

Wyzwania i szanse nowej ery  
Podręcznik dla nauczycieli szkół podstawowych -  
uczniowie zaawansowani

# ARTIE

# ARTIE

## Wyzwania i szanse nowej ery: Podręcznik dla nauczycieli szkół podstawowych - uczniowie zaawansowani

### Autorzy

Željko Krnajić  
Janko Radigović

### Wydawca

Hrvatski robotički savez, Chorwacja  
„ARTIE: Sztuczna inteligencja w edukacji – wyzwania i szanse nowej ery: opracowanie nowego programu nauczania, przewodnika dla nauczycieli i kursu online dla uczniów”, Projekt współfinansowany przez Unię Europejską w ramach Programu Erasmus+, numer projektu 2020-1-HR01-KA201-077800

### Konsultanci/Recenzenci

Katarzyna Garbacik  
Andrzej Garbacik  
Bogusław Klimczuk  
Ivana Ružić  
Ana Pina  
Christina Eirini Karvouna

### Projekt graficzny i ilustracje

Christina Eirini Karvouna

### Tłumaczenia

Jura Cmrečak (Chorwacki)  
Bogusława Denys (Polski)  
Ana Pina (Portugalski)  
Christina Eirini Karvouna (Holenderski)



3	Sztuczna inteligencja w robotyce
11	Wykrywanie i rozpoznawanie twarzy w Scratchu
19	Programowanie wykrywania twarzy w Scratchu
26	Programowanie rozpoznawania twarzy w Scratchu
41	Wykrywanie i klasyfikacja obiektów dla początkujących
49	Programowanie wykrywania obiektów w Scratchu
57	Projekt klasyfikacji obiektów
63	Rozpoznawanie i generowanie mowy w Scratchu dla początkujących
68	Programowanie rozpoznawania mowy w Scratchu
73	Programowanie generowania mowy w Scratchu
77	Projekt z obiektem sterowanym głosem
83	Wprowadzenie do sprzętu - mikrokontroler, kamera i sterownik silnika
93	Podłączanie mikrokontrolera do kamery i komputera, rozpoczęcie pracy
100	Składanie robota
119	Programowanie robotów - poruszanie się
128	Programowanie robotów - śledzenie obiektów
140	Programowanie robota - podążanie za linią



# Scenariusz lekcji 1

**Czas (minuty):**  
**90**

## Tematyka

AI w robotyce  
roboty

## Cele

Zapoznanie się i  
rozumienie AI w robotyce

## Oczekiwane efekty

Zrozumienie, w jaki  
sposób sztuczna  
inteligencja jest  
wykorzystywana w  
robotyce

## Sztuczna inteligencja w robotyce

Uczniowie poznają roboty wykorzystujące sztuczną inteligencję i ich aktualne zastosowanie.

**Nauczyciel przedstawia temat i rozpoczyna dyskusję:**

Czy sztuczna inteligencja w robotyce może zmienić przyszłość?  
Czym jest sztuczna inteligencja w robotyce?  
Czy używamy już robotów z AI w naszym codziennym życiu?  
Jak? Gdzie?

### Prezentacja celu głównego lekcji:

Celem dzisiejszej lekcji jest poznanie, czym jest sztuczna inteligencja w robotyce i jak można ją wykorzystać w naszym codziennym życiu.



<http://erasmus-artie.eu>



# CZĘŚĆ GŁÓWNA

## Zagadnienia do dyskusji:

Czym jest robot?

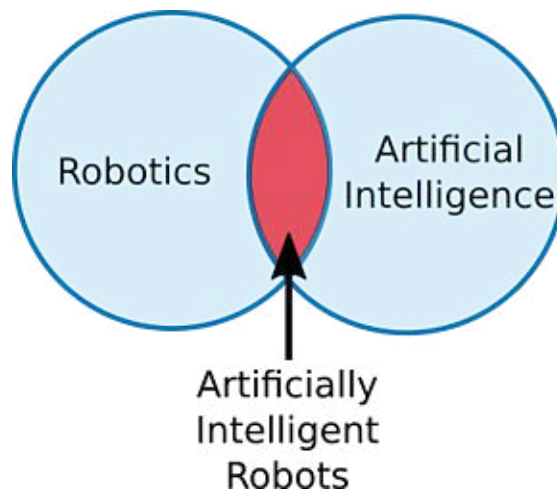
Czy każdy robot obsługuje AI?

Czy znasz przykłady sztucznej inteligencji w robotyce?

Czy roboty ze sztuczną inteligencją mogą zastąpić ludzi?

Pierwszą rzeczą, którą warto wyjaśnić, jest to, że robotyka i sztuczna inteligencja to wcale nie to samo. W rzeczywistości te dwa pola są prawie całkowicie oddzielne.

Diagram Venna dwóch pól wyglądałby tak:



Istnieje mały obszar, w którym te dwie dziedziny się pokrywają: sztucznie inteligentne roboty. W ramach tego nakładania się ludzie czasami myślą te dwa pojęcia.

Czym jest sztuczna inteligencja?

Sztuczna inteligencja (AI) to gałąź informatyki. Polega na opracowywaniu programów komputerowych do wykonywania zadań, które w przeciwnym razie wymagałyby ludzkiej inteligencji. Algorytmy sztucznej inteligencji mogą zajmować się uczeniem, percepcją, rozwiązywaniem problemów, rozumieniem języka i/lub logicznym rozumowaniem. Według ojca Sztucznej Inteligencji, Johna McCarthy'ego, jest to również „Nauka i inżynieria tworzenia inteligentnych maszyn, zwłaszcza inteligentnych programów komputerowych”.



Mówiąc najprościej, sztuczna inteligencja to sposób na sprawienie, by **komputer, sterowany komputerowo robot lub oprogramowanie myślało inteligentnie**, w podobny sposób, jak myślą inteligentni ludzie.

### Cele AI

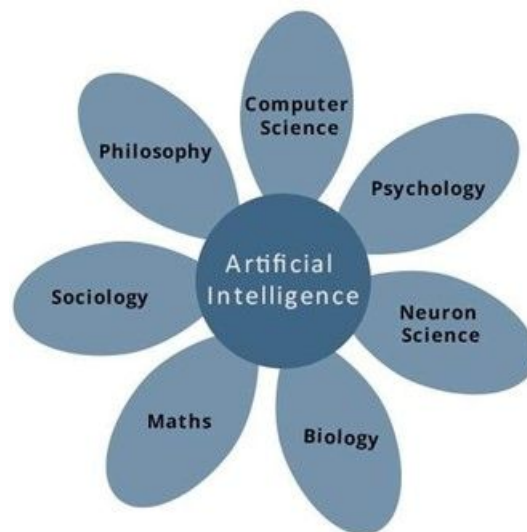
**tworzyć systemy eksperckie** – systemy te wykazują inteligentne zachowanie, uczą się, demonstrują, wyjaśniają i doradzają swoim użytkownikom.

**zaimplementować ludzką inteligencję w maszynach** – takie systemy rozumiałyby, myślały, uczyły się i zachowywały jak ludzie.

### Co składa się na sztuczną inteligencję?

Sztuczna inteligencja to nauka i technologia oparta na dyscyplinach takich jak informatyka, biologia, psychologia, językoznawstwo, matematyka i inżynieria. Głównym celem sztucznej inteligencji jest rozwój funkcji komputerowych związanych z ludzką inteligencją, takich jak rozumowanie, uczenie się i rozwiązywanie problemów.

Spośród poniższych obszarów jeden lub wiele obszarów może przyczynić się do zbudowania inteligentnego systemu.



### Czym jest robotyka?

Robotyka to gałąź sztucznej inteligencji, która składa się z inżynierii elektrycznej, inżynierii mechanicznej i informatyki do projektowania, budowy i stosowania robotów.

### Czym są roboty?

Roboty to sztuczne postaci działające w realnym środowisku.

Są to programowalne maszyny, które zwykle są w stanie wykonać szereg działań autonomicznie lub półautonomicznie.



## Cel

Roboty są opracowywane tak, aby miały zdolność manipulowania obiektami poprzez postrzeganie, wybieranie, przesuwanie, modyfikowanie właściwości fizycznych obiektu, niszczenie go lub uwalnianie siły roboczej od wykonywania powtarzalnych zadań bez znużenia, rozproszenia lub wyczerpania.

## Aspekty robotyki

Roboty mają **konstrukcję mechaniczną**, formę lub kształt zaprojektowane do wykonania określonego zadania.

Posiadają **komponenty elektryczne**, które zasilają i kontrolują maszynę.

Zawierają pewien poziom **programu komputerowego**, który określa, co, kiedy i jak robot coś robi.

## Sztuczna inteligencja w robotyce

Sztuczna inteligencja w robotyce pomaga robotom wykonywać kluczowe zadania z wizją podobną do ludzkiej, aby wykrywać lub rozpoznawać różne obiekty. Roboty są opracowywane poprzez uczenie maszynowe i szkolenie, a ogromna liczba zestawów danych jest wykorzystywana do trenowania komputerowego modelu widzenia, dzięki czemu roboty mogą rozpoznawać różne obiekty, odpowiednio wykonywać działania i osiągać pożądane wyniki. Wizja komputerowa to po prostu proces postrzegania obrazów i filmów dostępnych w formatach cyfrowych. Sztuczna inteligencja w robotyce nie tylko pomaga nauczyć model wykonywania określonych zadań, ale także sprawia, że maszyny są bardziej inteligentne, dzięki czemu mogą działać w różnych scenariuszach.

Oto kilka przykładów najbardziej zaawansowanych robotów humanoidalnych, przemysłowych i usługowych, które zmieniają przyszłość z pomocą sztucznej inteligencji.

## Sophia

Najbardziej zaawansowany, przypominający człowieka robot firmy Hanson Robotics, Sophia, uosabia nasze marzenia o przyszłość sztucznej inteligencji. Jako wyjątkowe połączenie nauki, inżynierii i sztuki, Sophia jest jednocześnie stworzoną przez człowieka postacią science fiction, przedstawiającą przyszłość sztucznej inteligencji i robotyki, a także platformą do zaawansowanej robotyki i badań nad sztuczną inteligencją.

Postać Sophii pobudza wyobraźnię widzów na całym świecie. Jest pierwszym na świecie robotem-obywatelem i pierwszym ambasadorem innowacji robotów w Programie Narodów Zjednoczonych ds. Rozwoju. Sophia jest obecnie powszechnie znana, występowała w programach Tonight Show i Good Morning Britain, a także przemawiała na setkach konferencji na całym świecie.

Poznaj Sophię: <https://www.youtube.com/watch?v=BhU9hOo5Cuc>





### Digit

Digit ma pomagać w opiece nad ludźmi w ich domach, pomagać w reagowaniu na katastrofy i dostarczać paczki do drzwi wejściowych. Dzięki zwinnym kończynom i torsowi wypełnionemu czujnikami Digit może poruszać się w złożonych środowiskach i wykonywać zadania, takie jak dostarczanie paczek. W maju 2019 r. Ford Motor Company i Agility ogłosiły partnerstwo w celu opracowania rozwiązania logistycznego, które łączy technologię pojazdów autonomicznych Forda i Agility Digit.



### Pepper

Pepper to pierwszy na świecie humanoidalny robot społeczny, który potrafi rozpoznawać twarze i podstawowe ludzkie emocje. Pepper został przyjęty przez ponad 2000 firm na całym świecie. Idealny w handlu detalicznym i finansach, Pepper ma wiele funkcji, w tym zwiększanie ruchu w sklepie poprzez przyciąganie uwagi kupujących, tworzenie niezapomnianych wrażeń w sklepie, stymulowanie zakupów i utrzymanie klientów. Pepper może również gromadzić obszerne dane w celu wzbogacenia bazy klientów i generowania spostrzeżeń dotyczących kupujących.



<http://erasmus-artie.eu>

### Atlas

Atlas to najbardziej dynamiczny humanoidalny robot na świecie zbudowany przez BostonDynamics, firmę, która wcześniej należała do Google, a obecnie do SoftBanku. Atlas z roku na rok staje się coraz bardziej wyrafinowany dzięki najnowocześniejszemu sprzętowi i algorytmowi, który pozwala mu szybko zrozumieć instrukcje. Dzięki 28 przegubom hydraulicznym, wysokości 4,9 stopy i wadze 176 funtów, robot może wykonywać zarówno imponujące, jak i przerażające czynności, w tym poruszanie się po nierównym terenie, skakanie po torze parkour i wykonywanie salta. Wszystkie te czynności wykazują zwinność na poziomie człowieka, więc robot może być idealny do akcji poszukiwawczych i ratowniczych oraz wykonywania zadań ludzkich w środowiskach, w których ludzie nie mogliby przetrwać.







### Spot

Spot to pies-robot przeznaczony do zastosowań przemysłowych, takich jak przenoszenie towarów przez magazyn i inspekcja odległego miejsca o niekorzystnym środowisku dla ludzkich operatorów. Może działać z prędkością 5,2 stopy na sekundę, ma kamery 360 stopni i może działać w temperaturach od 4 do 113 Fahrenheita. Dzięki interfejsowi API i elastycznemu interfejsowi ładunku robota można łatwo dostosować do żądanych zadań. Spot jest również produkowany przez BostonDynamics i jest obecnie wynajmowany kwalifikującym się do tego firmom.



f

### HRP-5P

HRP-5P to zaawansowany robot humanoidalny zaprojektowany do autonomicznej pracy i wykonywania ciężkich prac w niebezpiecznych środowiskach. Jest wyposażony w czujniki środowiskowe i rozpoznawanie obiektów, planowanie i kontrolę ruchu całego ciała oraz zarządzanie opisem i wykonaniem zadań. HRP-5P opiera się na ponad 20 latach badań nad humanoidami w AIST. W ciągu tych 20 lat instytut stworzył 4 inne roboty będące poprzednikami HRP-5P.



<http://erasmus-artie.eu>

### Surena IV

Surena IV to czwarta generacja serii robotów humanoidalnych Surena opracowana przez Uniwersytet w Teheranie w Iranie. Ten robot o wysokości 5,6 stopy i wadze 154 funtów jest w stanie chodzić z prędkością 0,43 mili na godzinę. Niestandardowe czujniki siły na spodzie stóp pomagają robotowi przechodzić przez nierówne powierzchnie, dostosowując kąt i pozycję każdej stopy.





### Aquanaut

Aquanaut to zaawansowany bezzałogowy transformator podwodny, który może przekształcić się ze zwinnej łodzi podwodnej do dalekiego zasięgu w pół-humanoidalnego robota zdolnego do wykonywania skomplikowanych podwodnych zadań manipulacyjnych. Zaprojektowany przez Houston Mechatronics Inc, Aquanaut może kontrolować podmorską infrastrukturę naftową i gazową, obsługiwać zawory i używać narzędzi podwodnych za pomocą zaledwie kilku kliknięć myszką. Działając całkowicie swobodnie i bez statków pomocniczych, Aquanaut może podróżować ponad 124 mil w trybie łodzi podwodnej, ma maksymalną prędkość 7 węzłów i maksymalną głębokość operacyjną 984 stóp.



### Stuntronic robot

Stuntronic robot to animatroniczny kaskader zaprojektowany do zabawiania tłumów w parkach rozrywki i kurortach Disneya. Dzięki wbudowanym wyrafinowanym czujnikom może podejmować własne decyzje w czasie rzeczywistym – a wszystko to podczas lotu na wysokości 60 stóp. Wie, kiedy podwinąć kolana, aby wykonać salto, kiedy wyciągnąć ramiona, aby skrócić, a nawet kiedy zwolnić obrót, aby mieć pewność, że wyląduje idealnie.



### Handle

Handle to kolejny robot od Boston Dynamics. Dzięki oprogramowaniu wizyjnemu do głębokiego uczenia się robot ten może identyfikować i lokalizować pudła, rozładowywać ciężarówki, paletyzować i depaletyzować za naciśnięciem jednego przycisku. Jego mobilność umożliwia pracę w wielu gniazdach roboczych, poruszanie się po obiektach wraz z przepływem towarów. Może podnieść do 360 pudełek na godzinę.





Pokaż uczniom wideo (opcjonalnie) i przedyskutuj;  
<https://www.youtube.com/watch?v=Jky9I1ihAkg>

Dziś wykorzystujemy sztuczną inteligencję w robotyce w służbie zdrowia, rolnictwie, przemyśle motoryzacyjnym, w magazynach, w łańcuchu dostaw...itp.

Później zbudujemy własnego robota ze sztuczną inteligencją i wyszkolimy go w wykrywaniu i rozpoznawaniu twarzy, wykrywaniu obiektów i rozpoznawaniu mowy.

Oto robot mobilny z możliwościami kamery i sztucznej inteligencji opracowanymi specjalnie na potrzeby tego projektu.

Pokażemy Wam, jak go wykonać i wykorzystać do:

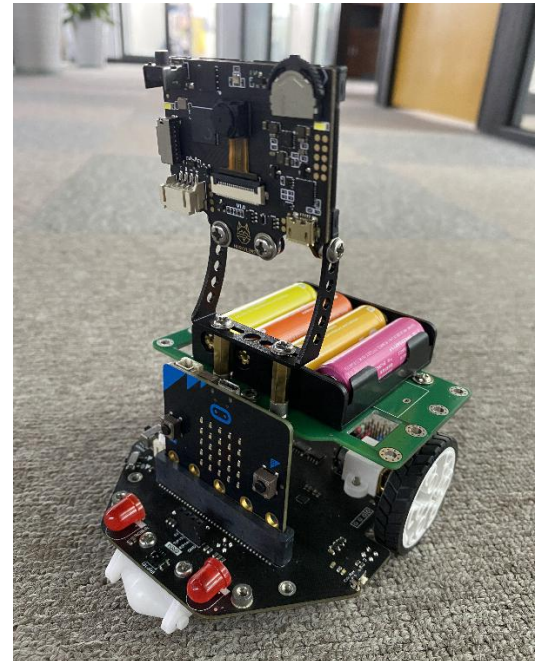
- Wykrywania twarzy
- Rozpoznawania twarzy
- Śledzenia twarzy
- Wykrywania obiektów
- Śledzenia obiektów
- Śledzenia linii

Sztuczna inteligencja wreszcie jest tutaj i większość z nas już aktywnie korzysta z niej w naszym codziennym życiu (nawet nie wiedząc o tym). Przyszłe pokolenia muszą przede wszystkim zrozumieć, jak korzystać z AI! Tylko wtedy mogą wykorzystywać sztuczną inteligencję do ułatwiania nauki i rozwiązywania rzeczywistych problemów.

Sztuczna inteligencja (AI) i robotyka są dzisiaj silnie powiązane.

Sztuczna inteligencja w robotyce jest coraz częściej wykorzystywana w życiu codziennym i odgrywa kluczową rolę w różnych dziedzinach, takich jak przemysł, wojsko, medycyna, badania, rozrywka.

Pamiętajcie, że sztuczna inteligencja jest najprawdopodobniej najpotężniejszą technologią, jaką kiedykolwiek wymyślił człowiek. Może być używana zarówno do dobrych, jak i złych rzeczy. Ostatecznie jednak, to od nas zależy, jak ją wykorzystamy.



## PODSUMOWANIE

Sztuczna inteligencja to sposób, w jaki komputer, robot sterowany przez komputer lub program myśli inteligentnie, podobnie jak w sposób inteligentny myślą ludzie.



## Scenariusz lekcji 2

**Czas (minuty):**

**90**

### Tematyka

Wykrywanie i rozpoznawanie twarzy w Scratchu dla początkujących

### Cele

Nauka wykrywania i rozpoznawania twarzy w Scratchu dla początkujących

### Oczekiwane efekty

Zapoznanie z wykrywaniem i rozpoznawaniem twarzy na prostych przykładach w Scratchu,  
Zrozumienie różnicy między wykrywaniem twarzy a rozpoznawaniem twarzy

## Wykrywanie i rozpoznawanie twarzy w Scratchu dla początkujących

Należy poprosić uczniów o zdefiniowanie, czym jest rozpoznawanie twarzy i wykrywanie twarzy. Należy dać im możliwość opisanie różnicy.

Terminy wykrywanie twarzy i rozpoznawanie twarzy są czasami używane zamiennie, ale istnieją pewne kluczowe różnice. Aby pomóc w wyjaśnieniu problemu, przyjrzyjmy się terminowi wykrywanie twarzy i tym, czym różni się on od terminu rozpoznawanie twarzy.

#### Przedstawienie celu głównego lekcji:

Wprowadzenie do wykrywania i rozpoznawania twarzy dla początkujących na przykładach różnych aplikacji.



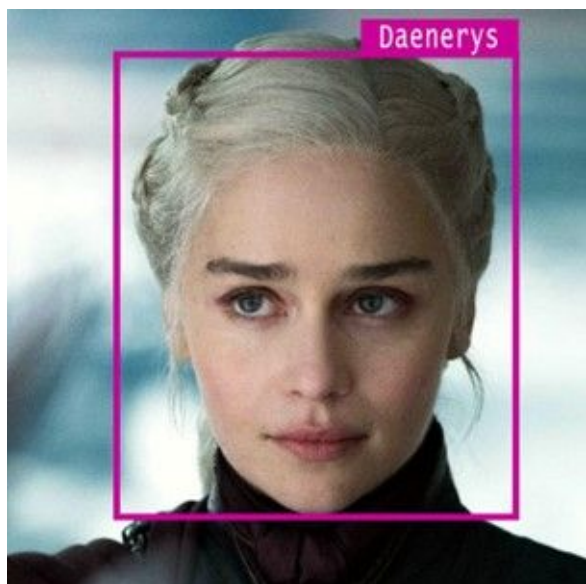
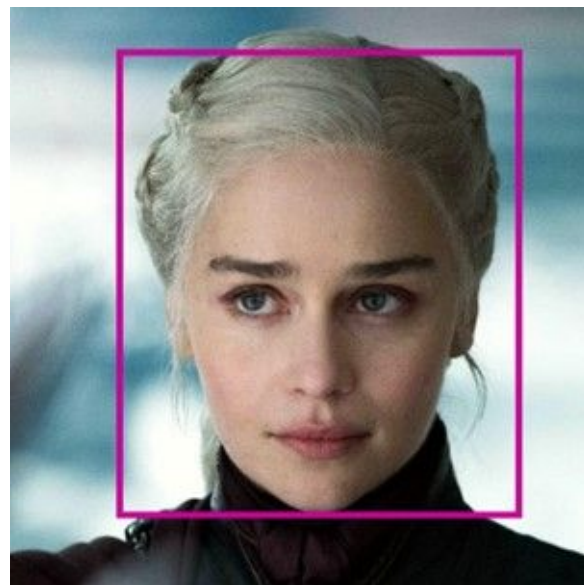
<http://erasmus-artie.eu>

11



# CZĘŚĆ GŁÓWNA

Wykrywanie twarzy to proces, który określa obecność twarzy na zdjęciu lub w klipie wideo. Na przykład taka funkcja jest dostępna w większości oprogramowania aparatu w smartfonie. Ale moduł wykrywania twarzy nie określa, czyja twarz znajduje się w kadrze.



Funkcja wykrywania twarzy nie zapamiętuje ani nie zapisuje rysów twarzy. Jeśli oprogramowanie wykryje twarz jakiejś konkretnej osoby w kadrze, a później znajdzie tę samą twarz na innym zdjęciu, nie ustali, czy twarz należy do tej samej osoby; po prostu wykryje obecność twarzy w ramkach. Oprogramowanie może dostarczać dane dotyczące wieku i płci osoby na każdej klatce, ale nic poza tym. Oprogramowanie do wykrywania twarzy nie może rozpoznać określonych osób. Natomiast rozpoznawanie twarzy dotyczy identyfikacji i rozpoznawania osób.

Celem oprogramowania do rozpoznawania twarzy jest identyfikacja osób występujących na zdjęciu lub filmie wideo poprzez porównanie go z bazą danych. Aby zapewnić pomyślną identyfikację, odpowiednie twarze muszą zostać najpierw wprowadzone do bazy danych.

Oprogramowanie określa unikalne cechy twarzy, zapisuje je i wykorzystuje do późniejszej identyfikacji. Później, podczas procesu identyfikacji, oprogramowanie porówna unikalne cechy i zidentyfikuje twarz konkretnej osoby, jeśli cechy te będą pasować.

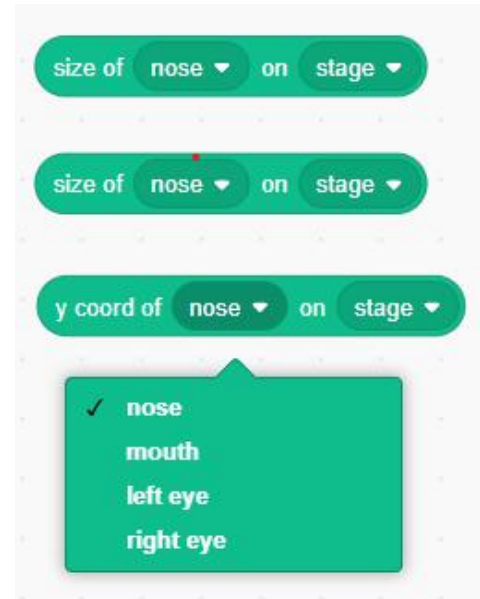


### Scratch/aplikacje bazujące na Scratchu i inne

#### Scratch (ML4KIDS)

<https://machinelearningforkids.co.uk/scratch3/>

Dostępne jest rozszerzenie wykrywania twarzy z 3 blokami typu reporter. Jeśli używasz kamery internetowej jako źródła, połącz ją z rozszerzeniem wykrywania wideo, aby włączyć/wyłączyć wideo z kamery i ustawić przejrzystość.

