliwiającym bezpieczne dopasowanie go do kwadratowego otworu w elemencie konstrukcyjnym. Środek elementu ma taki rozmiar i szczelinę, aby bezpiecznie dopasować nakrętkę sześciokątną, umożliwiając łatwe dokręcenie śruby bez użycia klucza lub kombinerek.

Aby skorzystać z retainera:

- Wyrównaj go na elemencie konstrukcyjnym VEX tak, aby środkowy otwór znajdował się w żądanym miejscu, a każdy narożnik był wsparty elementem konstrukcyjnym.
- Włóż kwadratowy słupek wystający z elementu do elementu konstrukcyjnego, aby pomóc go utrzymać na miejscu.

- Włożyć nakrętkę sześciokątną w środkową część retainera tak aby zrównała się z resztą elementu.
- W stosownych przypadkach wyrównaj wszelkie dodatkowe elementy konstrukcyjne z tyłu głównego elementu konstrukcyjnego.
- Użyj śruby 8-32 odpowiedniej długości, aby przymocować elementy konstrukcyjne.

Używanie 1 Post Hex Nut Retainer



1 Post Hex Nut Retainer

Używanie the 1 Post Hex Nut Retainer

1 Post Hex Nut Retainer zapewnia dwa punkty styku do łączenia elementu konstrukcyjnego z innym elementem za pomocą jednej śruby i nakrętki. Jeden koniec elementu zawiera słupek o wymiarach umożliwiających bezpieczne dopasowanie do kwadratowego otworu elementu konstrukcyjnego. Drugi koniec retainera ma taki rozmiar i szczelinę, aby bezpiecznie dopasować nakrętkę sześciokątną, umożliwiając łatwe dokręcenie śruby 8-32 bez użycia klucza lub kombinerek.

Aby skorzystać z retainera:

- Wyrównaj go na elemencie konstrukcyjnym VEX tak, aby oba końce były podparte przez element konstrukcyjny i ustawione tak, aby zamocować drugi element.
- Włóż kwadratowy słupek wystający z retainera do elementu konstrukcyjnego, aby pomóc go utrzymać na miejscu.
- Jeśli element jest używany do mocowania dwóch elementów konstrukcyjnych, włóż nakrętkę sześciokątną do drugiego końca retainera, tak aby był wyrównany z resztą

elementu. W przypadku użycia do zamocowania innego rodzaju elementu, takiego jak np.: element dystansowy, może być właściwe włożenie śruby przez tę stronę.

- Jeśli jest to niezbędne, wyrównaj wszelkie dodatkowe komponenty z tyłu głównego elementu konstrukcyjnego.
- Jeśli retainer jest używany do łączenia dwóch elementów konstrukcyjnych, użyj śruby 8-32 o odpowiedniej długości, aby zabezpieczyć elementy konstrukcyjne przez otwór i nakrętkę sześciokątną. Jeśli jest używany do łączenia innego typu elementu, takiego jak np.: element dystansowy, zabezpiecz go bezpośrednio lub za pomocą nakrętki sześciokątnej.

Notes inżyniera

March 107 1876 see you " To my delight he came and declared That he was heard and understood what I said . MD I asked him to repeat the words - the mint He areneved "you said " M. Watson - come here. receiving with I want to see you " . We then changed places and I listened at S while Withow read a few passages from a book into the month piece M. It was cutainly The case That articulate sounds proceeded from S. The 1. The improved instrument shown in Fig. I was effect was loud but indistinct and muffled . If I had read beforehand The passage given by Mr Watcon I should have recognized constructed This morning and tried This latting . Pis a trass pipe and W The platimum wire every word. As it was I could not M the month file and S The armatine of The Receiving Instrument . M. Watson was stationed in one room word here and there was quite distinct. I made oat "to" and "out" and "further"; with the Receiving Listrument . He pressed ous ear closely against S and closely his other and finally the sentence "W" Bell Do you welentand what I say? Do-you - un ear with his hand. The Transmitting Instrument der - stand - what - I - say " came was placed in another room and the doors of both rooms were closed. I then shorted into M the following sentence: "W" Watson - Come here - I want to was andifle when The armature S was reneoved .

Notatki Alexandra Grahama Bell'a z udanego eksperymentu z jego pierwszym telefonem

Notatnik inżyniera dokumentuje Twoją pracę

Nie tylko używasz notatnika do organizowania i dokumentowania swojej pracy, ale jest to także miejscem do refleksji nad działaniami i projektami. Podczas pracy w zespole każdy członek zespołu będzie prowadził własny dziennik, aby ułatwić współpracę w grupie.

Twój notatnik inżyniera powinien mieć następujące elementy:

- Wpis na każdy dzień lub sesję, w której pracowałeś nad rozwiązaniem
- Wpisy chronologiczne, z datą każdego wpisu
- Jasne, schludne i zwięzłe notatki, dobrze zorganizowane
- Etykiety, aby czytelnik zrozumiał wszystkie Twoje notatki i ich dopasowanie do iteracyjnego procesu projektowania

Wpis może obejmować:

- Burzę mózgów
- Szkice lub zdjęcia prototypów
- Pseudokod i schematy blokowe planowania
- Wszelkie zastosowane obliczenia lub algorytmy
- Odpowiedzi na pytania przewodnie
- Uwagi dotyczące obserwacji i / lub przeprowadzonych testów
- Notatki i refleksje na temat różnych iteracji