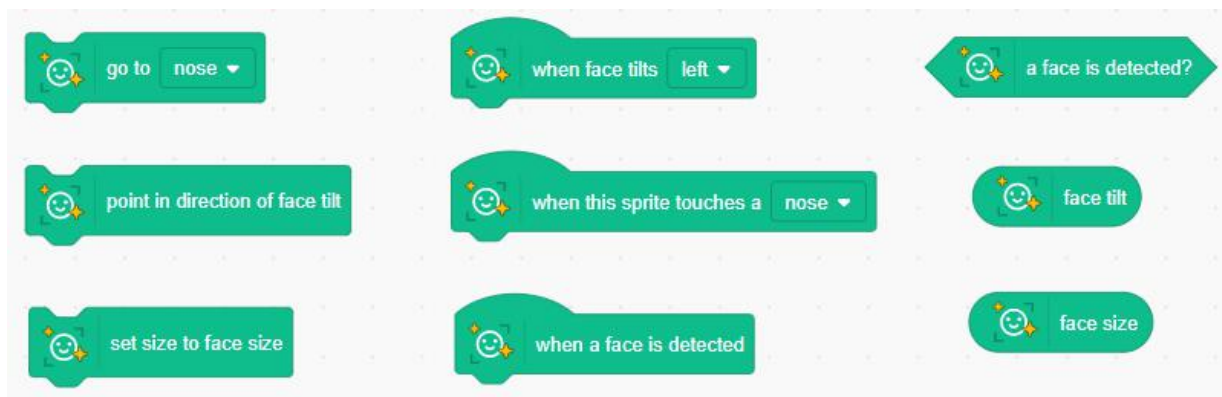


#### Scratch (MIT) - <https://lab.scratch.mit.edu/face/>

Kliknij „Wypróbuj”, a będziesz mieć 9 bloków do obsługi rozpoznawania twarzy.



<http://erasmus-artie.eu>



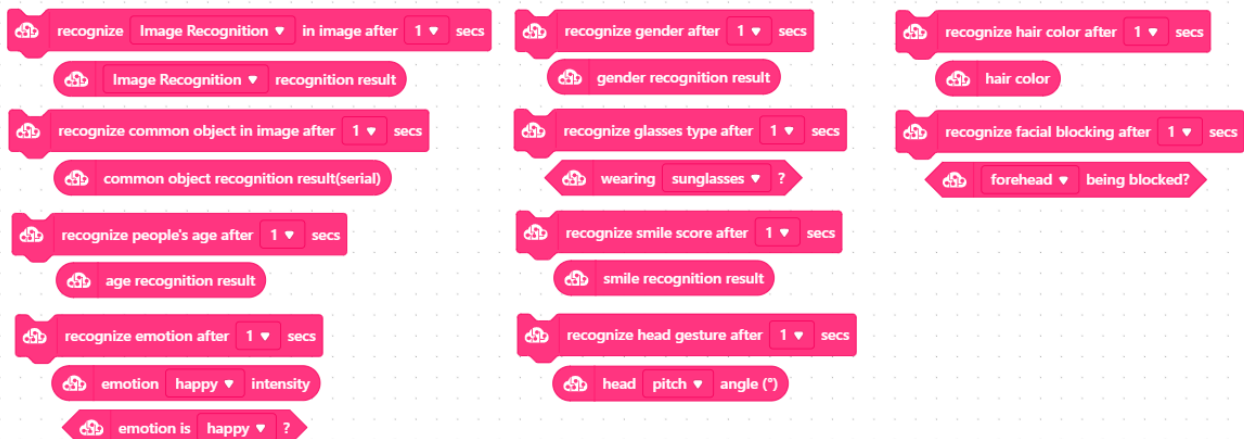


**Scratch (MITMEDIALAB)** - <https://mitmedialab.github.io/prg-extension-boilerplate/create/>  
 Załaduj rozszerzenie wykrywania twarzy, a będziesz mieć 9 bloków do obsługi wykrywania twarzy. Niektóre bloki służą do rozpoznawania mimiki twarzy i emocji. Możesz także użyć rozszerzenia Teachable machine w połączeniu z Google Teachable machine.



**Makeblock (mBlock)** - <https://ide.mblock.cc/>

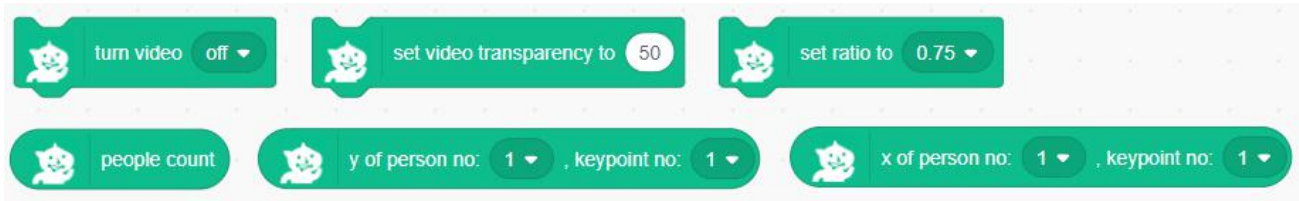
Załaduj usługi Cognitive Services i rozszerzenia Video Sensing, a będziesz mieć różnorodne bloki. Nie ma określonego bloku do wykrywania twarzy, ale można wykryć osobę wewnątrz bloku „rozpoznaj”. Istnieje wiele blokad, które poradzą sobie z emocjami, wiekiem, płcią, uśmiechem, kolorem włosów, okularami, a nawet sytuacją, gdy zakrywasz część twarzy. **Makeblock** zapewnia również rozszerzenie Teachable machine (nie mylić z Google), w którym można trenować do 3 klas i używać go do rozpoznawania twarzy lub wykrywania obiektów.





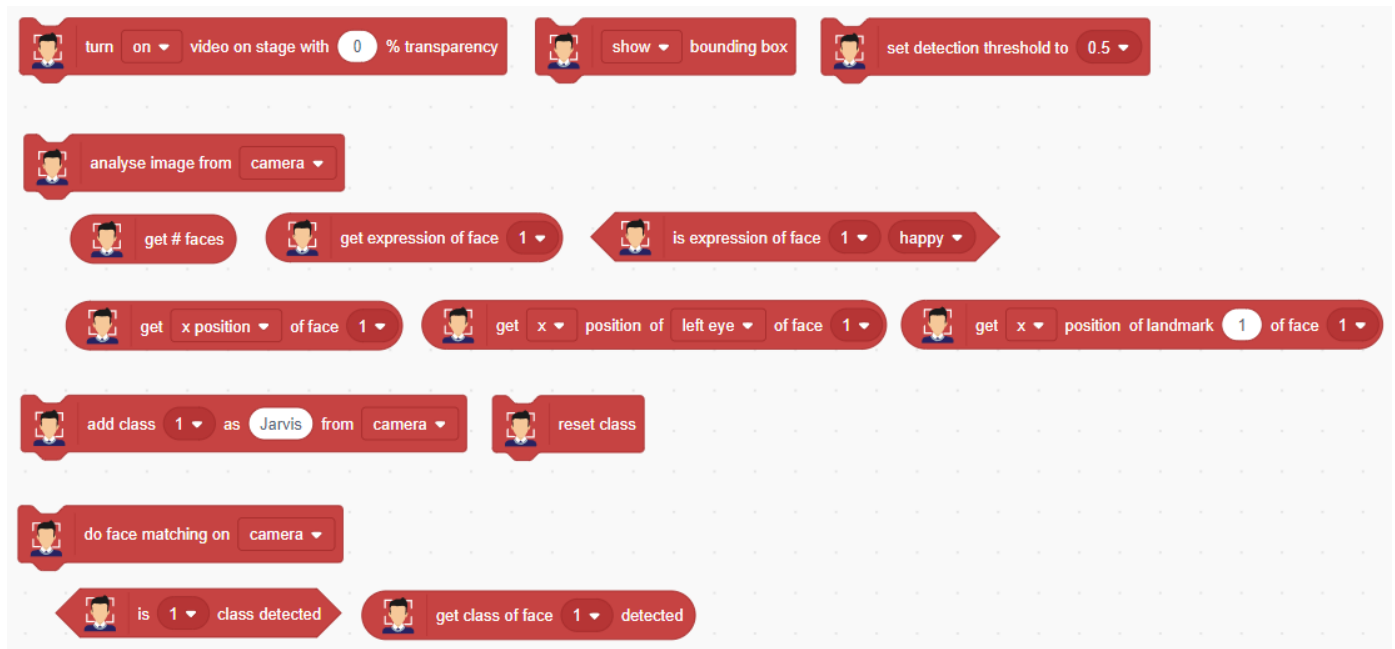
**Stretch3 (github.io)** - <https://stretch3.github.io/>

Załaduj rozszerzenie **Facemesh2Scratch**, aby użyć 3 bloków do wykrywania twarzy (istnieją dodatkowe 3 bloki do obsługi wideo). Główną cechą jest możliwość wykrywania wielu twarzy, dzięki czemu możesz wykryć więcej niż jedną osobę w strumieniu z kamery.



Ostatnim przykładem do wykorzystania jest **PictoBlox**, aplikacja typu desktop, którą należy najpierw zainstalować z <https://thetempedia.com/product/pictoblox/download-pictoblox/> (427 Mb)

Po instalacji użyj rozszerzenia Face Detection, a zobaczysz prawdziwy skarb - możesz wykryć wiele twarzy i ich mimikę. Dostępna jest również funkcja szkolenia klas, która prowadzi nas bezpośrednio do rozpoznawania twarzy.

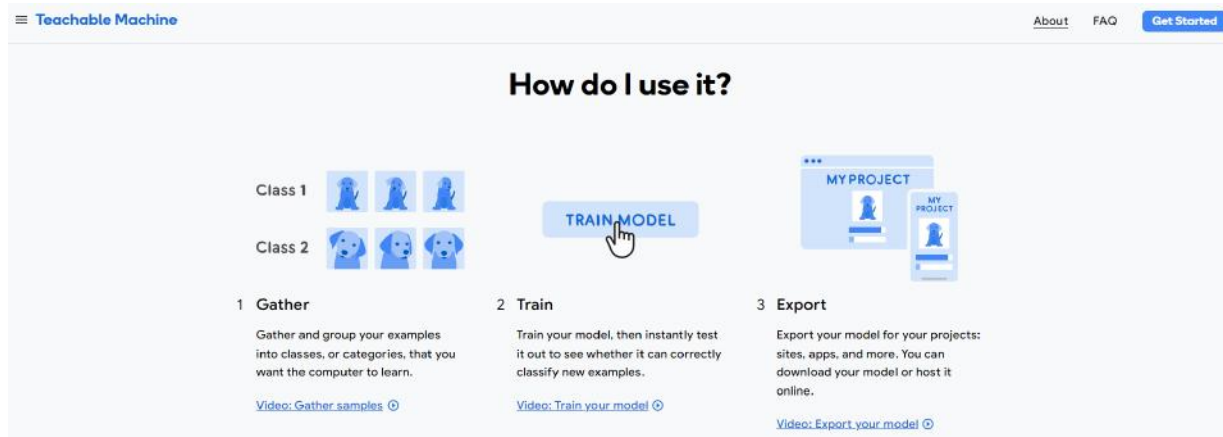






## Teachable machine (Google) - <https://teachablemachine.withgoogle.com/>

Ta aplikacja służy do trenowania Twojego modelu i używania go do rozpoznawania twarzy w połączeniu z rozszerzeniem Teachable machine dostępnym w Scratch (MITMEDIALAB)

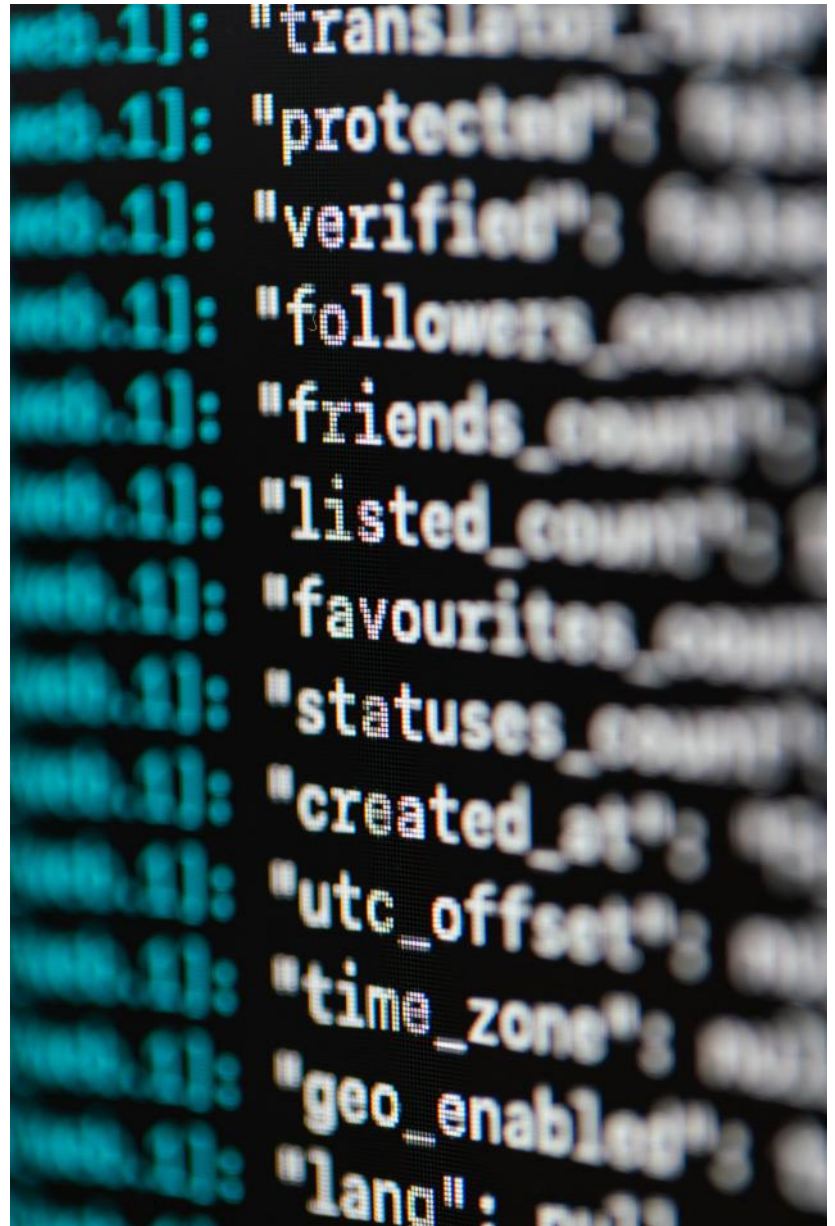


Widzieliśmy trzy różne aplikacje, ale także bardzo podobną do wykrywania i rozpoznawania twarzy.

Wykrywanie twarzy różni się od rozpoznawania twarzy (terminy te nie powinny być używane zamiennie) tym, że wykrywanie twarzy polega jedynie na wykryciu twarzy na cyfrowym obrazie lub wideo. Oznacza to po prostu, że system wykrywania twarzy może rozpoznać ludzką twarz na obrazie lub wideo – nie może zidentyfikować osoby. Wykrywanie twarzy jest elementem systemów rozpoznawania twarzy – pierwszym etapem rozpoznawania twarzy jest wykrycie jej obecności. Wykrywanie twarzy może być również wykorzystywane w aparatach do automatycznego ustawiania ostrości – prawdopodobnie zauważyłeś, że w niektórych aparatach cyfrowych i telefonach wokół twarzy osób wykrytych na zdjęciu pojawia się mała ramka, co pozwala aparatowi ustawić na nich priorytet ostrości. Identyfikacja obecności ludzkiej twarzy odbywa się za pomocą formuł i algorytmów. Zazwyczaj pierwszą rzeczą, której szuka system wykrywania twarzy, są oczy, ponieważ są to jedne z najłatwiejszych do zidentyfikowania cech. Wtedy może również wyszukiwać obecność ust, brwi, nosa i nozdrzy. Wykrywanie twarzy jest ważną częścią procesu rozpoznawania twarzy, jednak z punktu widzenia bezpieczeństwa nie ma niezależnej korzyści z systemu wykrywania twarzy – po prostu rozpoznaje twarz, ale nie ma pojęcia o jej tożsamości. Rozpoznawanie twarzy odgrywa istotną rolę w wielu gałęziach przemysłu, zwłaszcza w kontroli granicznej i egzekwowaniu prawa. Dokładna identyfikacja osób pomaga poprawić bezpieczeństwo na lotniskach oraz w miastach na całym świecie, a można to zrobić tylko za pomocą wiodących na rynku systemów rozpoznawania twarzy.



Jak widać, technologia rozpoznawania twarzy zmienia świat, w którym żyjemy, i wydaje się, że jesteśmy na początku, jeśli chodzi o potencjalne zastosowania oprogramowania do rozpoznawania twarzy. Chociaż zastosowania rozpoznawania twarzy mogą wydawać się nieskończone, musimy upewnić się, że ta technologia jest używana odpowiednio i odpowiedzialnie.



## **PODSUMOWANIE**

Wykrywanie twarzy to proces określający obecność twarzy na obrazie lub w klipie wideo, podczas gdy rozpoznawanie twarzy odnosi się do identyfikacji i rozpoznawania osób.



## Scenariusz lekcji 3

Czas (minuty):  
90

### Tematyka

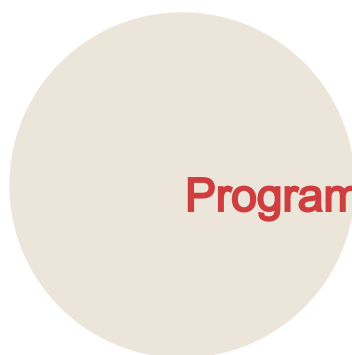
Programowanie wykrywania twarzy w Scratchu

### Cele

Nauka programowania wykrywania twarzy na podanych przykładach

### Oczekiwane efekty

Umiejętność napisania programu do wykrywania twarzy za pomocą Scratcha



## Programowanie wykrywania twarzy w Scratchu

Jak zaprogramować wykrywanie twarzy w Scratchu?

Przypomnimy sobie, czego dowiedzieliśmy się o wykrywaniu twarzy na ostatnich zajęciach.

Zapytaj uczniów o ich doświadczenia z wykrywaniem twarzy. Zanim ich wprowadzisz, zapytaj, czy wiedzą, jak wykonać program do wykrywania twarzy.

Nauczyciel wprowadza uczniów w programowanie wykrywania twarzy w Scratchu. Przejdźmy przez kilka prostych przykładów programowania wykrywania twarzy w aplikacji Scratch i innych opartych na niej.

Zapewne zauważyliście, że niektóre aplikacje na smartfonie rysują prostokąt wokół twarzy w wyniku wykrycia twarzy. Jest to również możliwe w Scratchu.

### Przedstawienie celu głównego lekcji:

Zrozumienie programu wykrywania twarzy i jego użycia na konkretnych przykładach.



<http://erasmus-artie.eu>



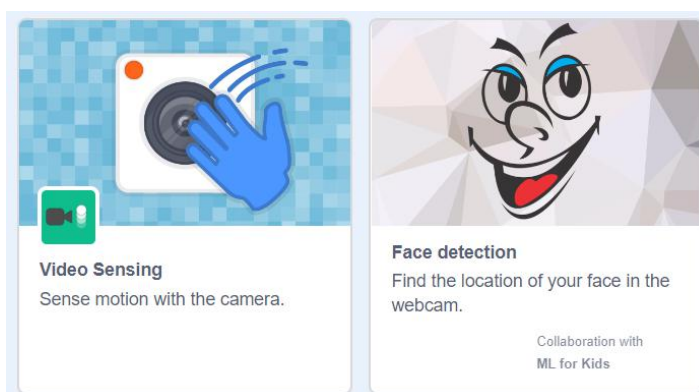
# CZĘŚĆ GŁÓWNA

## Pierwszy projekt - SCRATCH (ML4KIDS):

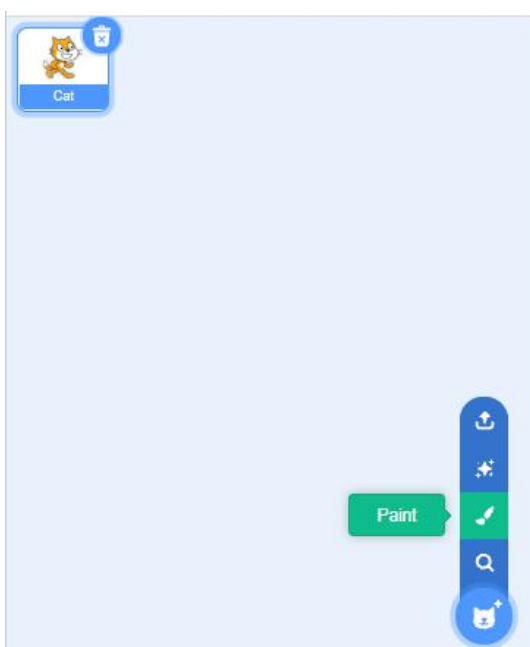
Krok 1: Otwórz przeglądarkę internetową Chrome i przejdź do:

<https://machinelearningforkids.co.uk/scratch3/>

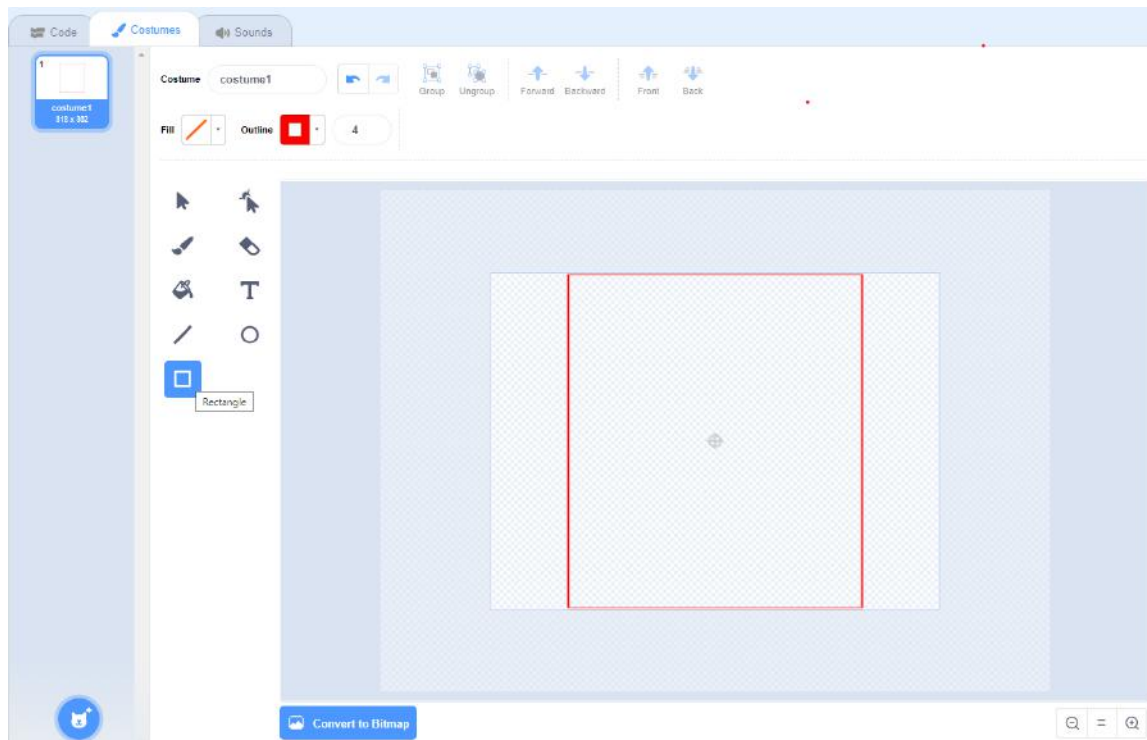
Krok 2: Załaduj rozszerzenia wykrywania wideo i wykrywania twarzy oraz podłącz kamerę internetową (jeśli nie masz wbudowanej)



Krok 3: Usuń duszka Kota, klikając ikonę kosza (prawy górny róg) i wybierz opcję Narysuj nowego duszka

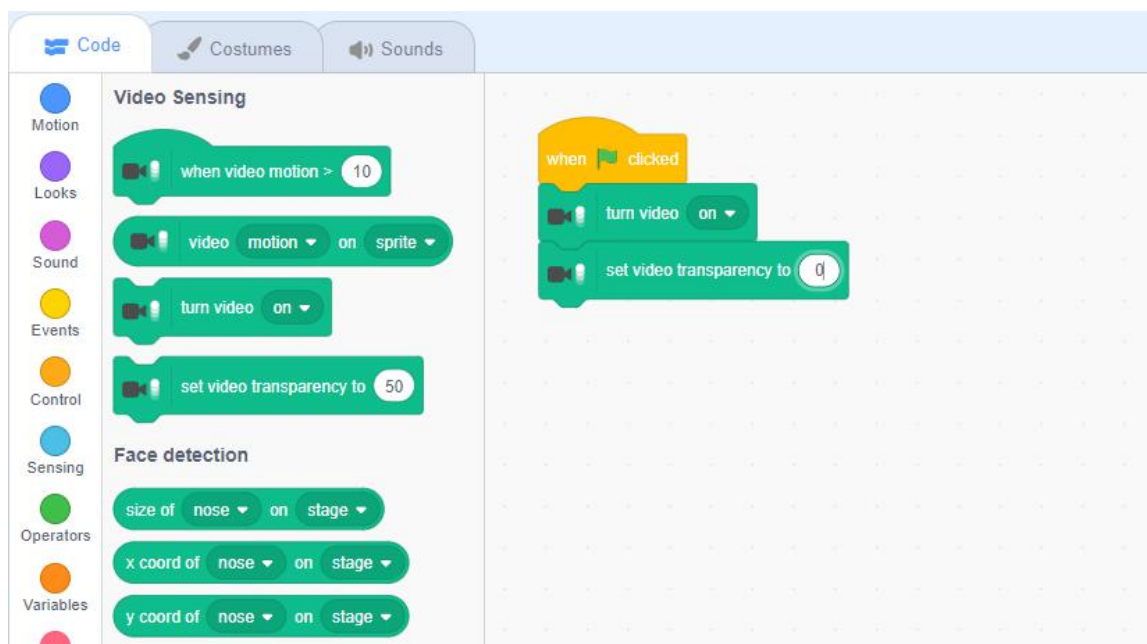


Krok 4: Narysuj prostokąt bez wypełnienia (będzie używany jako ramka ograniczająca) i ustaw kontur na czerwony (4) jak na poniższym obrazku



f

Krok 5: Przejdź do zakładki Kod i rozpocznij programowanie. Najpierw włączamy wideo i ustawiamy przezroczystość na 0 (nieprzezroczysty).

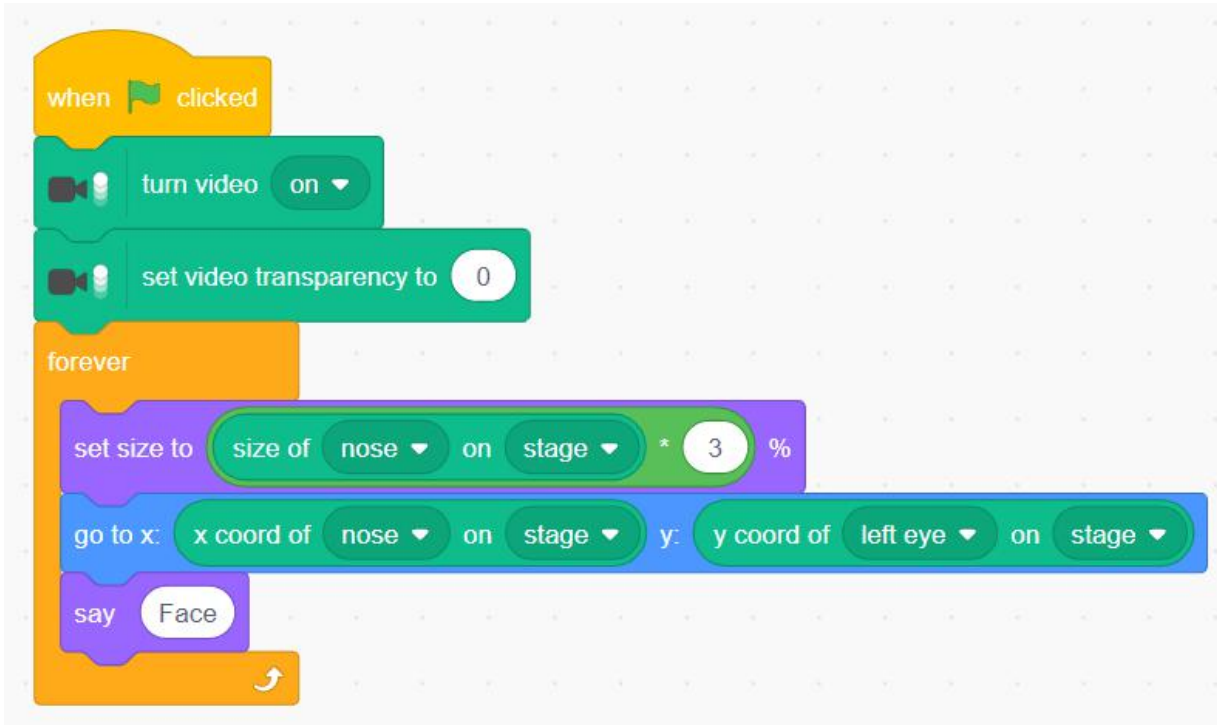


<http://erasmus-artie.eu>

21



Krok 6: Następnie mamy pętlę (Forever) z 3 blokami, aby ustawić rozmiar prostokąta (rozmiar nosa pomnożony przez 3) i jego współrzędne x i y. Możesz chcieć dostosować wartość mnożnika z 4 do innej liczby, zwłaszcza jeśli przełączysz się na większą scenę (lepiej 1,5).



## Drugi projekt - wykrywanie twarzy połączone z rozszerzoną rzeczywistością

Krok 1: Otwórz Scratch na <https://machinelearningforkids.co.uk/scratch3/>

Krok 2: Dodaj rozszerzenie „Wykrywanie twarzy”

Krok 3: Dodaj rozszerzenie „Wykrywanie wideo” i podłącz kamerę internetową (jeśli nie masz wbudowanej)

Krok 4: Usuń duszka Cat

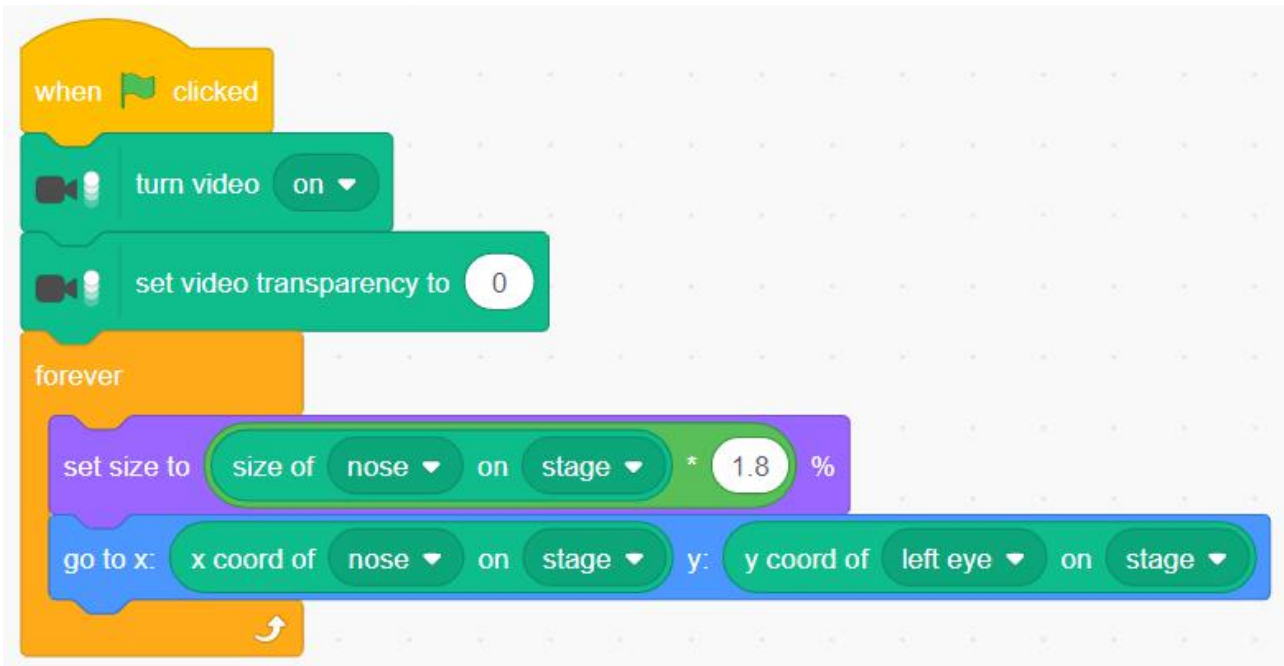
Krok 5: Pobierz zdjęcie z linku [https://toppng.com/transparent-glasses-PNG-free-PNG-Images\\_110945](https://toppng.com/transparent-glasses-PNG-free-PNG-Images_110945)

Krok 6: Prześlij obrazek do Scratcha jako duszka, zmień jego nazwę na „glasses”

Krok 7: Kod duszka:

Krok 8: Uruchom program i porusz głową

Krok 9: Omów dokładność algorytmu i sposoby jego ulepszenia.



### Trzeci projekt z PICTOBLOX (aplikacja komputerowa):

Krok 1a: Ponieważ nie ma dostępnego GUI online, musisz zainstalować PictoBlox z:

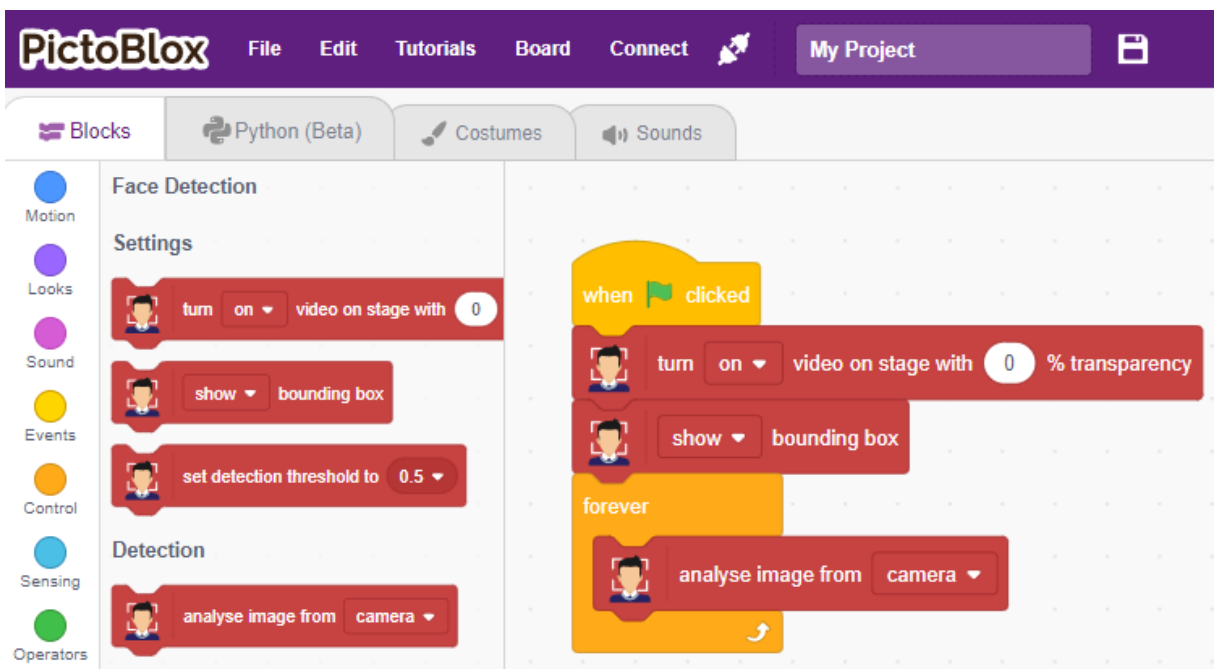
<https://thetempedia.com/product/pictoblox/download-pictoblox/> (427 Mb)

Krok 2a: Otwórz PictoBlox i wybierz rozszerzenie Wykrywanie twarzy

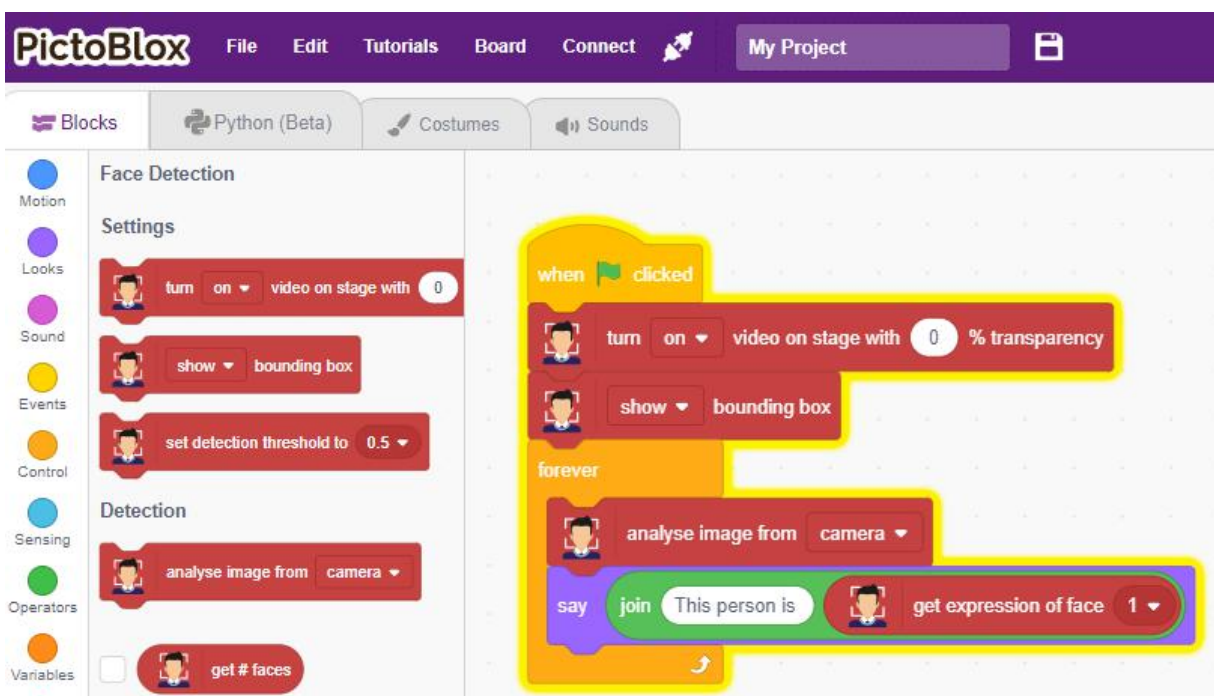
<http://erasmus-artie.eu>



Krok 3a: Użyj bloków, jak na poniższym obrazku, bardzo łatwo zobaczyć, jak to działa. Teraz mamy ramkę ograniczającą jako blok i nie ma potrzeby rysowania prostokąta. Ale główną cechą jest to, że może wykrywać wiele twarzy. Poproś kogoś, aby dołączył do ciebie przed kamerą, aby zobaczyć, jak to działa. Sprawdź blok `get#faces`, aby zobaczyć, ile twarzy zostało wykrytych.



Krok 4a: Uczyńmy to bardziej skomplikowanym, używając operatora join do wyświetlenia wyrazu twarzy osoby. Możesz dokładniej zbadać, jak to działa z wieloma twarzami.







Obecnie oprogramowanie do wykrywania twarzy jest używane w prawie każdej dziedzinie, od urządzeń mobilnych, poprzez filtry twarzy na czacie, po różne aplikacje zabezpieczające. Wykrywanie twarzy pomaga rozpoznawać twarze, ich wiek, wyraz twarzy, płeć, lokalizację i wiele innych funkcji. Wykrywanie twarzy to szerszy termin nadawany każdemu systemowi, który może zidentyfikować obecność ludzkiej twarzy na obrazie wizualnym. Wykrywanie twarzy ma wiele zastosowań, w tym liczenie osób, marketing online, a nawet automatyczne ustawianie ostrości obiektywu aparatu. Jego głównym celem jest oznaczanie obecności twarzy. Rozpoznawanie twarzy zyskało na znaczeniu w ostatnich latach ze względu na jego potencjalne zastosowania. Ponieważ twarze są bardzo dynamiczne i stwarzają więcej problemów i wyzwań do rozwiązania, naukowcy zajmujący się rozpoznawaniem wzorców, widzeniem komputerowym i sztuczną inteligencją zaproponowali wiele rozwiązań zmniejszających takie trudności, aby poprawić niezawodność i dokładność rozpoznawania.

Obecnie wykrywanie twarzy jest stosowane w następujących:

Aplikacje w świecie rzeczywistym (Amazon Recognition: funkcje obejmują weryfikację użytkowników, liczenie osób i moderację treści, często używane przez firmy medialne, firmy zajmujące się analizą rynku, witryny e-commerce i przy rozwiązaniach kredytowych, BioID: zgodne z RODO rozwiązanie stosowane w celu zapobiegania oszustwom internetowym i kradzieży tożsamości, Cognitec: rozpoznaje twarze w strumieniach wideo na żywo, z klientami, od organów ścigania po kontrolę graniczną, FaceFirst: rozwiązanie bezpieczeństwa, które ma na celu wykorzystanie DigitalID do zastąpienia kart i haseł, Trueface.ai: usługi obejmują wykrywanie broni i są wykorzystywane przez wiele sektorów, w tym edukację i bezpieczeństwo...)

Diagnozy medyczne

Łapanie przestępców

Monitoring

## **PODSUMOWANIE**

Wykrywanie twarzy jest wykorzystywane w różnych zaawansowanych systemach i możliwe jest stworzenie prostych przykładów za pomocą narzędzi takich jak Scratch i PictoBlox.





## Scenariusz lekcji 4

**Czas (minuty):**  
90

### Tematyka

Programowanie rozpoznawania twarzy w Scratchu  
Interpretacja wyników algorytmu rozpoznawania twarzy

### Cele

Praktyczne zastosowanie rozpoznawania twarzy

### Oczekiwane efekty

Umiejętność napisania programu do rozpoznawania twarzy za pomocą Scratch  
Badanie możliwości rozszerzenia rozpoznawania twarzy w Scratchu

## Programowanie rozpoznawania twarzy w Scratchu

Celem tej lekcji jest nauczenie się, jak zaprogramować rozpoznawanie twarzy w Scratch.

#### Tematy dyskusji:

Co to jest rozpoznawanie twarzy?  
Jak działa rozpoznawanie twarzy?  
Jakie praktyczne zastosowania może mieć?

Porozmawiaj z uczniami o tym, co wiedzą o rozpoznawaniu twarzy.

Czy wiedzą coś o tworzeniu programu do rozpoznawania twarzy, zanim go wprowadzisz?

Nauczyciel zapoznaje uczniów z programowaniem rozpoznawania twarzy w Scratch i instruuje ich, jak przesyłać obrazy, a następnie wykorzystywać te obrazy w programach rozpoznawania.

#### Przedstawienie celu głównego lekcji:

Dzięki przykładom opartym na jednym programie lepiej zrozumiecie programy do rozpoznawania twarzy i ich zastosowanie.



# CZĘŚĆ GŁÓWNA

Krok 1: Otwórz przeglądarkę internetową i pobierz wszystkie obrazy (20) z:  
<https://bit.ly/daenerys-data>

Posłuż to do trenowania klasy 1

Krok 2: Otwórz przeglądarkę internetową i pobierz wszystkie obrazy (20) z:  
<https://bit.ly/arya-data>

Posłuż to do szkolenia klasy 2

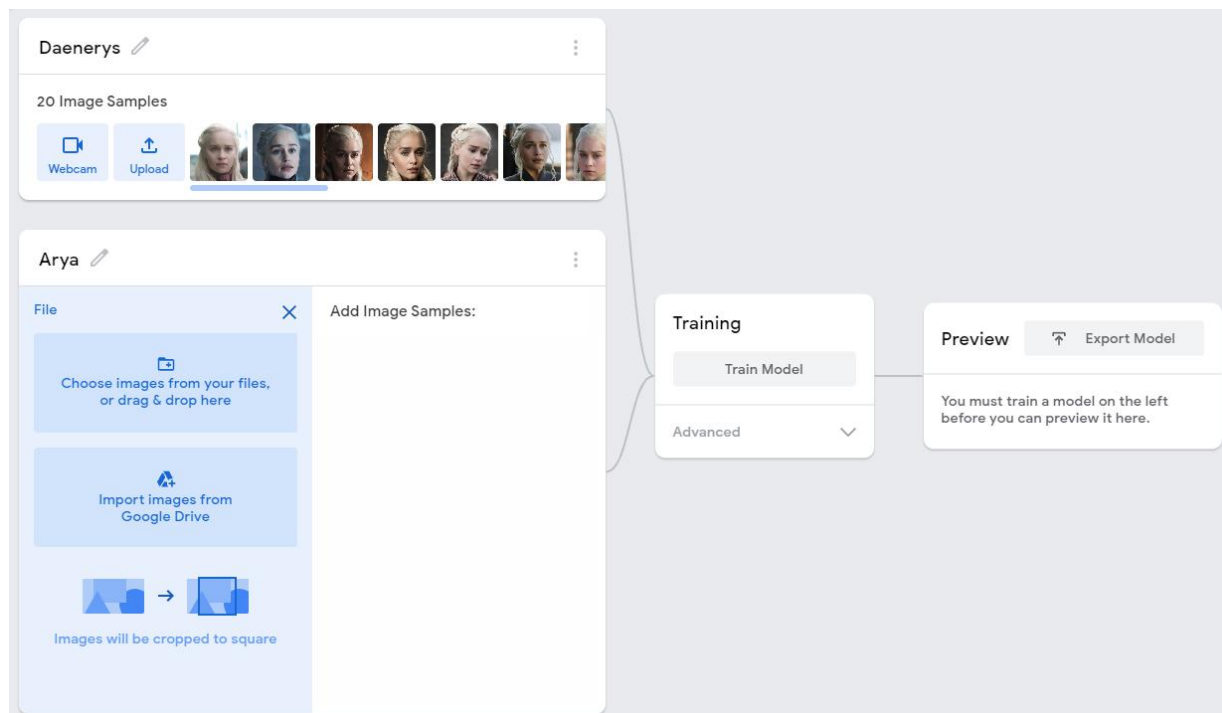
Krok 3: Otwórz przeglądarkę internetową i przejdź do:  
<https://teachablemachine.withgoogle.com/>

Krok 4: Kliknij Rozpocznij.

Krok 5: Wybierz projekt graficzny

Krok 6: Wybierz standardowy model graficzny

Krok 7: Zmień nazwę klasy 1 na Daenerys, a klasy 2 na Arya. Prześlij obrazy Daenerys do plików Daenerys i obrazy Arya do plików Arya, jak pokazano na poniższym obrazku





Krok 8: Trenuj swój model. Nie przełączaj kart przeglądarki podczas procesu uczenia.

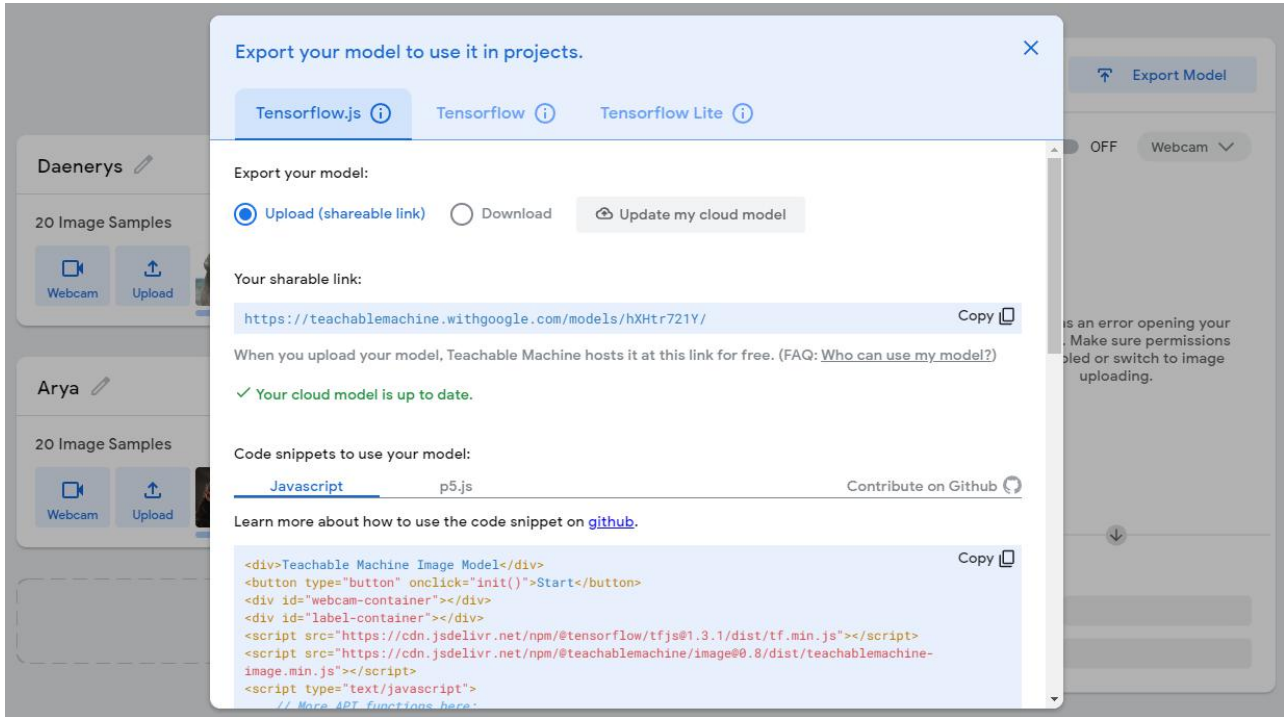


Krok 9: Eksportuj swój model. W wyskakującym okienku wybierz przesłanie go do chmury (trzecia opcja), a Google będzie bezpłatnie przechowywać Twoje dane.

<http://erasmus-artie.eu>



Krok 10: Skopiuj link podany w polu tekstowym poniżej - jest to adres URL Twojego modelu.  
W tym przypadku to <https://teachablemachine.withgoogle.com/models/hXHtr721Y/>



<http://erasmus-artie.eu>

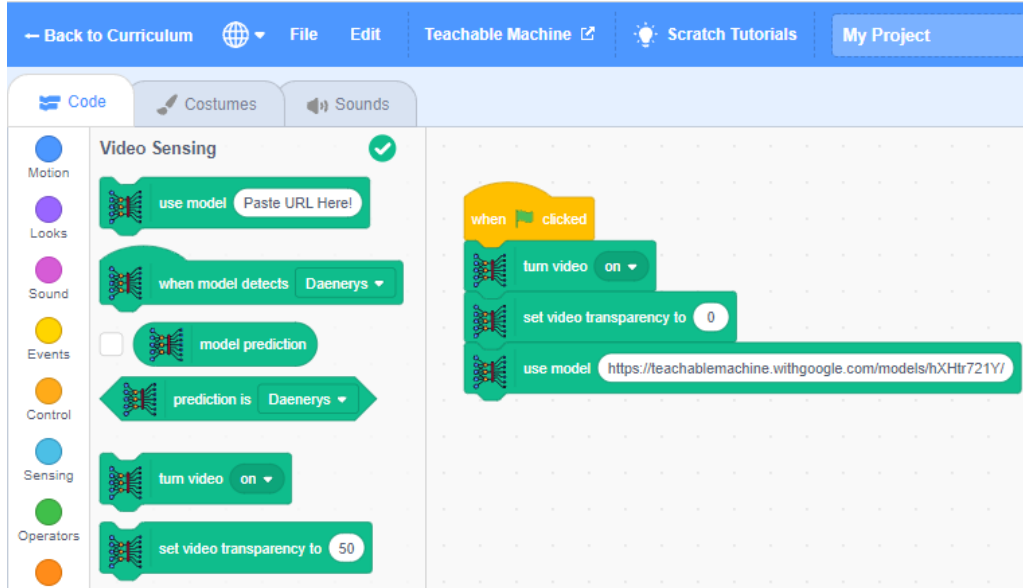
Krok 11: Twój model jest gotowy do użycia

Krok 12: Otwórz GUI Scratch na: <https://mitmedialab.github.io/prg-extension-boilerplate/create/i> załaduj rozszerzenie Teachable Machine.



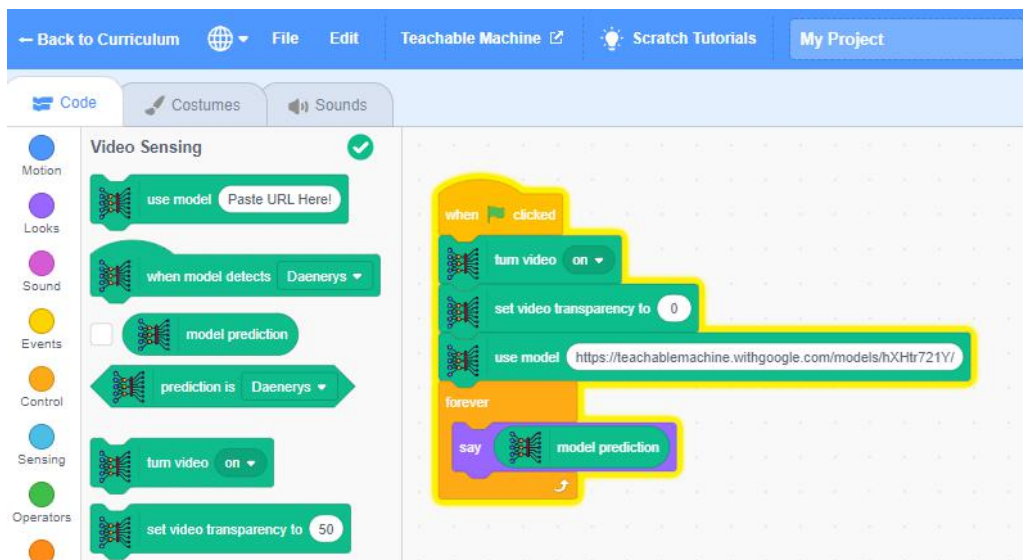


Krok 13: Najpierw musimy włączyć wideo w kamerze i ustawić przezroczystość na 0 (nieprzezroczysty). Następnie używamy bloku adresu URL modelu i wklejamy tam nasz link do modelu.



Krok 14: Ostatnie kilka bloków to pętla nieskończona (forever) i blok zawierający blok raportowania z przewidywanym wynikiem. Prawdopodobnie nie będziecie w stanie uchwycić prawdziwych Daenerys lub Aryi za pomocą kamerki, więc użyjcie smartfona z ich zdjęciami i skierujcie go na kamerkę, aby zobaczyć wyniki. Możecie trenować model własnymi zdjęciami lub zdjęciami znajomych.

\*\*\* Upewnijcie się, że nie robicie nikomu zdjęcia bez jego zgody.





## PICTOBLOX (aplikacja komputerowa):

Krok 1a: Ponieważ nie ma dostępnego GUI online, musicie zainstalować PictoBlox z:  
<https://thestempedia.com/product/pictoblox/download-pictoblox/> (427 Mb)

Krok 2a: Otwórzcie PictoBlox i wybierzcie rozszerzenie Wykrywanie twarzy



Krok 3a: Otwórzcie przeglądarkę internetową i pobierzcie wszystkie 20 obrazów z:

<https://bit.ly/daenerys-data>

Będą służyły do trenowania klasy 1



Krok 4a: Otwórzcie przeglądarkę internetową i pobierzcie wszystkie 20 obrazów z:

<https://bit.ly/arya-data>

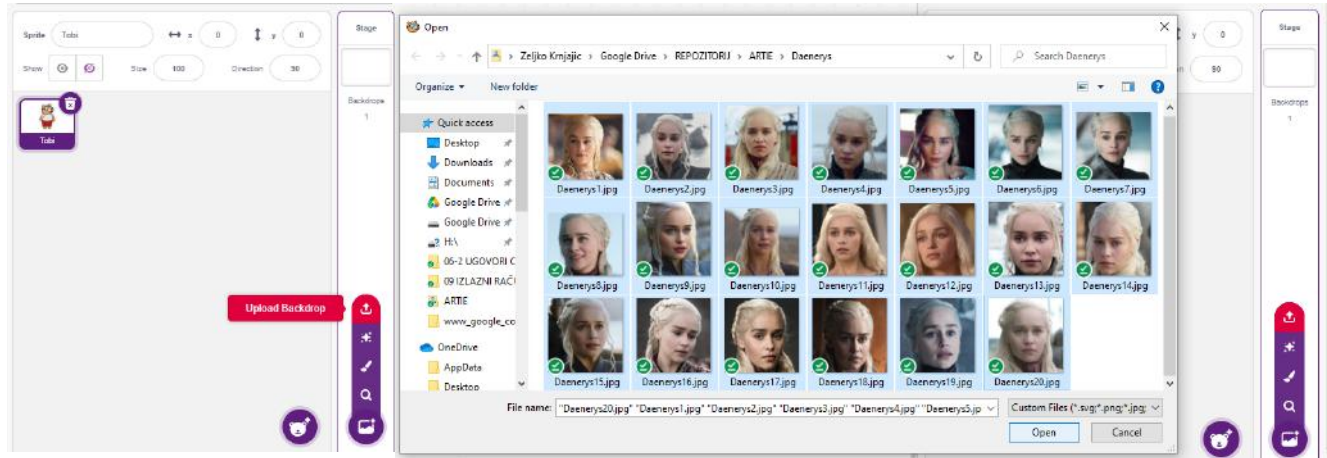
Będą używane do trenowania klasy 2

Krok 5a: Ukryjcie duszka Tobiego, klikając ikonę, jak pokazano na poniższym obrazku.

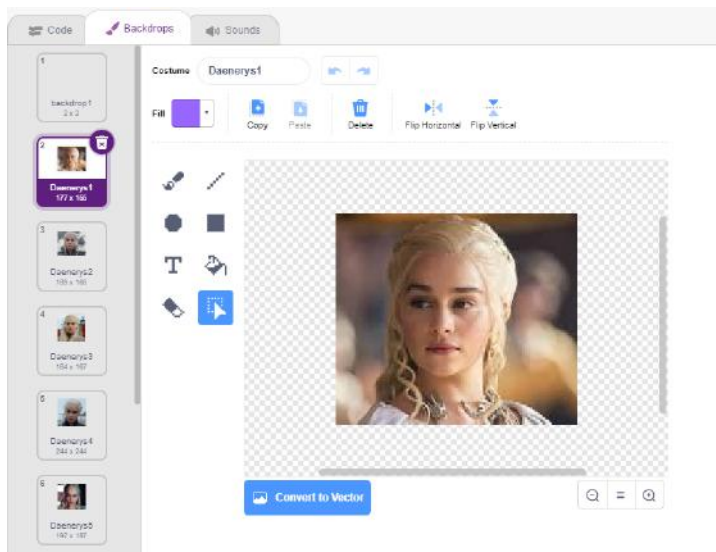




Krok 6a: Prześlijcie wszystkie obrazy Daenerys do tła (Prześlij tło – Wybierz wszystkie obrazy – Otwórz)

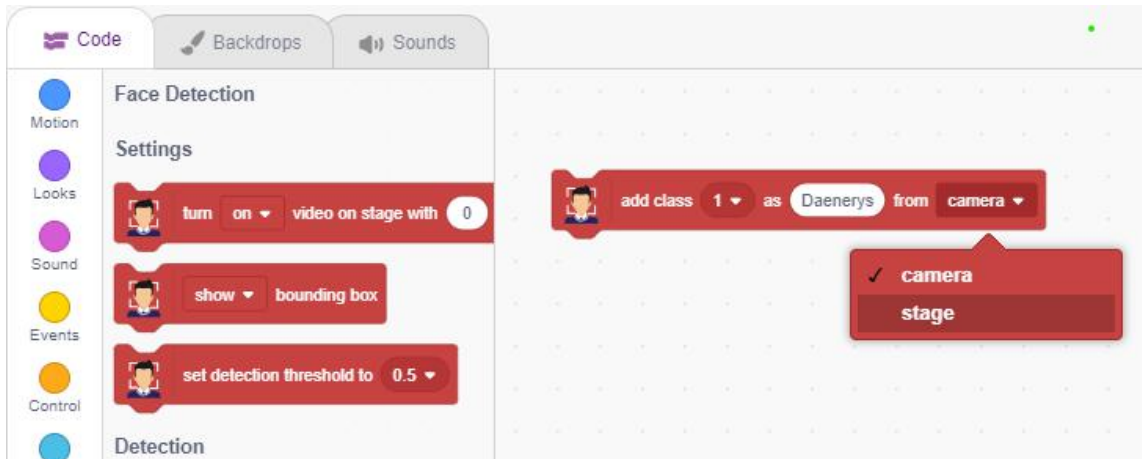


Krok 7a: Wybierz obraz Daenerys1 i przejdź do zakładki Kod

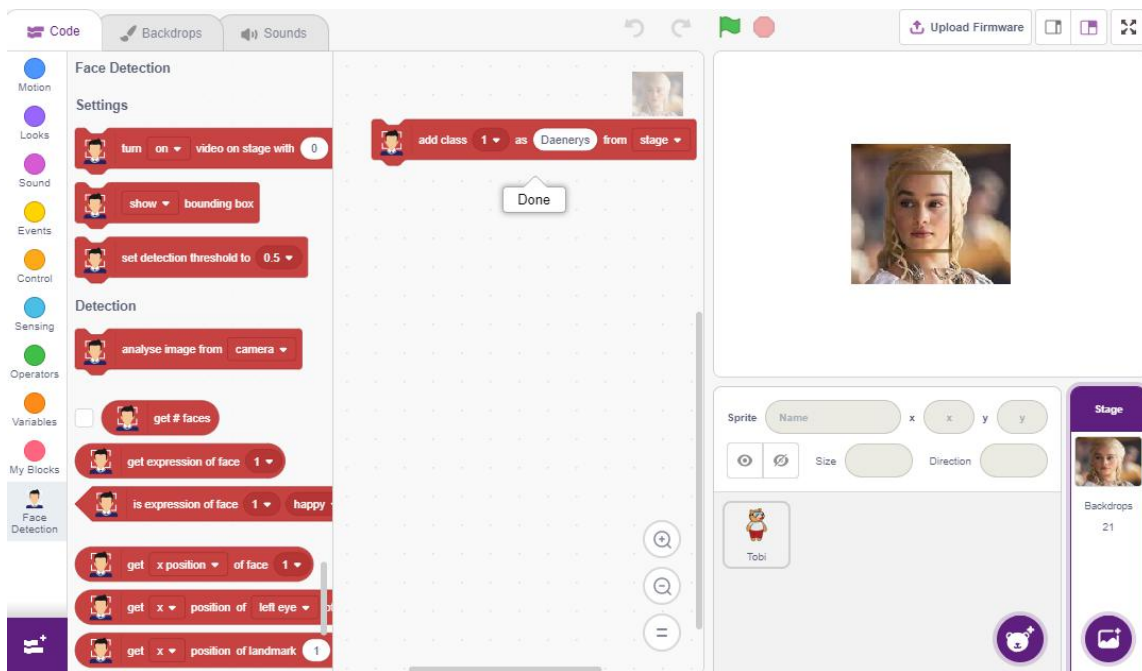


Krok 8a: W grupie Wykrywanie twarzy wyszukaj blok dodawania klasy, przeciągnij go i upuść w obszarze kodu, a następnie zmień nazwę klasy z 'Jarvis' na 'Daenerys' oraz zmień źródło z kamery na scenę, zgodnie z ilustracją poniżej.

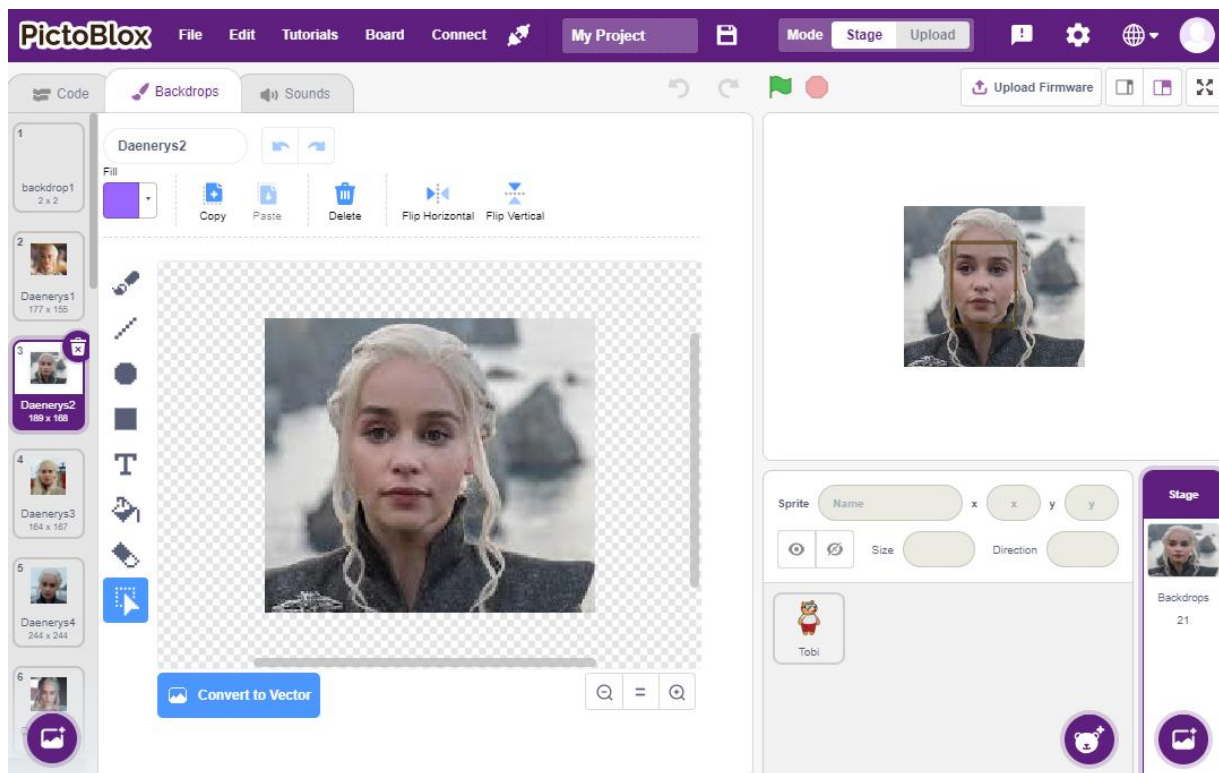




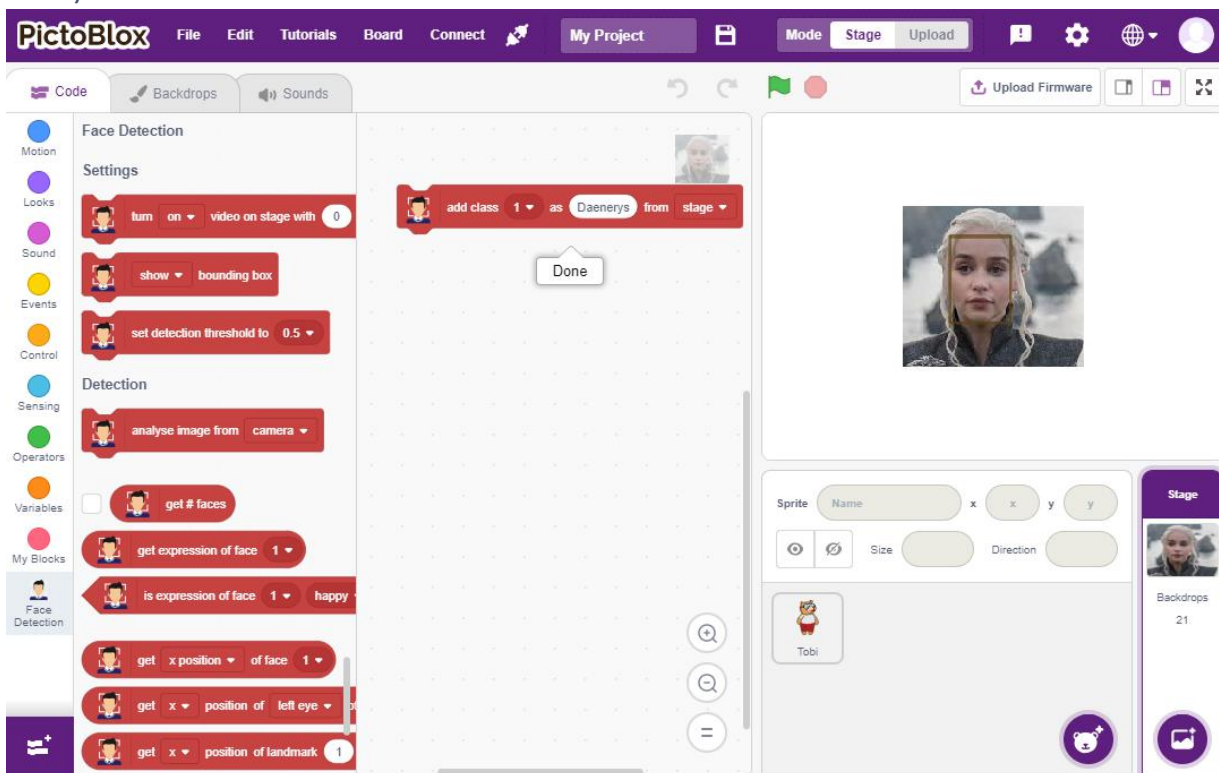
Krok 9a: Po dokonaniu zmian, wystarczy kliknąć na blok dodawania klasy, aby rozpocząć proces uczenia. Otrzymacie komunikat "Zakończono" i zobaczycie ramkę wokół twarzy Daenerys. Uczenie z pierwszym obrazem jest zakończone, a te kroki należy powtórzyć dla każdego kolejnego obrazu.



Krok 10a: Przełącz się z powrotem do zakładki Tła i wybierz obraz Daenerys 2.



Krok 11a: Przełącz się z powrotem do zakładki Kodu i ponownie kliknij na blok dodawania klasy.

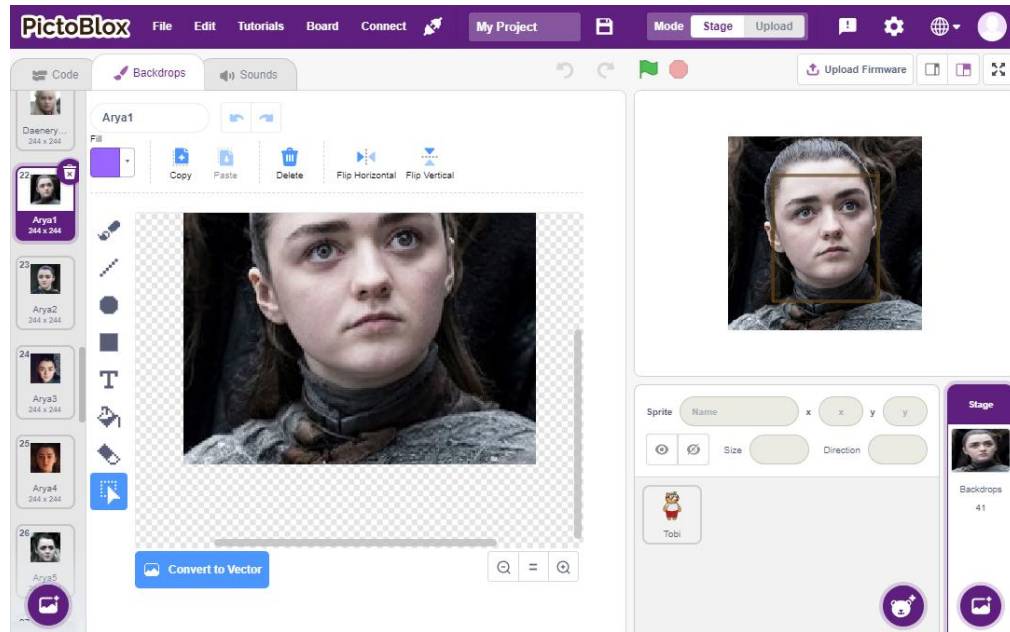




Krok 12a: Powtórz kroki 10a i 11a dla każdego pojedynczego obrazu (aż do Daenerys20).

Krok 13a: Teraz prześlij wszystkie obrazy Aryi do tła w ten sam sposób, jak w kroku 6a.

Krok 14a: Wybierz obraz Arya1.

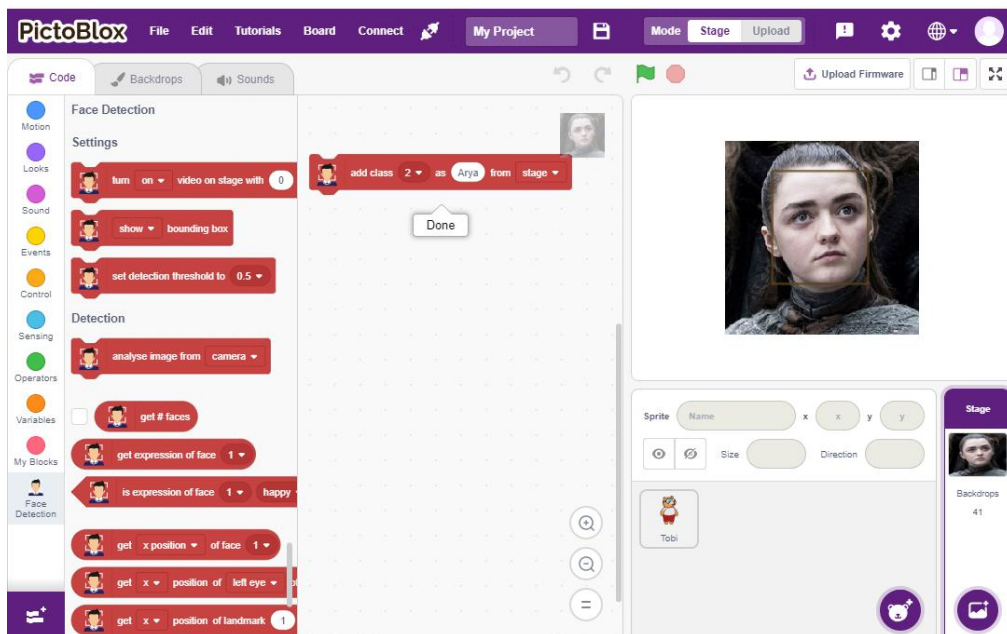


Krok 15a: Przełącz się z powrotem do zakładki Kodu i zmień (to jest ważne, ponieważ trenujemy drugą klasę) nazwę dodawanej klasy z Daenerys na Arya. Źródło pozostaje takie samo - ze sceny. Następnie kliknij ten blok, aby rozpocząć trening pierwszego obrazu z klasy Arya.

f

<http://erasmus-artie.eu>

35





Krok 16a: Przełącz się do zakładki Tła i wybierz obraz Arya2.

Krok 17a: Przełącz się do zakładki Kodu i kliknij na blok dodawania klasy.

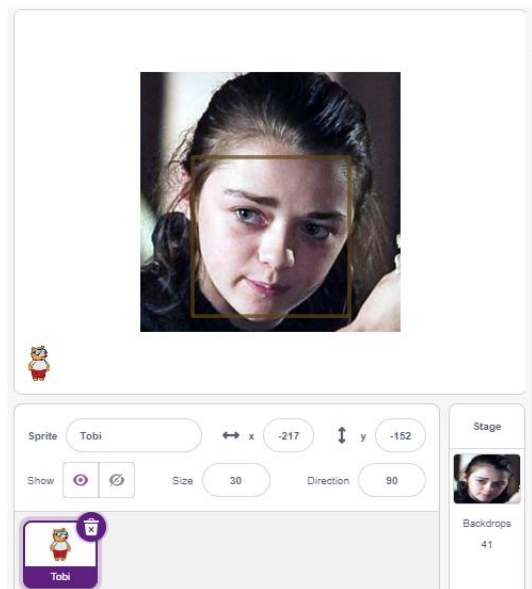
Krok 18a: Powtórz kroki 16a i 17a dla każdego pojedynczego obrazu (aż do Arya20).

Krok 19a: Twój model jest teraz gotowy do testowania. Wybierz sprite Tobi i ustaw go jako widoczny (show). Ustaw rozmiar Tobiego na 30% i przemieszczaj go z centrum w róg ekranu

Krok 20a: Podłącz kamerę (jeśli jej nie masz) i rozpocznij kodowanie. Włącz transmisję wideo z przezroczystością 0% i pokaż ramkę wokół twarzy. Następny blok to pętla nieskończona, w której Tobi będzie wyświetlał wynik rozpoznawania twarzy. Jest to blok z podwójnym if-else, a ostatnia opcja 'else' zwróci pusty ciąg znaków, jeśli nie zostanie rozpoznana żadna twarz.

Krok 21a: Uruchom program. Użyj swojego smartfona z obrazami Daenerys lub Aryi i skieruj go w stronę kamery, aby zobaczyć wyniki. Możesz trenować model za pomocą własnych zdjęć lub zdjęć swoich znajomych.

\*\*\* Upewnij się, że nie robisz zdjęć nikomu bez ich zgody.





## PROJEKT Z ROZPOZNAWANIEM TWARZY

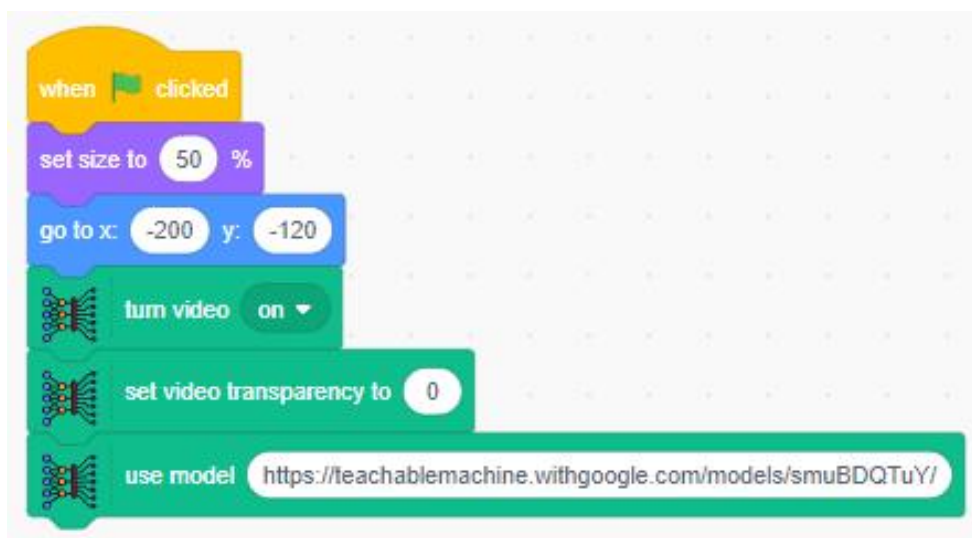
Oto mały projekt, który jest praktycznym zastosowaniem rozpoznawania twarzy. Przygotowaliśmy dla Ciebie wytrenowany model ośmiu znanych osób, który możesz porównać ze swoją twarzą w Teachable Machine. Mamy Adrianę Limę, Emilię Clarke, Gal Gadot, Natalie Portman, Selenę Gomez, Emmę Stone, Zoe Saldanę i Maisie Williams. Musisz dowiedzieć się, która twarz z tej listy jest najbardziej podobna do Twojej.

Krok 1b: Otwórz GUI Scratcha pod adresem:

<https://mitmedialab.github.io/prg-extension-boilerplate/create/> i załaduj rozszerzenie Teachable Machine.



Krok 2b: Najpierw ustawiamy rozmiar i pozycję duszka, a następnie włączamy transmisję video z kamery i ustawiamy przezroczystość na 0 (brak przezroczystości). Następnie korzystamy z bloku URL modelu i wklejamy ten link do modelu: <https://teachablemachine.withgoogle.com/models/smuBDQTuY/>





Krok 3b: Następnie skorzystamy z bloku typu zdarzenia, które będzie aktywowane przy dopasowaniu twarzy. Dodajmy do tego bloku blok mówiący 'Wyglądasz jak...!'

```
when clicked
  set size to 50 %
  go to x: -200 y: -120
  turn video on
  set video transparency to 0
  use model https://teachablemachine.withgoogle.com/models/smuBDQTuY/

when model detects Adriana
  say You look like Adriana Lima
```

Krok 4b: Dodaj tę kombinację bloków dla każdej znanej osoby z listy.

```
when clicked
  set size to 50 %
  go to x: -200 y: -120
  turn video on
  set video transparency to 0
  use model https://teachablemachine.withgoogle.com/models/smuBDQTuY/

when model detects Adriana
  say You look like Adriana Lima

when model detects Emilia
  say You look like Emilia Clarke

when model detects Gal
  say You look like Gal Gadot

when model detects Natalie
  say You look like Natalie Portman

when model detects Selena
  say You look like Selena Gomez

when model detects Emma
  say You look like Emma Stone

when model detects Zoe
  say You look like Zoe Saldana

when model detects Maisie
  say You look like Maisie Williams
```





Krok 5b: Podłącz/włącz kamerę internetową, uruchom kod i zobacz, kto jest Twoim odpowiednikiem!

Krok 6b: Chcesz zmienić obecny model na własny? Na stronie

Być może zostaniesz poproszony/poproszona o rejestrację na Kaggle przed pobraniem danych. Możesz również ręcznie zgromadzić obrazy za pomocą wyszukiwarki Google. Po zebraniu wszystkich potrzebnych obrazów użyj narzędzia Teachable Machine na stronie <https://teachablemachine.withgoogle.com/>, aby wytrenować swój własny model, podobnie jak w scenariuszu lekcji Programowanie rozpoznawania twarzy w Scratchu.

Rozpoznawanie twarzy to technologia zdolna do identyfikowania lub weryfikacji osoby na podstawie obrazu, wideo lub innego materiału audiowizualnego jej twarzy. Generalnie, identyfikacja ta jest wykorzystywana do dostępu do aplikacji, systemu lub usługi. Przed rozpoczęciem programowania rozpoznawania twarzy musimy zebrać zdjęcia konkretnej twarzy i przetrenować model. W tym projekcie skorzystaliśmy z aplikacji o nazwie Teachable Machine. Teachable Machine to narzędzie internetowe, które umożliwia szybkie, łatwe i dostępne dla wszystkich tworzenie modeli uczenia maszynowego.

Czy zauważyłeś/aś, że ten projekt jest całkowicie oparty na technologii internetowej i nie wymaga instalacji żadnego oprogramowania?

Dzisiaj wiele osób na całym świecie regularnie korzysta z technologii rozpoznawania twarzy.

### **Główne funkcje rozpoznawania twarzy:**

#### **Weryfikacja tożsamości:**

Identyfikacja jednostek i zastosowanie określonych reguł na podstawie przynależności do określonej kategorii, takiej jak VIP, zarejestrowany gość, osoba na czarnej liście, pracownik lub student. Wykorzystanie tych informacji do usprawnienia i automatyzacji procesów, takich jak: (1) kontrola dostępu, (2) ochrona bezpieczeństwa, (3) powitania klientów lub gości, oraz (4) rejestrowanie czasu pracy pracowników.

#### **eKYC and Spoofing Prevention:**

Weryfikacja tożsamości osoby poprzez użycie żywego zdjęcia lub nagrania wideo w połączeniu ze zeskanowanym (i zweryfikowanym) dowodem tożsamości. Proces ten nazywany jest eKYC (elektroniczne potwierdzenie tożsamości klienta) i jest powszechnie stosowany w sektorze finansowym, w tym w bankowości, usługach finansowych, ubezpieczeniach i podobnych przypadkach.





#### Autoryzacja:

Weryfikacja, czy dana osoba znajduje się w uprzednio autoryzowanej bazie danych w celu (1) wypłacenia gotówki z bankomatu, (2) dostępu do szafki medycznej zawierającej zabezpieczone leki lub (3) odblokowania kosztownych urządzeń wymagających wyszkolonych operatorów.

#### Segmentacja klientów i analiza:

W przypadku inteligentnej reklamy, analizuje się cechy osoby stojącej przed cyfrowym znakiem, takie jak płeć, wiek i emocje.

#### Środki zdrowotne:

Potwierdzanie, czy osoba prawidłowo nosi maskę oraz sprawdzanie, czy nie ma gorączki przed udzieleniem dostępu do budynku lub restauracji.

Przy analizie rynków wertykalnych, wyróżnia się 10 branż jako idealne do integracji rozpoznawania twarzy, które w wielu przypadkach już je wprowadzają:

1. Przemysł produkcyjny i magazynowy
2. Bankowość, usługi finansowe i ubezpieczenia (BFSI)
3. Inteligentne biura
4. Inteligentne domy i kompleksy mieszkaniowe
5. Handel detaliczny
6. Transport publiczny i lotniska
7. Placówki opieki zdrowotnej
8. Szkoły i uniwersytety
9. Branża hotelarska
10. Restauracje i bary

## **PODSUMOWANIE**

Rozpoznawanie twarzy jest wykorzystywane w różnych zaawansowanych systemach, a proste przykłady można stworzyć przy użyciu narzędzi takich jak Scratch i PictoBlox.





## Scenariusz lekcji 5

Czas (minuty):  
90

### Tematyka

Wykrywanie i klasyfikacja obiektów dla początkujących w Scratchu

### Cele

Nauka czym jest wykrywanie i klasyfikacja obiektów w Scratchu dla początkujących

### Oczekiwane efekty

Zrozumienie, czym jest wykrywanie i klasyfikacja obiektów za pomocą prostego przykładu w Scratchu  
Zrozumienie różnicy między wykrywaniem obiektów, a klasyfikacją obiektów

## Wykrywanie i klasyfikacja obiektów dla początkujących w Scratchu

Wykrywanie i klasyfikacja obiektów przyciągnęła wiele uwagi w ostatnich dziesięcioleciach. W dziedzinie widzenia komputerowego jedno z najczęstszych pytań dotyczy różnicy między klasyfikacją obrazu, wykrywaniem obiektów i segmentacją obrazu.

#### Przedstawienie celu głównego lekcji:

Wprowadzenie do wykrywania i klasyfikacji obiektów dla początkujących poprzez przykłady różnych zastosowań.

Zacznijmy od zrozumienia, czym jest klasyfikacja obrazu: Rozważmy poniższy obraz:





# CZĘŚĆ GŁÓWNA

Rozpoznajemy natychmiast. To jest pies. Cofnijmy się o krok i przeanalizujmy, jak doszliśmy do tego wniosku. Pokazano nam obraz i sklasyfikowaliśmy klasę, do której należał (w tym przypadku psa). I o to, w skrócie, chodzi w klasyfikacji obrazów.

Jak widzieliśmy, jest tu tylko jeden obiekt: pies. Możemy łatwo użyć modelu klasyfikacji obrazu i przewidzieć, że na danym obrazie znajduje się pies. Ale co, jeśli mamy zarówno kota, jak i psa na jednym obrazie?



W takim przypadku możemy trenować wieloetykietowy klasyfikator. Jednak nie będziemy znać lokalizacji żadnego zwierzęcia/obiektu na obrazie. Tutaj wkracza lokalizacja obrazu. Pomaga nam określić położenie pojedynczego obiektu na danym obrazie. W przypadku obecności wielu obiektów polegamy na koncepcji detekcji obiektów. Za pomocą detekcji obiektów możemy przewidzieć lokalizację wraz z klasą dla każdego obiektu.



## Classification



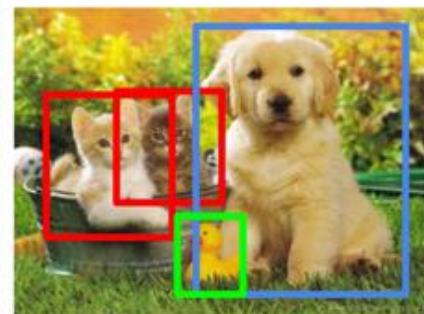
CAT

## Classification + Localization



CAT

## Object Detection



CAT, DOG, DUCK

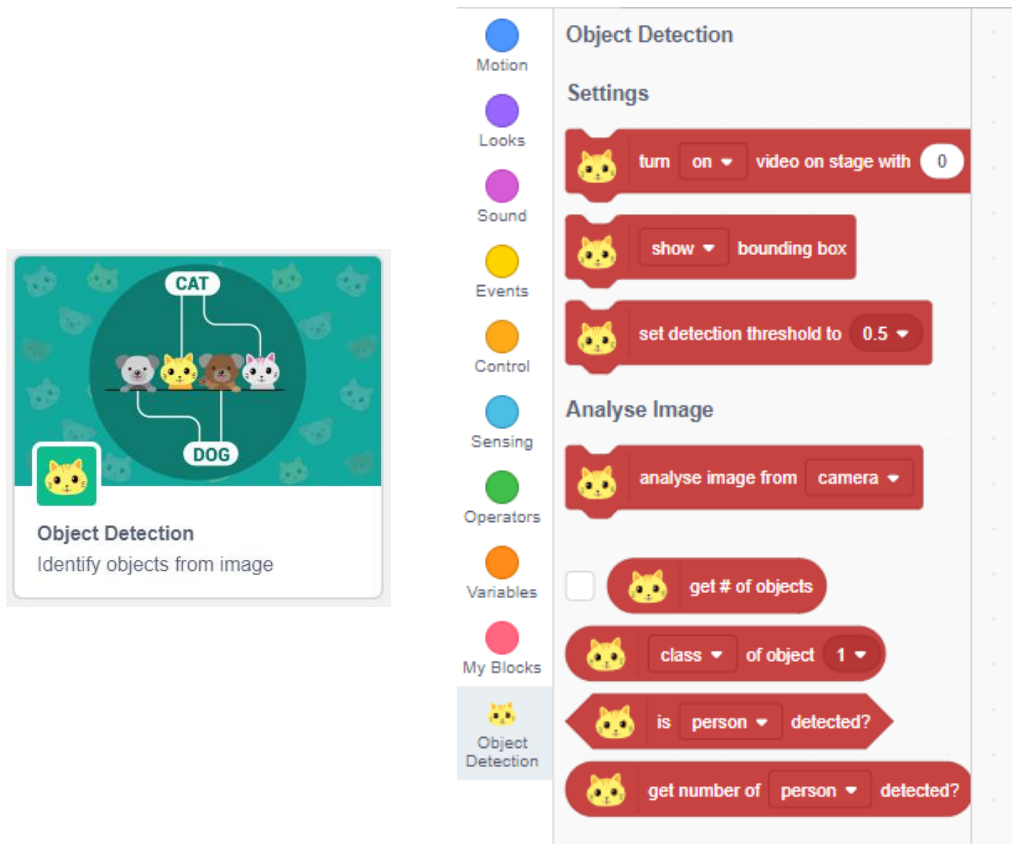


### Aplikacja z wykrywaniem obiektów

PictoBlox jest obecnie jedyną aplikacją posiadającą zdolności detekcji obiektów. Jest to aplikacja typu desktop, którą należy najpierw zainstalować ze strony <https://thetempedia.com/product/pictoblox/download-pictoblox/> (427 Mb).

Krok 1: Po zainstalowaniu załaduj rozszerzenie do detekcji obiektów.

Krok 2: Oto dostępne bloki do programowania.

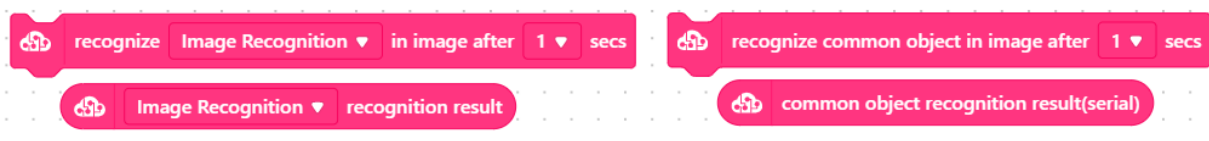


Pierwsze 3 bloki dotyczą ustawień strumienia kamery, a pozostałe są wykorzystywane do analizowania i raportowania.

Sytuacja z klasyfikacją obiektów jest znacznie lepsza, ponieważ mamy kilka innych aplikacji do wyboru (oprócz PictoBlox).

### Makeblock (mBlock) - <https://ide.mblock.cc/>

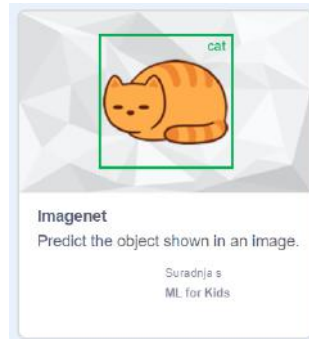
Załaduj rozszerzenie Cognitive Services, a znajdziesz 4 bloki do użycia w celu rozpoznawania obiektów (klasyfikacji).





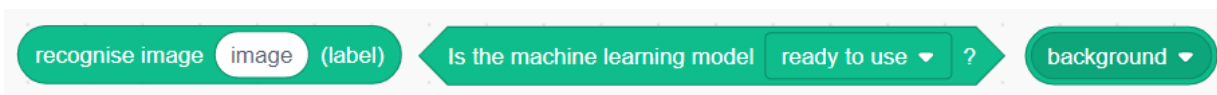
Makeblock udostępnia również rozszerzenie Teachable Machine (nie mylić z Google), w którym możesz trenować do 3 klas i używać go do klasyfikacji obiektów.

**Scratch (ML4KIDS)** - <https://machinelearningforkids.co.uk/scratch3/>  
Załaduj rozszerzenie ImageNet.



Dostępne są 3 bloki. Użyj ich w połączeniu z rozszerzeniem Video Sensing, aby włączyć/wyłączyć strumień wideo z kamery oraz ustawić przezroczystość. Rozszerzenie to zostało przeszkolone do rozpoznawania zdjęć tysiąca powszechnych obiektów. Model uczenia maszynowego oparty jest na MobileNet (modelu ML zaprojektowanym dla urządzeń mobilnych, co oznacza, że nie wymaga dużych mocy obliczeniowych). Pełna lista obiektów jest dostępna tutaj:

<https://storage.googleapis.com/download.tensorflow.org/data/ImageNetLabels.txt>

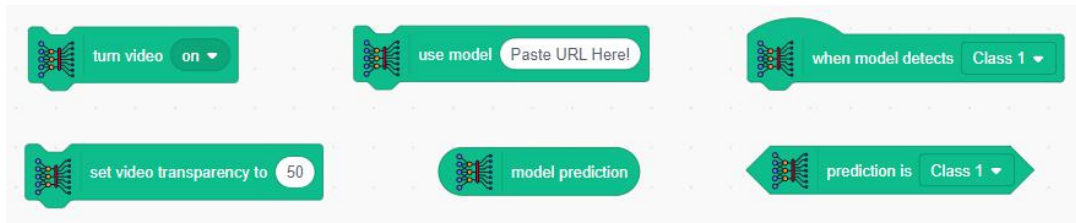


**Scratch (MITMEDIALAB)** - <https://mitmedialab.github.io/prg-extension-boilerplate/create/>  
Załaduj rozszerzenie Teachable Machine.



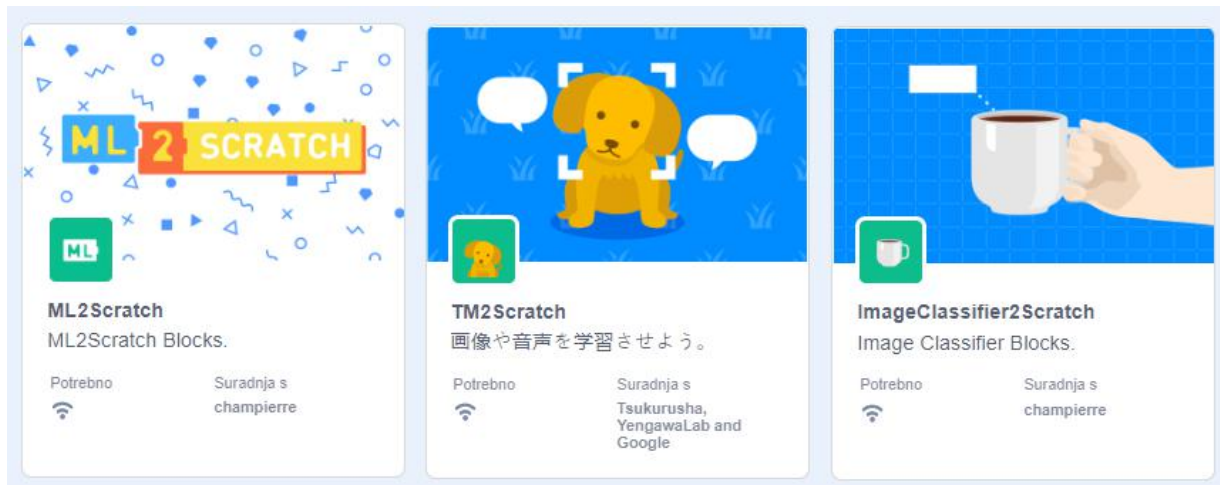


Użyj tych bloków w połączeniu z Google Teachable Machine.

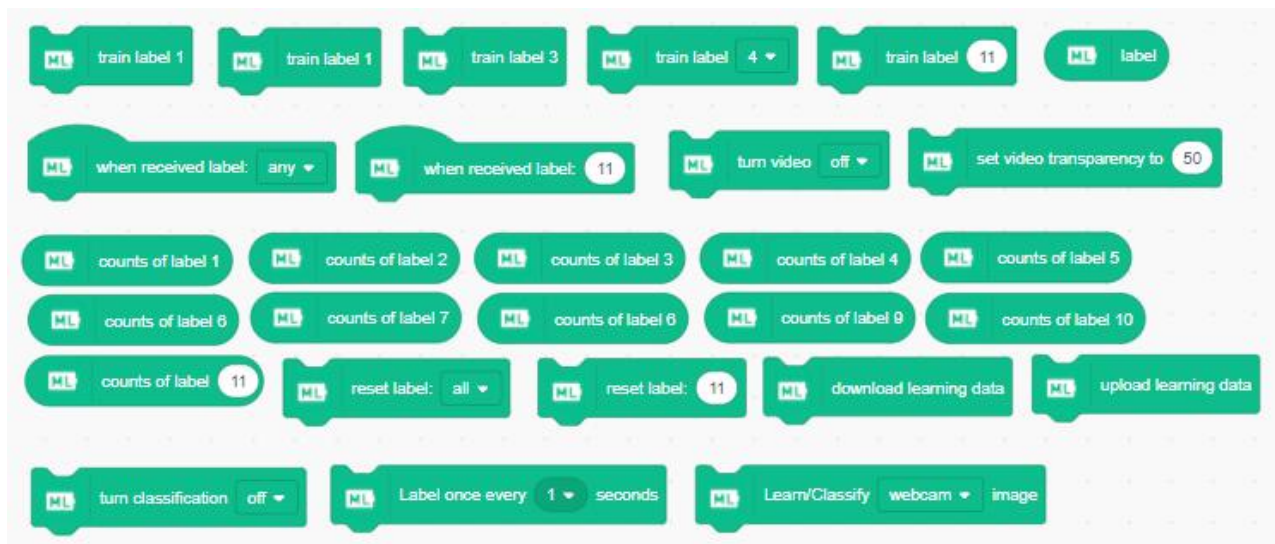


Stretch3 (github.io) - <https://stretch3.github.io/>

Załaduj rozszerzenia ML2Scratch, TM2Scratch i ImageClassifier2Scratch, aby używać wielu bloków do klasyfikacji obiektów i szkolenia.

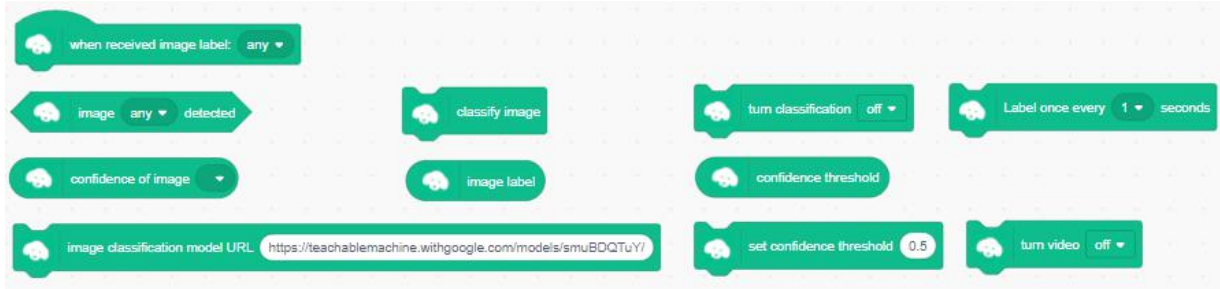


### ML2Scratch blocks





### TM2Scratch blocks



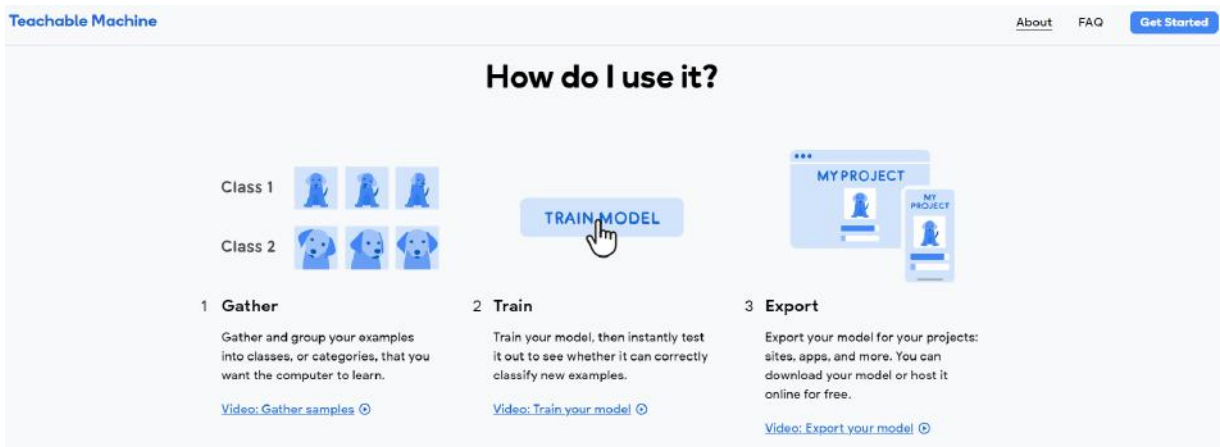
### ImageClassifier2Scratch blocks



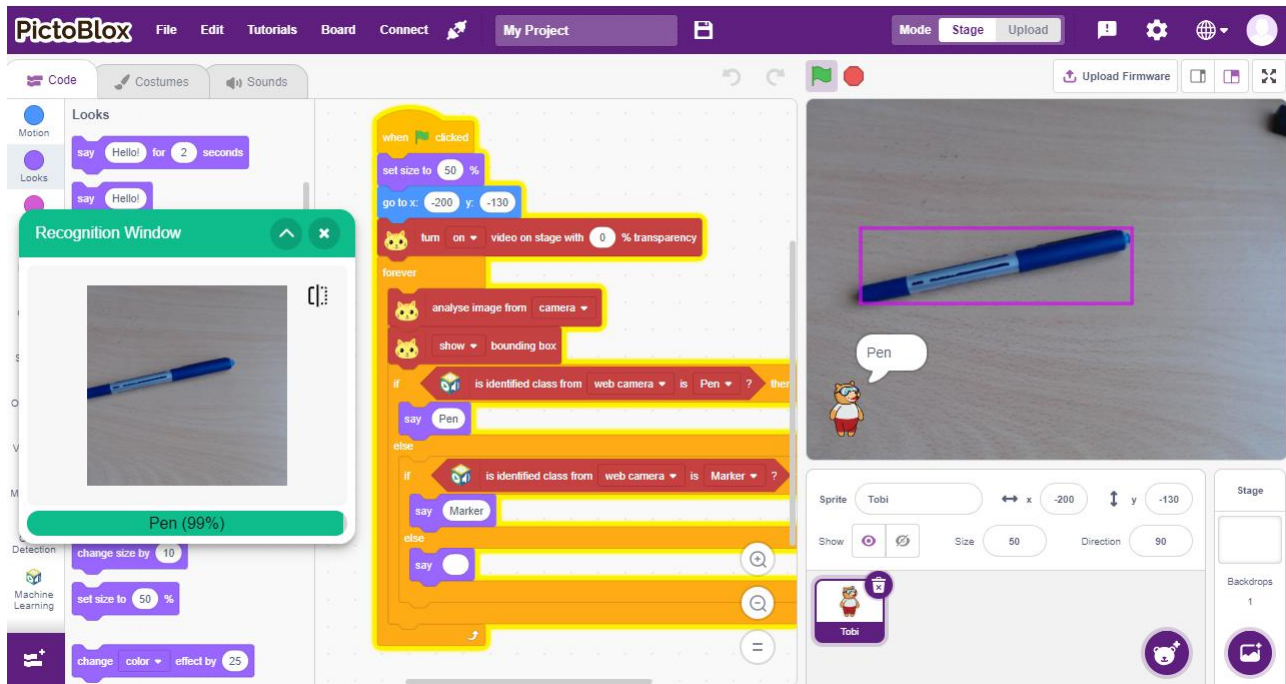
Teachable machine (Google) - <https://teachablemachine.withgoogle.com/>  
Ta aplikacja służy do szkolenia modelu i wykorzystywania go do rozpoznawania obiektów w połączeniu z rozszerzeniami Teachable Machine dostępnymi w PictoBlox, Scratch (MITMEDIALAB) i Stretch3.



<http://erasmus-artie.eu>



Przykład użycia detekcji obiektów - długopis lub marker.  
Teachable Machine jest używane do szkolenia 2 klas. Link do wytrenowanego modelu to: <https://teachablemachine.withgoogle.com/models/FdWn0CA2a/>  
Rozszerzenia używane w PictoBlox to Object detection i Machine learning



Detekcja i klasyfikacja obiektów są dwiema kluczowymi zadaniami w rozumieniu obrazu. Rozpoznawanie obiektów na obrazie wymaga połączenia wielu różnych sygnałów z danych obrazowych. Często wykorzystuje się dwa rodzaje informacji: lokalny wygląd, który opisuje sam obiekt, oraz globalną reprezentację, która przechwytuje informacje specyficzne dla obrazu. Te dwa rodzaje informacji są często wykorzystywane w dwóch zadaniach: detekcji obiektów i klasyfikacji.

### Klasyfikacja

Klasyfikacja to zadanie uczenia maszynowego polegające na określaniu, które obiekty znajdują się na obrazie lub wideo. Polega na szkoleniu modeli uczenia maszynowego, aby rozpoznawały obecność określonych klas (obiektów). Klasyfikacja jest przydatna na poziomie decyzji tak-nie, czyli stwierdzenia, czy obraz zawiera obiekt/anomalie, czy nie.

Odrębnym zadaniem od klasyfikacji jest lokalizacja lub określanie pozycji sklasyfikowanych obiektów na obrazie lub wideo.





### Wykrywanie/Detekcja obiektów

Wykrywanie obiektów łączy klasyfikację i lokalizację, aby określić, jakie obiekty znajdują się na obrazie lub wideo i gdzie dokładnie się one znajdują. Wykorzystuje klasyfikację dla rozróżnienia różnych obiektów i stosuje prostokątne obszary ograniczające (bounding boxes). Detekcja obiektów jest przydatna do identyfikacji obiektów na obrazie lub wideo. Przykłady zastosowań detekcji obiektów obejmują wykrywanie twarzy wraz z analizą po wykryciu, taką jak rozpoznawanie emocji, oszacowanie wieku lub wykrywanie senności. Istnieje wiele aplikacji detekcji obiektów w czasie rzeczywistym w zarządzaniu ruchem drogowym, na przykład systemy wykrywania pojazdów na podstawie scen ruchu drogowego. Jak opisano powyżej, najpopularniejsze podejścia w dziedzinie widzenia komputerowego to klasyfikacja i detekcja obiektów, które pozwalają zidentyfikować obiekty obecne na obrazie i określić ich pozycję.



## PODSUMOWANIE

Wykrywanie i klasyfikacja obiektów to dwie kluczowe zadania w analizie obrazów.







## Scenariusz lekcji 6

Czas (minuty):  
90

### Tematyka

Programowanie wykrywania obiektów w Scratchu

### Cele

Nauka programowania wykrywania obiektów na wybranych przykładach

### Oczekiwane efekty

Umiejętność napisania programu do wykrywania obiektów w języku Scratch

## Programowanie wykrywania obiektów w Scratchu

*Aby zrozumieć wykrywanie obiektów, przejrzyjmy to, czego się do tej pory nauczyliśmy.*

*Co to jest wykrywanie obiektów?*

*Jak działa wykrywanie obiektów?*

Wykrywanie/Detekcja obiektów to technika przetwarzania obrazu, której celem jest identyfikacja i lokalizacja obiektów na obrazie lub wideo. Konkretnie, detekcja obiektów tworzy ramki otaczające te wykryte obiekty, co pozwala nam określić, gdzie znajdują się te obiekty w danej scenie lub jak się w niej poruszają. Przed rozpoczęciem programowania detekcji obiektów musimy zbierać zdjęcia konkretnych twarzy i trenować model. Będziemy używać aplikacji o nazwie Teachable Machine. Teachable Machine to narzędzie internetowe, które umożliwia szybkie, łatwe i dostępne dla wszystkich tworzenie modeli uczenia maszynowego.

### Przedstawienie celu głównego lekcji:

Zrozumienie programu do wykrywania obiektów i jego zastosowania na przykładach.



# CZĘŚĆ GŁÓWNA

## ZADANIE: Czy to kot czy pies?

Stwórz model i program, który wykryje, czy w strumieniu kamery znajduje się kot lub pies.

### APLIKACJA INTERNETOWA (bez konieczności instalacji oprogramowania)

Krok 1: Otwórz przeglądarkę internetową, wybierz i pobierz obrazy ze strony:

<https://bit.ly/cats-image-dataset>

Będą one używane do treningu Klasy 1

Krok 2: Otwórz przeglądarkę internetową, wybierz i pobierz obrazy ze strony:

<https://bit.ly/dogs-image-dataset>

Będą one używane do treningu Klasy 2

Krok 3: Otwórz przeglądarkę internetową i przejdź do:

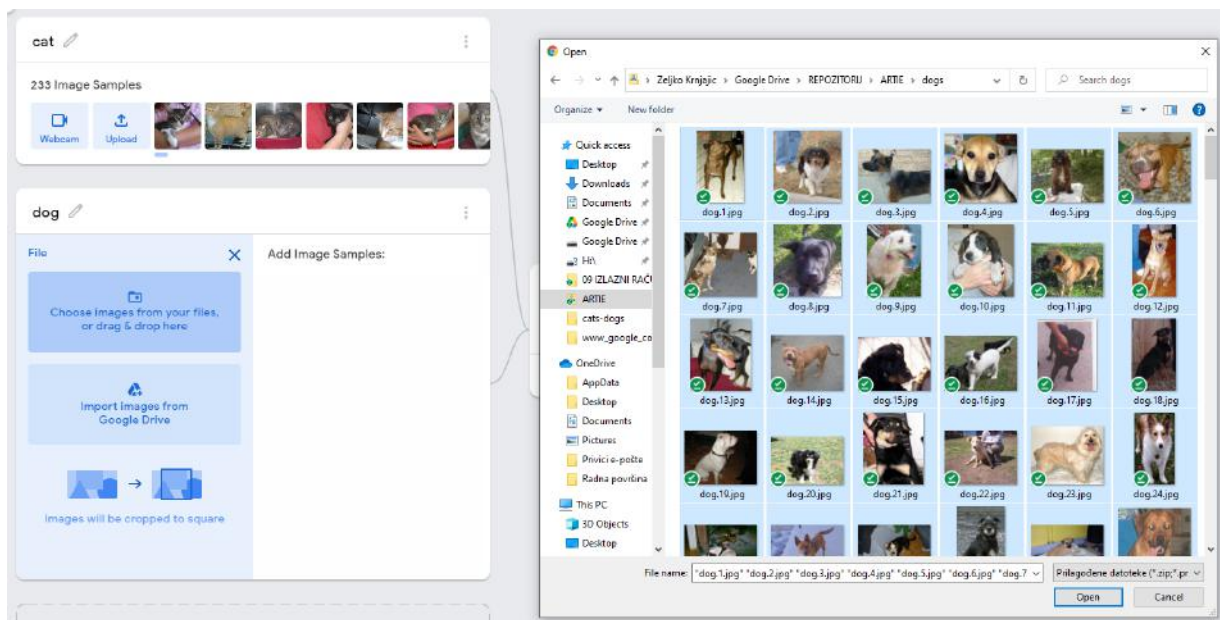
<https://teachablemachine.withgoogle.com/>

Krok 4: Kliknij przycisk Rozpocznij.

Krok 5: Wybierz projekt Obraz.

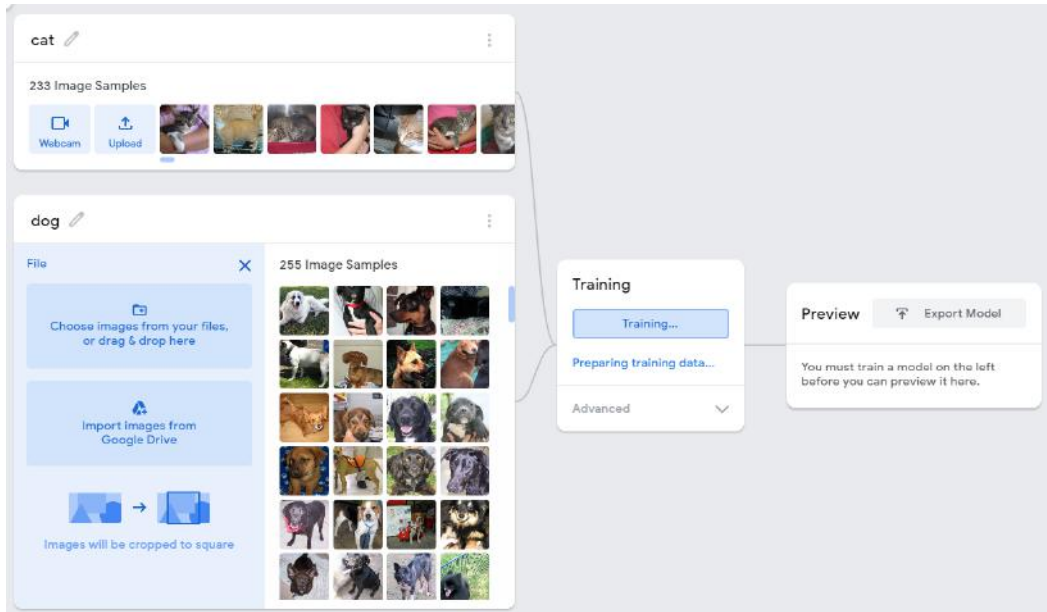
Krok 6: Wybierz standardowy model obrazowy.

Krok 7: Zmień nazwę Klasy 1 na kot i Klasy 2 na pies. Prześlij obrazy kotów do plików kotów, a obrazy psów do plików psów, zgodnie z poniższym obrazkiem.

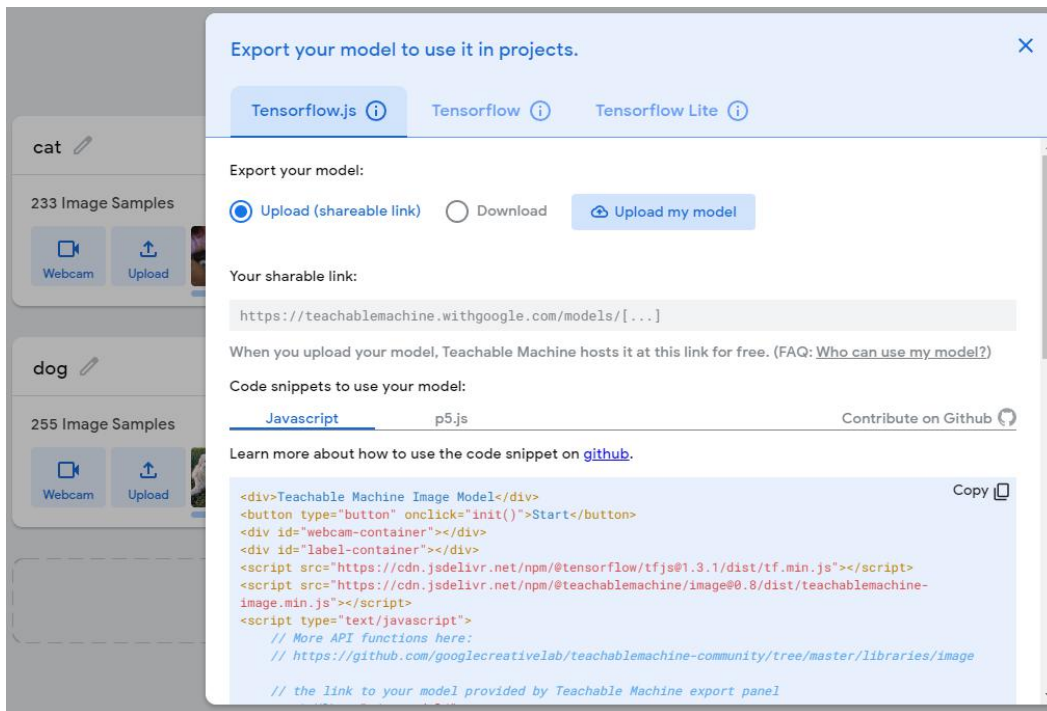




Krok 8: Trenuj swój model. Nie przełączaj kart przeglądarki podczas procesu treningu.



Krok 9: Wyeksportuj swój model. W oknie wyskakującym wybierz opcję przestania go do chmury (trzecia opcja - Prześlij mój model), a Google bezpłatnie przechowa Twoje dane





Krok 10: Skopiuj link podany w polu tekstowym poniżej - jest to adres URL Twojego modelu. W moim przypadku był to <https://teachablemachine.withgoogle.com/models/gS4NT1mgE/>

Upload (shareable link)
  Download

Your sharable link:

When you upload your model, Teachable Machine hosts it at this link for free. (FAQ: [Who can use my model?](#))

✓ Your cloud model is up to date.

Krok 11: Twój model jest gotowy do użycia.

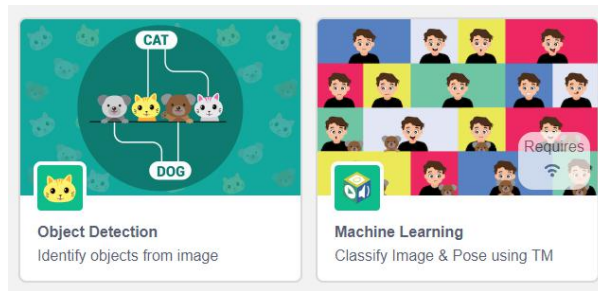
### PICTOBLOX (aplikacja na komputer):

Krok 1a: Pobierz i zainstaluj PictoBlox, ponieważ obecnie jest jedyną aplikacją umożliwiającą wykrywanie obiektów. Jest to aplikacja typu desktop i musisz ją najpierw zainstalować ze strony <https://thetempedia.com/product/pictoblox/download-pictoblox/> (427 Mb)

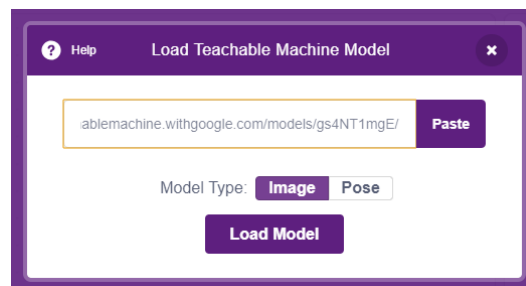
Krok 2a: Wczytaj rozszerzenia "Wykrywanie Obiektów" i "Uczenie Maszynowe".



<http://erasmus-artie.eu>



Krok 3a: Wybierz grupę "Uczenie Maszynowe" i wybierz opcję "Załaduj Model". Wklej link do modelu z Teachable Machine (poniżej): <https://teachablemachine.withgoogle.com/models/gS4NT1mgE/> i kliknij na "Załaduj Model".





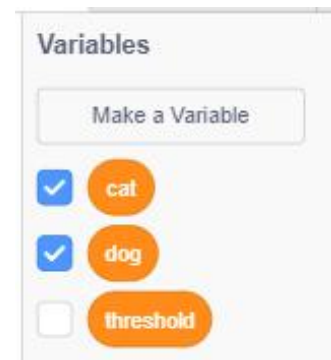
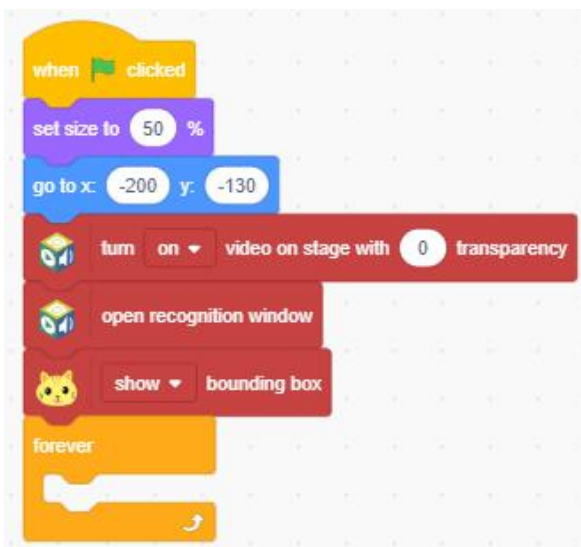
Krok 4a: Wykonamy jednocześnie klasyfikację i wykrywanie obiektów. Na początek uruchom program, zmień rozmiar i przesuń duszka o nazwie "Tobi" w prawy dolny róg ekranu. Następnie użyj kilku bloków z rozszerzenia "Uczenie Maszynowe" do włączenia wideo i otwarcia okna rozpoznawania. Jeśli Twoje wideo jest odwrócone, użyj opcji "Włącz odwrócone wideo na scenie" z przezroczystością 0.



Krok 5a: Z rozszerzenia "Wykrywanie Obiektów" użyj bloku "Pokaż obramowanie" ("show bounding box") w celu wyświetlenia położenia obiektu na strumieniu wideo. Dodaj także blok pętli "zawsze" ("forever loop").



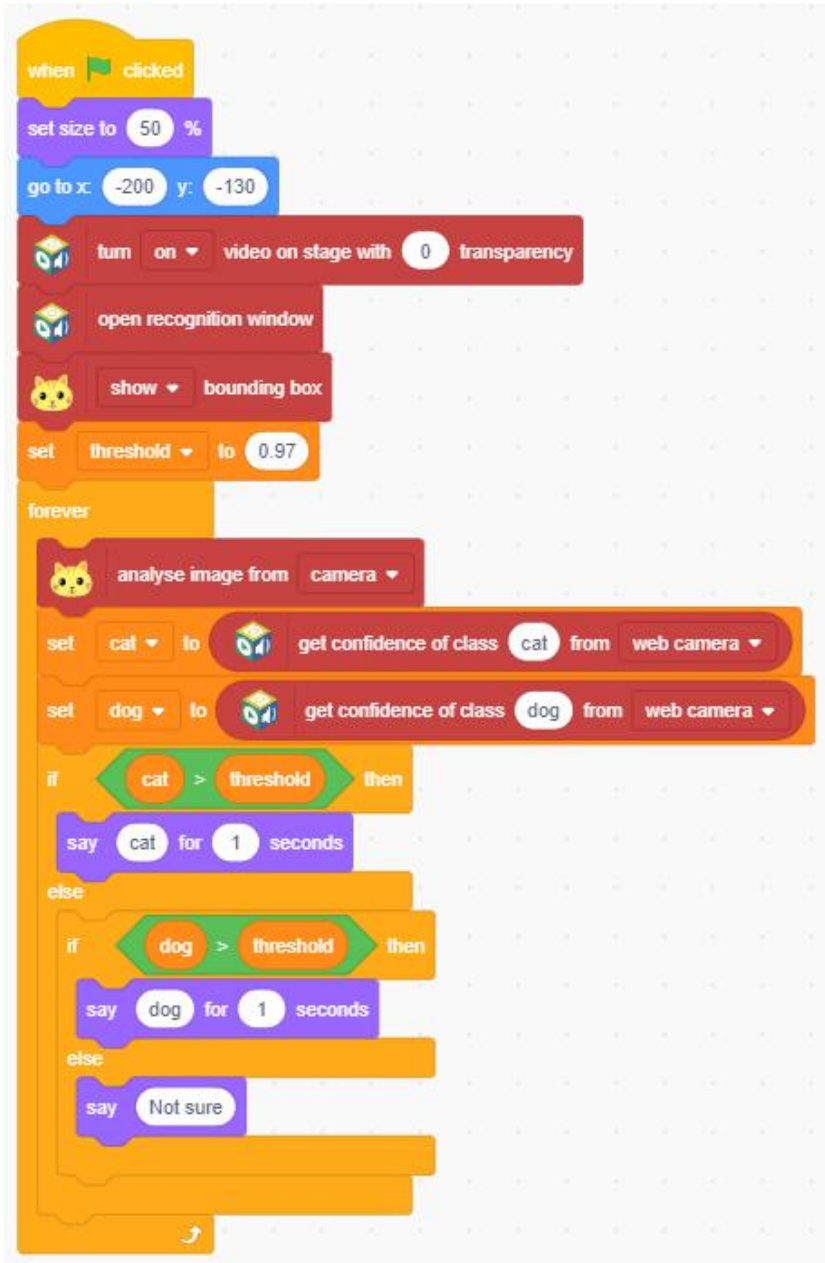
Krok 6a: Stwórz 3 zmienne: "cat" (kot), "dog" (pies) i "threshold" (próg). Wyświetl "cat" i "dog" na scenie, zaznaczając je. Pozostaw opcję "threshold" odznaczoną..



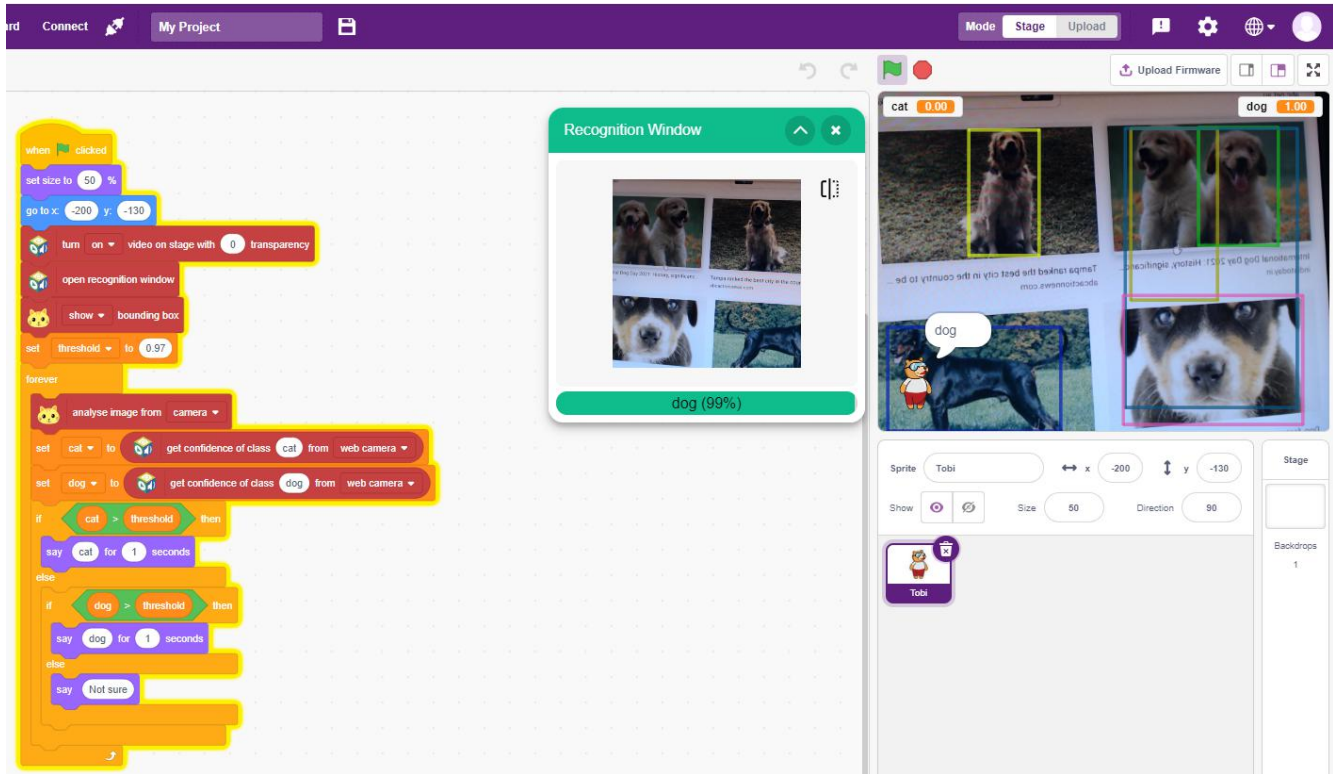


Krok 7a: Na początku, przed blokiem pętli "zawsze" ("forever loop"), wstaw blok "ustaw próg na 0.97" ("set threshold to 0.97"). Teraz przejdźmy do głównej części, w której analizujemy obraz z kamery i zapisujemy pewność wartości klas w zmiennych.





Krok 9a: Uruchom program i przetestuj go na swoim kocie lub psie. Jeśli nie masz żadnego zwierzaka w pobliżu, możesz skorzystać z Google Images - wystarczy wyszukać zdjęcia psów lub kotów i skierować kamerę internetową na ekran z wynikami wyszukiwania. Zobaczysz ramki otaczające wykryte obiekty. Spróbuj obniżyć lub podnieść wartość progu lub spróbuj wprowadzić utrudnienie w algorytm detekcji, korzystając z niektórych zdjęć konkretnych ras kotów lub psów.



### **Czym więc dokładnie jest wykrywanie obiektów?**

Aby odpowiedzieć na to pytanie, zacznijmy od klasyfikacji obrazów. W tym zadaniu mamy obraz i chcemy przypisać go do jednej z wielu różnych kategorii (np. samochód, pies, kot, człowiek...), więc w zasadzie chcemy odpowiedzieć na pytanie "Co jest na tym obrazie?". Należy zauważyć, że jeden obraz ma przypisaną tylko jedną kategorię. W prostych słowach, detekcja obiektów to rodzaj techniki klasyfikacji obrazów, a oprócz klasyfikacji ta technika identyfikuje również położenie instancji obiektu spośród dużej liczby predefiniowanych kategorii na obrazach naturalnych.

## **PODSUMOWANIE**

Wykrywanie obiektów jest techniką klasyfikacji obrazów, która dodatkowo identyfikuje pozycję znanego obiektu spośród dużej liczby wcześniej zdefiniowanych kategorii.





## Scenariusz lekcji 7

### Projekt klasyfikacji obiektów

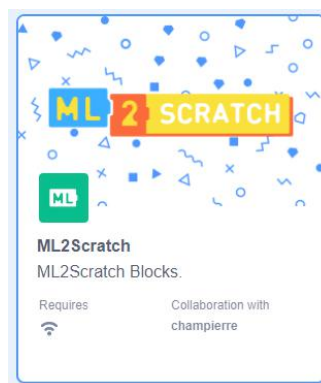
*Co to jest wykrywanie obiektów?  
Jak działa wykrywanie obiektów?*

**Przedstawienie celu głównego lekcji:**  
Zrozumienie procedury klasyfikacji obiektów oraz jej zastosowanie w praktycznych projektach.

# CZĘŚĆ GŁÓWNA

Krok 1: Podłącz kamerę internetową.

Krok 2: Otwórz Scratch na stronie <https://stretch3.github.io/> i dodaj rozszerzenie "ML2SCRATCH"



Czas (minuty):  
90

### Tematyka

Interpretacja wyników procedury klasyfikacji obiektów

### Cele

Praktyczne zastosowanie klasyfikacji obiektów

### Oczekiwane efekty

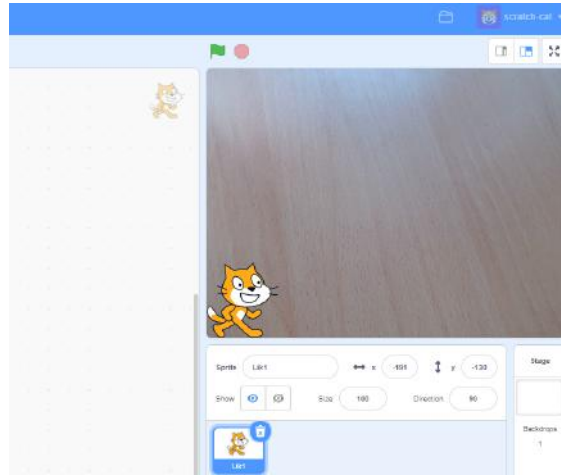
Badanie możliwości rozszerzenia uczenia maszynowego w Scratchu



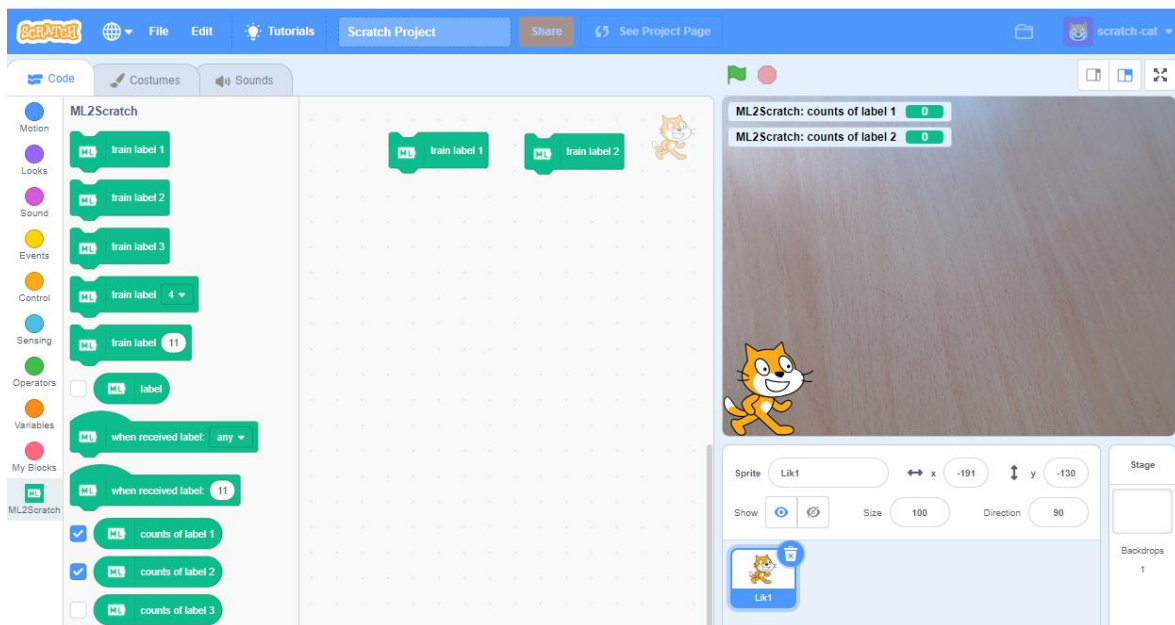
<http://erasmus-artie.eu>



Krok 3: Uporządkuj biurko, podłącz kamerę internetową i skieruj ją na puste miejsce. Przesuń duszka kota do rogu, jak pokazano na obrazku.



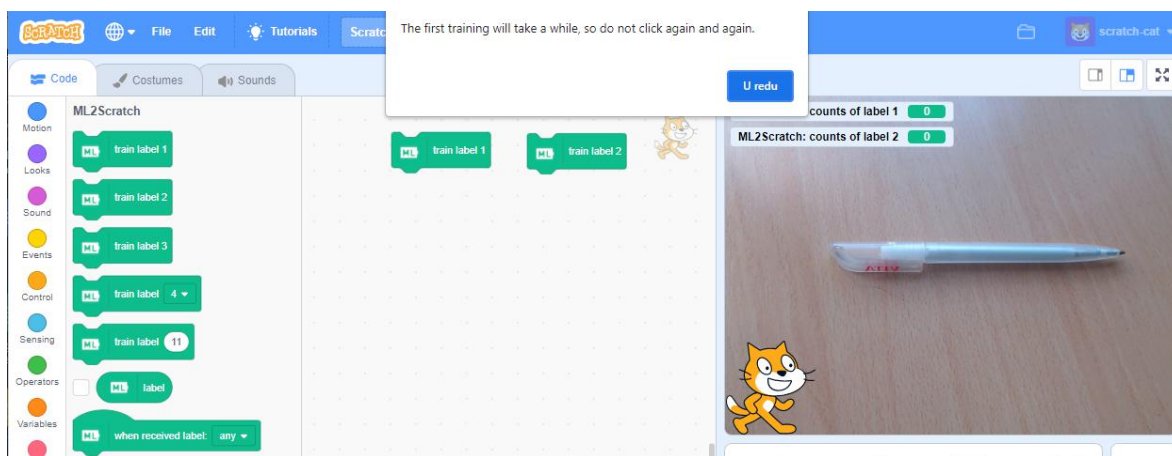
Krok 4: Z grupy ML2SCRATCH wybierz bloki "Train Label 1" i "Train Label 2" i umieść je na obszarze programowania. Sprawdź liczbę etykiet 1 (Label 1) i liczbę etykiet 2 (Label 2), jak pokazano na obrazku.



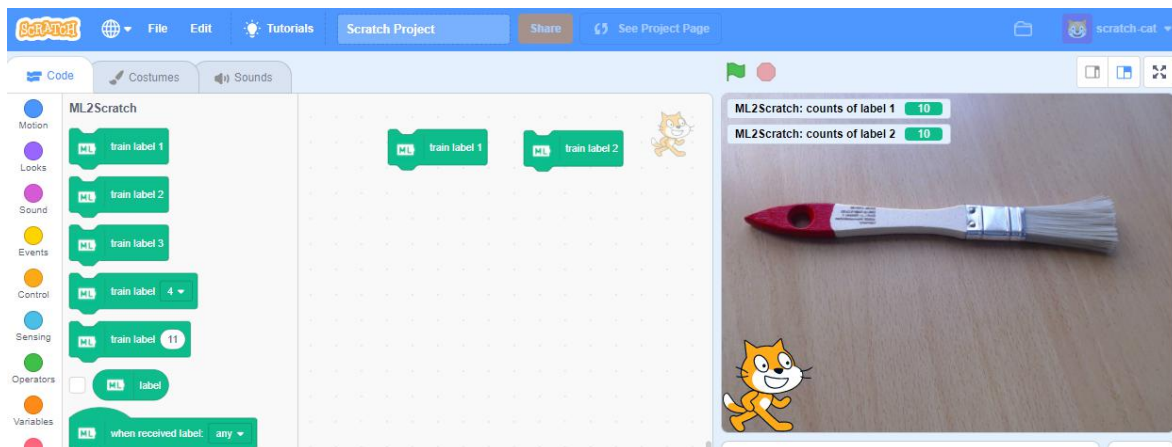


Krok 5: Przygotuj dwie grupy obiektów do uczenia maszynowego. W naszym przypadku użyjemy pędzli i długopisów do trenowania etykiet.

Krok 6: Umieść pierwszy obiekt z pierwszej grupy w obszarze, na który jest skierowana kamera internetowa, a następnie kliknij blok "Train Label 1". Otrzymasz powiadomienie o konieczności chwili oczekiwania, jak pokazano na obrazku, a liczba etykiet zmieni się na 1.

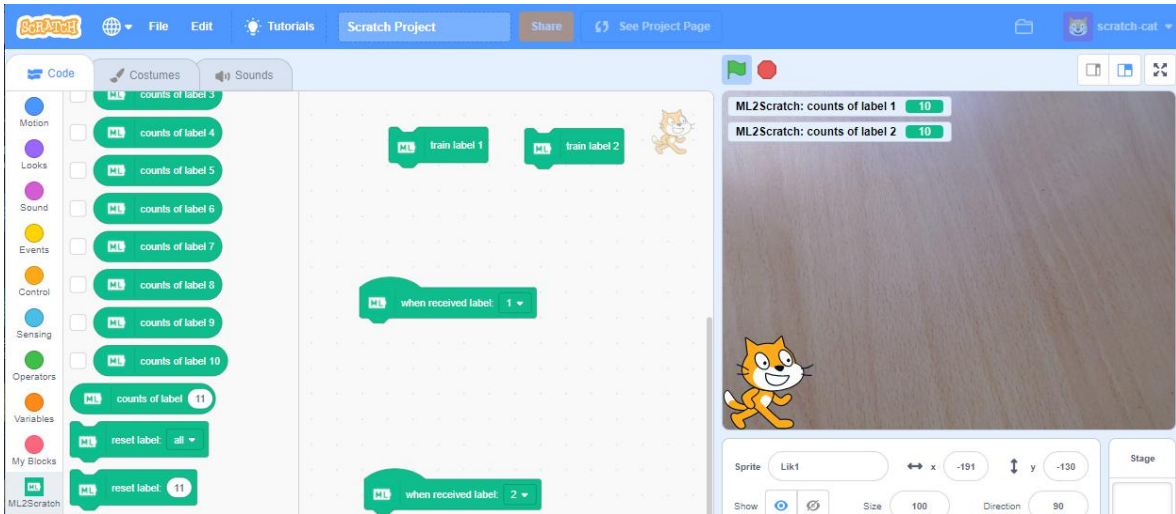


Krok 7: Wykonaj około 10 zdjęć każdego obiektu, zmieniając ich kształt i położenie. Pamiętaj, aby prawidłowo przypisać etykietę do każdego obiektu i nie mieszać obiektów z etykietami.



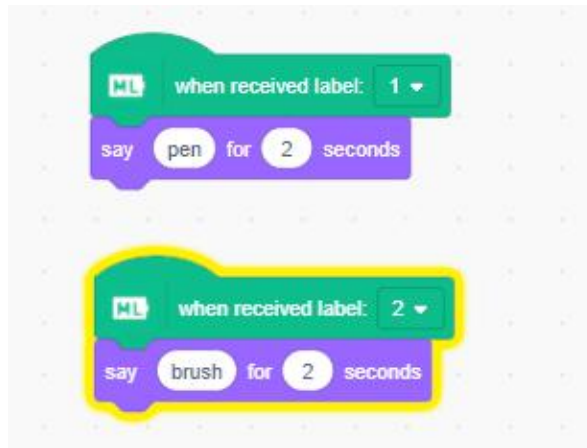


Krok 8: Teraz wybierz dwa bloki "when received label: any" i zmień "any" na 1 w pierwszym bloku i na 2 w drugim bloku, jak pokazano na obrazku.



f

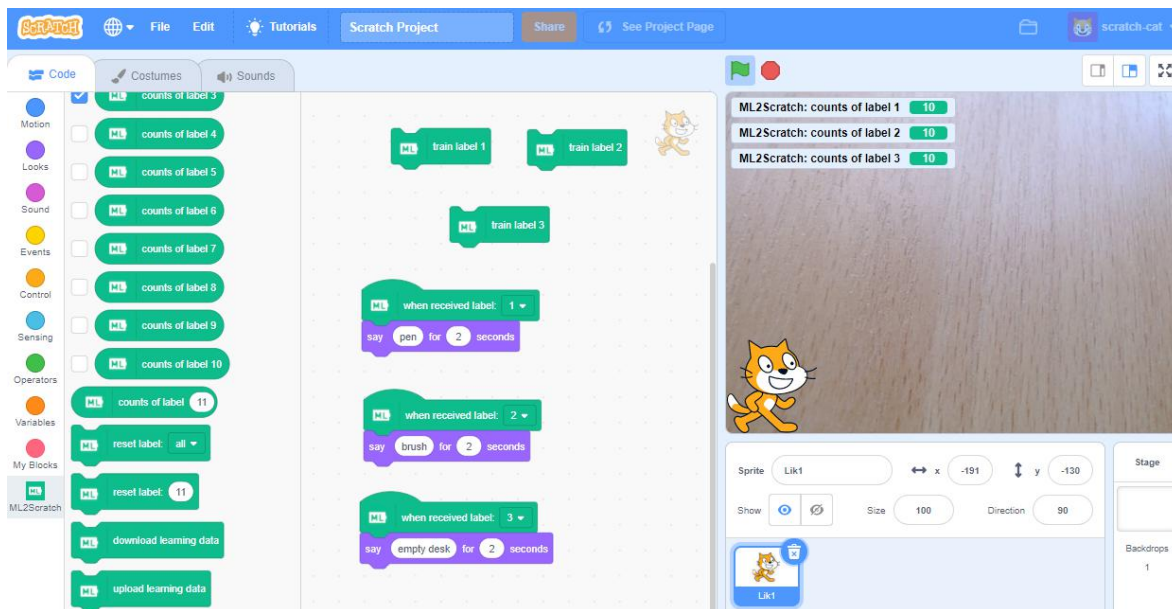
Krok 9: Wybierz z grupy "Look" dwa bloki "say Hello! for 2 seconds" i zmień "Hello!" na "pen" w pierwszym bloku oraz na "brush" w drugim bloku, jak pokazano na obrazku.





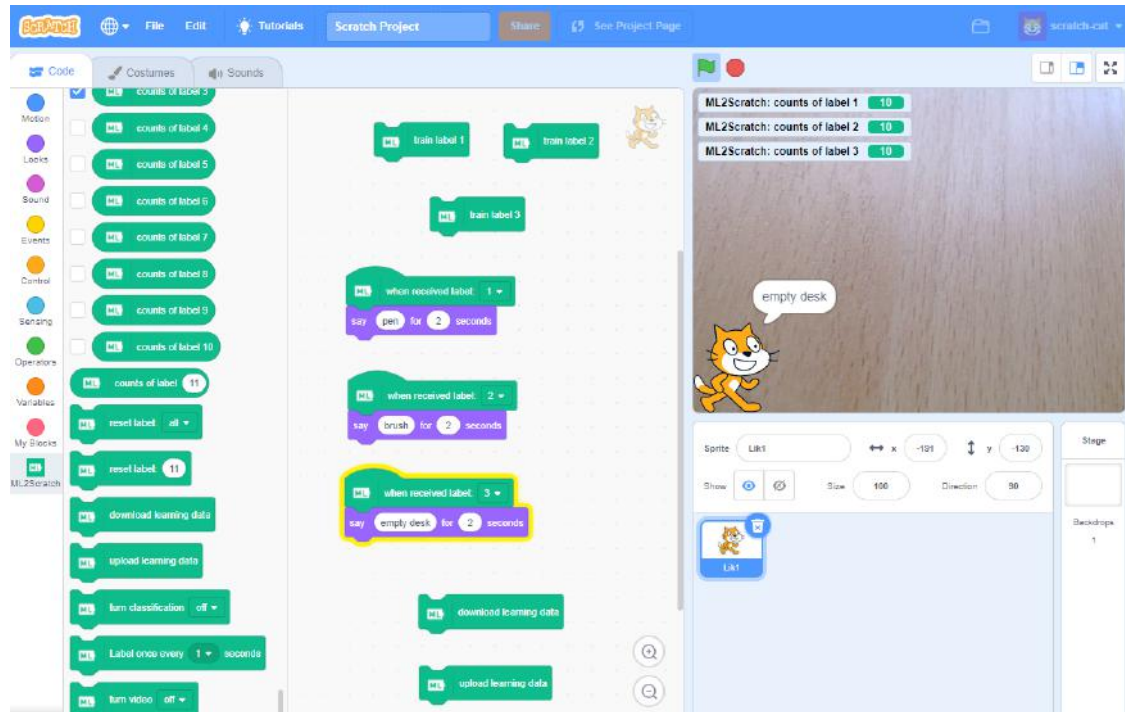
Krok 10: Umieść losowo obiekty z dwóch grup w obszarze widocznym dla kamery i zauważ, co się dzieje. Czy działa poprawnie? Co się dzieje, gdy nie ma umieszczonych żadnych obiektów? Jak rozwiązać problem pustego biurka (empty desk glitch)?

Krok 11: Rozwiązanie: Użyj trzeciej etykiety, naucz ją rozpoznawania pustego biurka i dodaj blok "when received label 3" z blokiem "say empty desk for 2 seconds", jak pokazano na obrazku.



Krok 12: Omów dokładność predykcji. Spróbuj trenować każdą etykietę bardziej i porównaj wyniki z poprzednimi. Czy dokładność poprawia się?

Krok 13: Użyj większej liczby grup obiektów, aby trenować więcej etykiet. Pobierz i przełącz swoje dane treningowe za pomocą bloków "download learning data" i "upload learning data" (po prostu kliknij na blok, aby zapisać lub przestać plik .json).



W prostych słowach, klasyfikacja obrazów to technika używana do klasyfikowania lub przewidywania klasy konkretnego obiektu na obrazie. Głównym celem tej techniki jest dokładne identyfikowanie cech w obrazie.

W ogólnym zakresie techniki klasyfikacji obrazów można podzielić na parametryczne i nieparametryczne, nadzorowane i nienadzorowane oraz na twarde i miękkie klasyfikatory. W przypadku klasyfikacji nadzorowanej, technika ta dostarcza wyników na podstawie granicy decyzyjnej utworzonej głównie na podstawie danych wejściowych i wyjściowych podanych podczas trenowania modelu. Natomiast w przypadku klasyfikacji nienadzorowanej, technika ta dostarcza wyników na podstawie analizy zbioru danych wejściowych; cechy nie są bezpośrednio podawane do modeli. Główne kroki związane z technikami klasyfikacji obrazów obejmują wybór odpowiedniego systemu klasyfikacji, ekstrakcję cech, wybór dobrych próbek treningowych, wstępną obróbkę obrazu, wybór odpowiedniej metody klasyfikacji, przetwarzanie po klasyfikacji oraz ocenę ogólnej dokładności. W tej technice, dane wejściowe to zazwyczaj obraz określonego obiektu, a dane wyjściowe to przewidywane klasy definiujące i dopasowujące obiekty wejściowe.

## PODSUMOWANIE

Klasifikacija slike je tehnika koja se koristi za klasificiranje ili predviđanje klase određenog objekta na slici. Glavni cilj ove tehnike je točno identificirati značajke na slici.



## Scenariusz lekcji 8

Czas (minuty):  
90

### Tematyka

Rozpoznawanie i generowanie mowy dla początkujących w Scratchu

### Cele

Nauka o rozpoznawaniu i generowaniu mowy dla początkujących w Scratchu

### Oczekiwane efekty

Zrozumienie rozpoznawania i generowania mowy za pomocą prostego przykładu w Scratchu.

## Rozpoznawanie i generowanie mowy w Scratchu dla początkujących

### Wprowadzenie do Rozpoznawania Mowy

**Rozpoznawanie Mowy** to umiejętność przekształcania wypowiedzi mówionej na tekst. Jest to również znane jako konwersja mowy na tekst (STT) oraz rozpoznawanie głosu.

Osiąga się to poprzez realizację określonych kroków, a oprogramowanie odpowiedzialne za to nazywane jest "Systemem Rozpoznawania Mowy" (SR). Systemy SR są zazwyczaj wdrażane w postaci oprogramowania do dyktowania i inteligentnych asystentów w komputerach osobistych, smartfonach, przeglądarkach internetowych i wielu innych urządzeniach.

### Wprowadzenie do Generowania Mowy

**Generowanie mowy** lub synteza mowy (również skrótowo TTS, Tekast-do-Mowy) w przeciwieństwie do rozpoznawania mowy, to nie technologia, która wykorzystuje głos, ale go produkuje. Sztuczne głosy są zazwyczaj ostatnim etapem procesu i stają się coraz bardziej powszechne, ponieważ są ważne dla ogólnego doświadczenia "głosu".

Synteza mowy (TTS) jest definiowana jako sztuczne generowanie ludzkich głosów. Głównym zastosowaniem (i tym, co skłoniło do jego stworzenia), jest umożliwienie automatycznego przekształcania tekstu na mowę.

### Przedstawienie celu głównego lekcji:

Wprowadzenie do rozpoznawania i generowania mowy dla początkujących na przykładzie jednego prostego programu w Makeblock.



<http://erasmus-artie.eu>



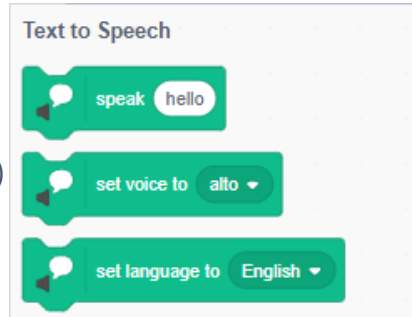
# CZĘŚĆ GŁÓWNA

## Aplikacje

### Scratch (MIT)

<https://scratch.mit.edu/projects/editor/>

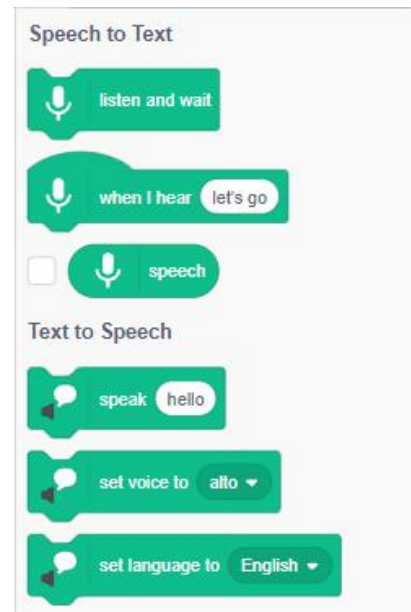
Dostępne jest tylko rozszerzenie Text-to-Speech (3 bloki)



### Scratch (ML4KIDS)

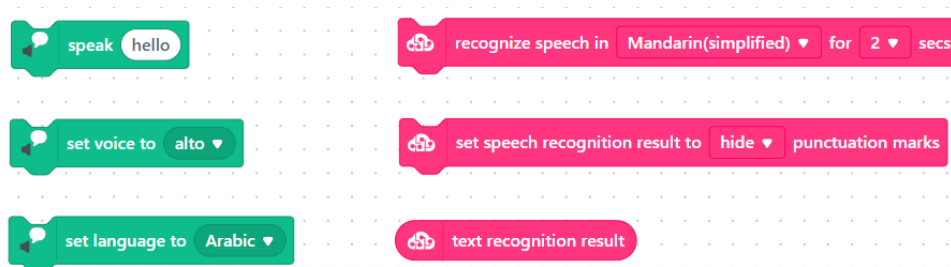
<https://machinelearningforkids.co.uk/scratch3/>

Dostępne są rozszerzenia Text-to-Speech (3 bloki) i Speech-to-Text (3 bloki)



Makeblock - <https://ide.mblock.cc/>

Text-to-Speech (3 bloki), Cognitive Services (3 bloki związane z Speech-to-Text)

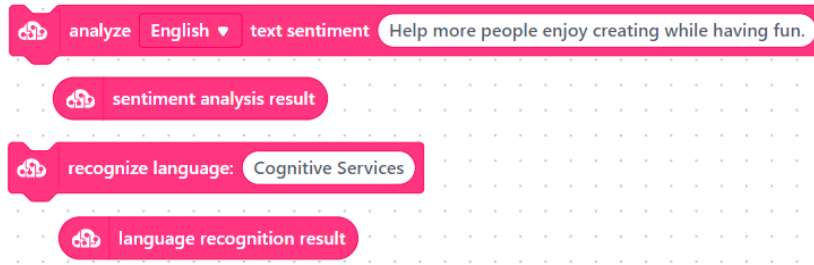






*\*BONUS*

Makeblock zawiera również bloki do rozpoznawania języka i analizy sentymentu tekstu.



**Analiza sentymentu** jest procesem wykrywania pozytywnego lub negatywnego nastroju w tekście. Jest często wykorzystywana przez firmy do wykrywania sentymentu w danych społecznościowych, oceny reputacji marki i zrozumienia potrzeb klientów. Modele analizy sentymentu skupiają się na polaryzacji (pozytywna, negatywna, neutralna), ale także na uczuciach i emocjach (złość, radość, smutek itp.), pilności (pilne, niepilne) i nawet intencjach (zainteresowany vs niezainteresowany).

Dowiedz się więcej o analizie sentymentu: <https://monkeylearn.com/sentiment-analysis/>

Aby to zobrazować, weźmy przykład z Makeblock.

Krok 1: Otwórz stronę Makeblock: <https://ide.mblock.cc/>

Krok 2: Dodaj rozszerzenia: Usługi poznawcze (Cognitive services) i Tekst do mowy (Text to Speech)

Krok 3: Sprawdź, czy wyświetlane są następujące bloki typu reporter:

- ✓ Wynik rozpoznawania mowy (speech recognition result)
- ✓ Wynik rozpoznawania języka (language recognition result)
- ✓ Wynik analizy sentymentu (sentiment analysis result)

Krok 4: Wykorzystaj **wynik rozpoznawania mowy** do **rozpoznawania języka** i analizowania bloków sentymentu tekstu

Krok 5: Użyj tej sekwencji bloków





```
when clicked
  recognize speech in English for 2 secs
  recognize language: speech recognition result
  analyze English text sentiment speech recognition result
  wait 2 seconds
  set voice to tenor
  set language to English
  if sentiment analysis result = negative then
    speak This is not good
  if sentiment analysis result = positive then
    speak This is good
```

**f**

Krok 6: Włącz mikrofon i głośniki, uruchom kod i powiedz "Jestem szczęśliwy/a" (pojawi się okno dialogowe nagrywające Twoją mowę przez 2 sekundy - jeśli potrzebujesz więcej, zwiększ wartość w bloku **rozpoznawania mowy w języku angielskim na 3 sekundy** (lub dłużej)). Usłyszysz odpowiedź "To jest dobre" lub "To nie jest dobre", w zależności od wyników analizy sentymentu.





Krok 7: Uruchom kod ponownie, powiedz coś innego i poczekaj na wyniki analizy sentymentu.

### Podstawowe zasady rozpoznawania mowy

Najmniejszą jednostką mówionego języka jest fonem. Język angielski zawiera około 44 fonemy, które reprezentują samogłoski i spółgłoski używane w mowie. Możemy przyjąć jako przykład typowe słowo takie jak "moon", które można podzielić na trzy fonemy: m, ue, n.

Aby interpretować mowę, musimy mieć sposób na identyfikację składników wypowiedzianych słów i fonemów, które pełnią rolę znaczników identyfikacyjnych w mowie. Do dalszej interpretacji mowy konieczne jest użycie odpowiedniego algorytmu. Ukryty Model Markowa (Hidden Markov Model) to powszechnie stosowany matematyczny model, który to umożliwia. Aby stworzyć silnik rozpoznawania mowy, tworzy się dużą bazę modeli, które odpowiadają każdemu fonemowi.

Dowiedz się więcej: <https://www.ibm.com/cloud/learn/speech-recognition>

### Podstawowe zasady generowania mowy

W przeciwieństwie do systemów rozpoznawania mowy, które w pierwszej kolejności korzystają z fonemów (najmniejszych jednostek dźwiękowych) do wyodrębnienia zdań, TTS (tekst do mowy) będzie oparte na tym, co nazywamy grafemami: literami i grupami liter, które transkrybują fonem. Oznacza to, że podstawowym źródłem nie jest dźwięk, lecz tekst. Proces ten zazwyczaj odbywa się w dwóch etapach.

Pierwszy etap polega na podzieleniu tekstu na zdania i słowa (nasze znane grafemy) oraz przypisaniu transkrypcji fonetycznej, czyli wymowy, do tych grup. Gdy różne grupy tekstowe/fonetyczne zostaną zidentyfikowane, drugi etap polega na przekształceniu tych reprezentacji lingwistycznych w dźwięk. Innymi słowy, odczytanie tych wskazówek w celu wygenerowania głosu, który przeczyta informacje.

Wypróbuj TTS online <https://www.readspeaker.com/>

Rozpoznawanie mowy to zdolność przekształcania mówionych słów na tekst. Jest również znane jako przekształcanie mowy na tekst (Speech-to-Text) oraz rozpoznawanie głosu. Osiąga się to poprzez przestrzeganie określonych kroków, a oprogramowanie odpowiedzialne za to nazywane jest "Systemem Rozpoznawania Mowy". Systemy rozpoznawania mowy są zazwyczaj wdrażane w postaci oprogramowania do dyktowania oraz inteligentnych asystentów w komputerach osobistych, smartfonach, przeglądarkach internetowych i wielu innych urządzeniach.

## PODSUMOWANIE

Rozpoznawanie mowy to zdolność przekładania wypowiedzi ustnych na tekst. Generowanie mowy to technologia, która tworzy głos.



## Scenariusz lekcji 9

Czas (minuty):  
90

### Tematyka

Programowanie rozpoznawania mowy w Scratchu

### Cele

Nauka programowania rozpoznawania mowy na przestanych przykładach

### Oczekiwane efekty

Wiedza, jak napisać program do generowania mowy za pomocą Scratch

## Programowanie rozpoznawania mowy w Scratchu

*Zapytaj uczniów, czym jest rozpoznawanie mowy?  
Zapytaj uczniów, czym jest SIRI i Google NOW?  
Czy znają jakiś inny system rozpoznawania mowy?*

W ostatnich latach technologia rozpoznawania mowy stała się coraz bardziej powszechna. Ta technologia jest często wykorzystywana zarówno przez firmy, jak i osoby prywatne, ze względu na liczne korzyści, które przynosi.

„Hej Siri”, „OK Google” i tak dalej - rozpoznawanie głosu, często nazywane technologią rozpoznawania mowy, to nie nowy koncept (SRT). Odnosi się do rodzaju technologii, która potrafi przekształcać mówione słowa w formy czytelne dla maszyn. Teraz możesz komunikować się ze swoimi urządzeniami i sprawiać, że wykonują one Twoje polecenia, podobnie jak w opowieściach science fiction. Ponieważ większość technologii rozpoznawania mowy ma dokładność powyżej 95%, nie dziwi, że najnowsze statystyki dotyczące wyszukiwania głosowego ujawniają, że prawie 50% wszystkich wyszukiwań w 2022 roku jest przeprowadzanych za pomocą mowy.

#### Przedstawienie celu głównego lekcji:

Wprowadzenie do programu rozpoznawania mowy i jego wykorzystanie na przykładzie jednego programu.



<http://erasmus-artie.eu>

# CZĘŚĆ GŁÓWNA

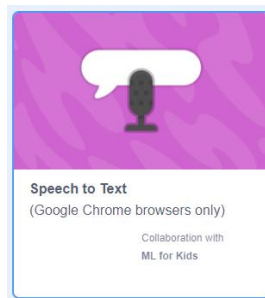
Jeśli chcemy pracować z rozpoznawaniem mowy, dostępne są rozszerzenia w aplikacjach Scratch i Makeblock.

## SCRATCH (ML4KIDS):

Krok 1: Otwórz swoją przeglądarkę internetową Chrome i przejdź pod adres:

<https://machinelearningforkids.co.uk/scratch3/>

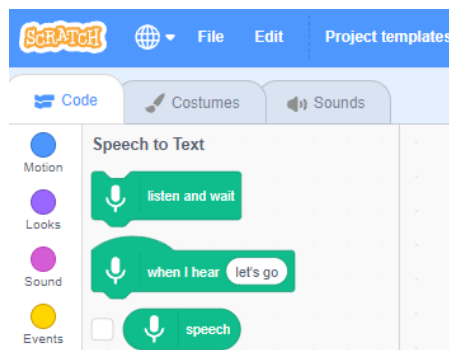
Krok 2: Załaduj rozszerzenie Mowa na Tekst (Speech to Text).



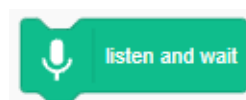
Krok 3: Zobaczysz nową grupę w Palecie Bloków o nazwie „Mowa na Tekst” (Speech to Text) oraz trzy nowe bloki w tej grupie.



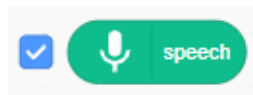
<http://erasmus-artie.eu>



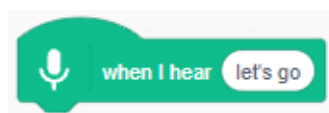
Krok 4: **Blok słuchaj i czekaj** rozpoczyna nasłuchiwanie i przetwarzanie wypowiedzianych słów.



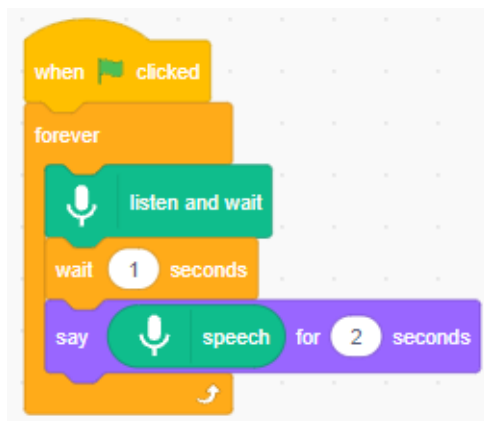
Krok 5: Wyniki rozpoznawania mowy są wyświetlane w bloku Raporter mowy (Speech Reporter). Sprawdź, czy chcesz, aby rezultat rozpoznawania mowy był wyświetlany na scenie.



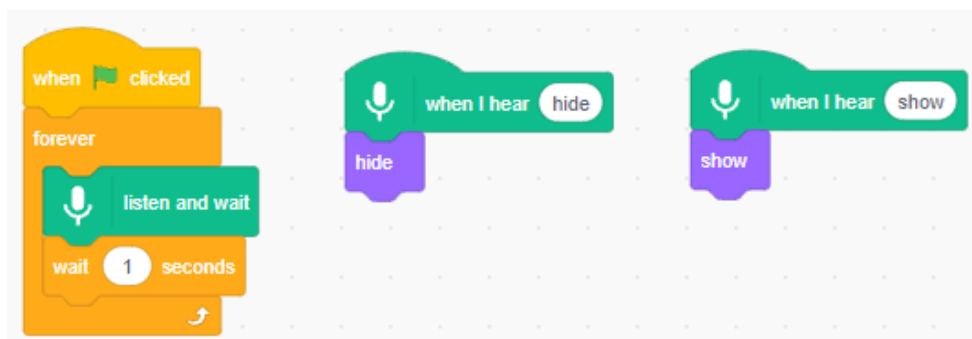
Krok 6: Ostatnim blokiem jest blok wyzwalacza zdarzenia (Event Trigger). Ten blok oczekuje na słowo w białej chmurce, na przykład „coś” w tym przykładzie, a następnie wykonuje sekwencję bloków do niego podłączonych.



Krok 7: Więc stwórzmy prosty program „słuchaj i mów”. Wystarczy umieścić blok **Słuchaj i czekaj** w pętli wraz z blokiem **Powiedz**, aby zobaczyć działanie rozpoznawania mowy.



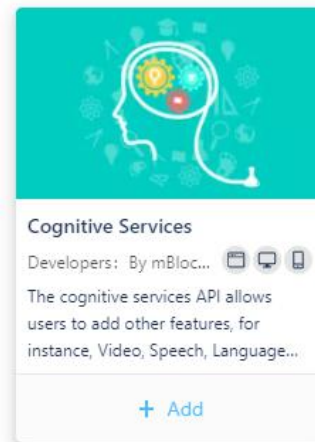
Krok 8: Możesz również oczekiwać na konkretne słowo lub słowa, aby wywołać zdarzenia. Na przykład w grze w chowanego.



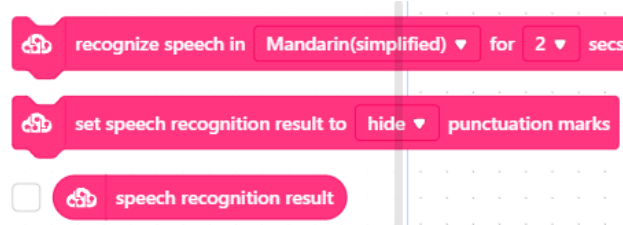
## MAKEBLOCK

Krok 1a: Otwórz przeglądarkę internetową i przejdź pod adres: <https://ide.mblock.cc/>

Krok 2a: Załaduj rozszerzenie Sprite Cognitive Services.



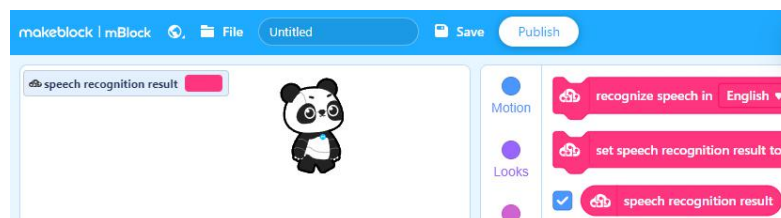
Krok 3a: W tej grupie znajduje się wiele bloków, ale tylko kilka z nich dotyczy rozpoznawania mowy.



Krok 4a: **Blok Rozpoznaj mowę w języku <język> przez <x> sekund** rozpoczyna rozpoznawanie mowy przez kilka sekund. Na ekranie pojawi się okno RECOGNITION, a podczas mówienia zobaczysz wykres fali dźwiękowej.



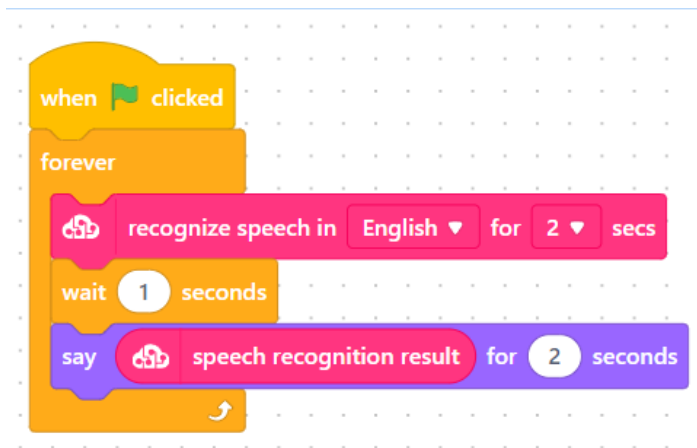
Krok 5a: Wyniki rozpoznawania mowy są wyświetlane w bloku Wynik rozpoznawania mowy. Sprawdź, czy chcesz, aby wynik rozpoznawania mowy był wyświetlany na scenie.



Krok 6a: Ostatni blok będzie wyświetlał lub ukrywał znaki interpunkcyjne w wyniku rozpoznawania mowy.



Krok 7a: Teraz stwórzmy prosty program "słuchaj i mów" w Makeblock. Zobaczysz, że jest bardzo podobny do wersji Scratch.



Technologia rozpoznawania mowy jest obecnie częścią naszego codziennego życia, choć wciąż ogranicza się do prostych poleceń. W miarę postępu technologicznego, badacze będą w stanie opracować bardziej zaawansowane systemy, które interpretują mowę w sposób konwersacyjny. Pewnego dnia będziecie mogli prowadzić rozmowę z komputerem tak samo, jak rozmawiacie z człowiekiem, a komputer będzie odpowiadał w sposób racjonalny. Technologia przetwarzania sygnałów uczyni to wszystko możliwym. Wzrasta zapotrzebowanie na specjalistów w tej dziedzinie, a wiele organizacji poszukuje utalentowanych osób do współpracy. Wiele nowych potężnych technologii i sposobów komunikacji opiera się na przetwarzaniu, interpretacji i zrozumieniu sygnałów głosowych. Biorąc pod uwagę obecne trendy, rozpoznawanie mowy będzie nadal dynamicznie rozwijającą się dziedziną przetwarzania sygnałów w kolejnych latach.

## PODSUMOWANIE

Technologia rozpoznawania mowy umożliwi komunikację głosową pomiędzy użytkownikami a komputerami.







## Scenariusz lekcji 10

Czas (minuty):  
90

### Tematyka

Programowanie generowania mowy w Scratchu

### Cele

Nauka programowania generowania mowy na wybranych przykładach

### Oczekiwane efekty

Umiejętność napisania programu do generowania mowy w języku Scratch

## Programowanie generowania mowy w Scratchu

Text-to-speech (TTS) to rodzaj technologii wspomagającej, która odczytuje cyfrowy tekst na głos. Czasami nazywana jest technologią "czytania na głos". Za pomocą jednego kliknięcia przycisku lub dotknięcia ekranu, TTS może przekształcić słowa z komputera lub innego urządzenia cyfrowego na dźwięk. TTS jest bardzo pomocne dla dzieci i dorosłych mających trudności z czytaniem. Ale może także pomagać w pisaniu, edytowaniu i skupianiu uwagi.

System Concept-to-Speech (CTS) przekształca konceptualne reprezentacje zdań na mowę. Choć niektóre systemy CTS składają się z niezależnych modułów generowania tekstu i przekształcania tekstu na mowę (TTS), większość istniejących systemów CTS poprawia połączenie między tymi dwoma modułami za pomocą modułu przewidywania prozodycznego, który wykorzystuje wiedzę językową z generatora tekstu do przewidywania cech prozodycznych generowanej mowy TTS.

Generowanie mowy polega na przekształcaniu dowolnego tekstu w mowę.

Generowanie mowy polega na produkowaniu wypowiedzi ustnych w odpowiedzi na sygnały z systemu przetwarzania danych lub systemu sterowania. Wybór wiadomości jest dokonywany poprzez składanie dźwięków mowy z zestawu podstawowych elementów, które mogą mieć charakter sztuczny lub zostały wyodrębnione poprzez przetwarzanie ludzkich wypowiedzi.

#### Przedstawienie celu głównego lekcji:

Wprowadzenie do programu generowania mowy i jego wykorzystania na przykładzie jednego programu.



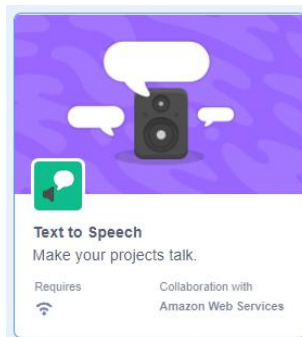
<http://erasmus-artie.eu>



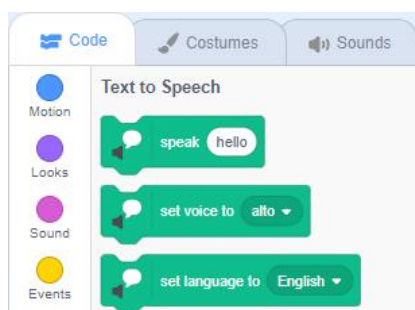
# CZĘŚĆ GŁÓWNA

Krok 1: Otwórz przeglądarkę internetową Chrome i przejdź do strony:  
<https://machinelearningforkids.co.uk/scratch3/>

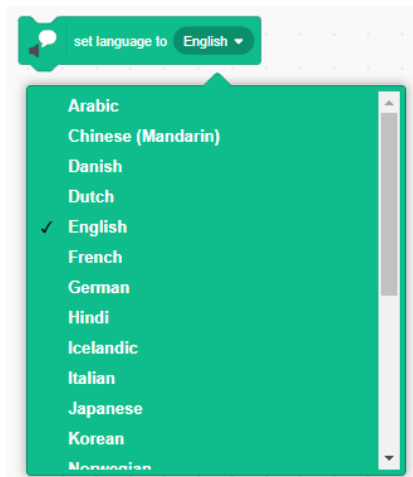
Krok 2: Wczytaj rozszerzenie "Text to Speech"



Krok 3: Zobaczysz nową grupę w Palecie Bloków o nazwie "Text to Speech" i trzy nowe bloki w tej grupie.

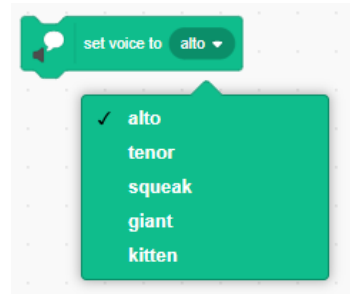


Krok 4: Przejdźmy od dołu - ustaw blok **Language**. Ten blok ustawia język wyjściowy - możesz wybrać go z listy rozwijanej.

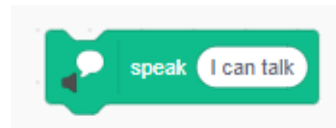




Krok 5: Kolejny blok - **set voice** (ustaw głos).  
Możesz wybrać: alto (alt), tenor (tenor),  
squeak (pisk), giant (olbrzym) lub kitten (kotek).



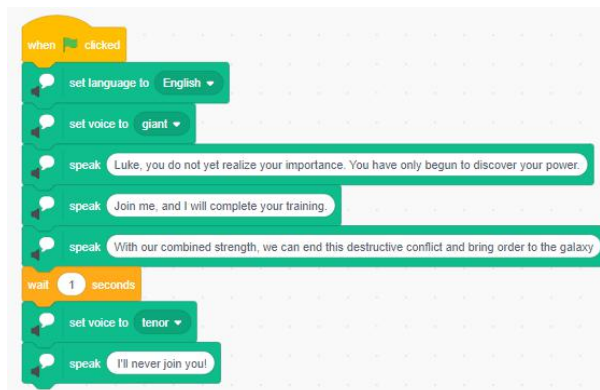
Krok 6: I najważniejszy blok - **speak** (mów). Ten blok  
"mówi" tekst w białym dymku, na przykład "Mogę  
mówić" jak w poniższym przykładzie. Zmień to na  
cokolwiek chcesz i kliknij na blok, aby go usłyszeć.  
Upewnij się, że głośniki są włączone przed  
przetestowaniem.



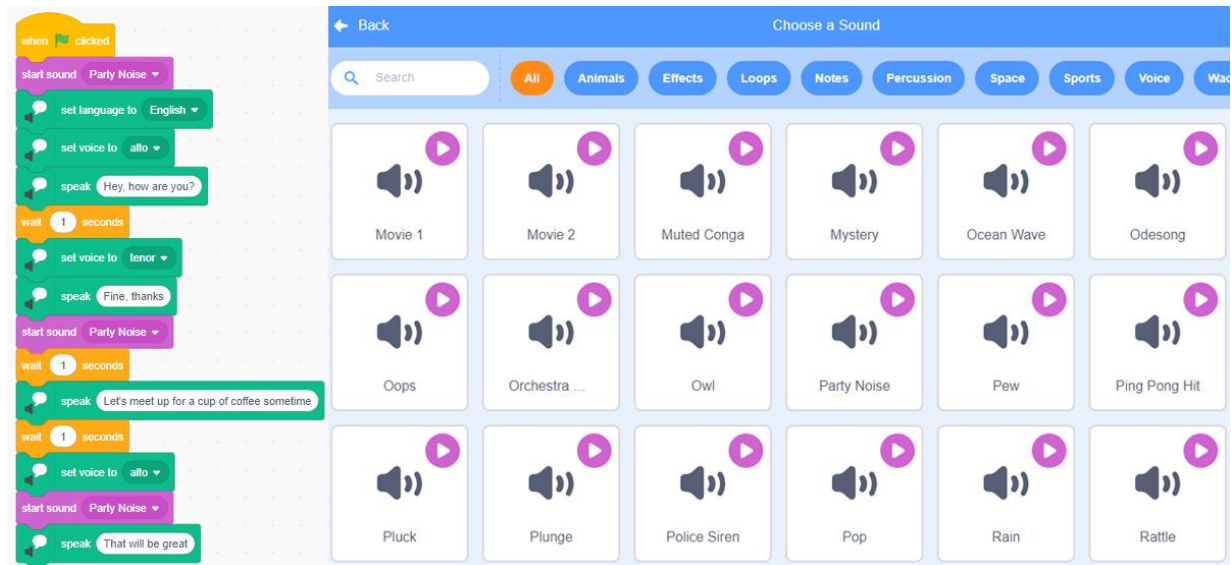
Krok 7: W zasadzie, nie jest trudno sprawić,  
aby postać w Scratchu mówiła. Wszystko, co  
musisz zrobić, to ustawić język, głos i rozpocząć  
mówienie.



Krok 8: Co możesz z tym zrobić?  
Możesz odtworzyć słynne wypowiedzi filmowe, takie jak ta z Gwiezdných Wojen:



Krok 9: Możesz również stworzyć swój własny film, opowiedzieć historię...  
Połącz go z innymi dźwiękowymi efektami ze zbioru dźwięków, aby scena była bardziej  
realistyczna.



Dzisiaj mamy wiele urządzeń generujących mowę. Urządzenia generujące mowę pozwalają ludziom "mówić" słowa i zdania za pomocą elektronicznych środków. Są to przenośne urządzenia elektroniczne, które odtwarzają słowa lub zdania, gdy użytkownik dotknie przełącznika lub naciśnie przyciski lub klawisze. Niektóre urządzenia "mówią" słowa, gdy są one wpisywane na klawiaturze. Urządzenia generujące mowę pozwalają osobom, które nie mogą używać mówionego języka, "mówić" za pomocą elektronicznych środków. Urządzenia generujące mowę są wykorzystywane od lat 90. XX wieku w terapii komunikacyjnej dla dzieci z autyzmem.



## PODSUMOWANIE

Text to Speech (TTS) to rodzaj technologii, która czyta na głos cyfrowy tekst.



## Scenariusz lekcji 11

Czas (minuty):  
90

### Tematyka

---

Interpretacja wyników  
algorytmu obiektu  
sterowanego głosem

### Cele

---

Praktyczne wykorzystanie  
obektu sterowanego  
głosem

### Oczekiwane efekty

---

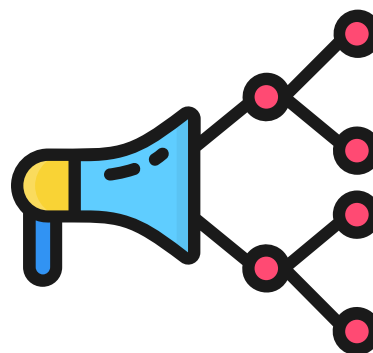
Badanie możliwości  
rozszerzenia zamiany  
mowy na tekst w Scratchu.

## Projekt z obiektem sterowanym głosem

*Zapytaj swoich uczniów, czy obiekt może być sterowany głosem.  
Czy istnieje możliwość skorzystania z obiektów sterowanych głosem w  
jakiś sposób?*

### Przedstawienie celu głównego lekcji:

Zrozumienie algorytmu obiektu sterowanego głosem i jego  
zastosowanie w praktycznym projekcie.





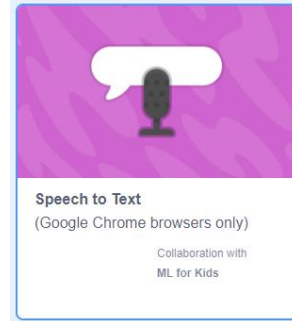
# CZEŚĆ GŁÓWNA

Poprzez projekt, nauczyciel instruuje uczniów w zakresie poleceń i umiejętności Scratch oraz w trenowaniu modelu do konwersji mowy na tekst.

Krok 1: Otwórz przeglądarkę Chrome i przejdź pod adres:  
<https://machinelearningforkids.co.uk/scratch3/>

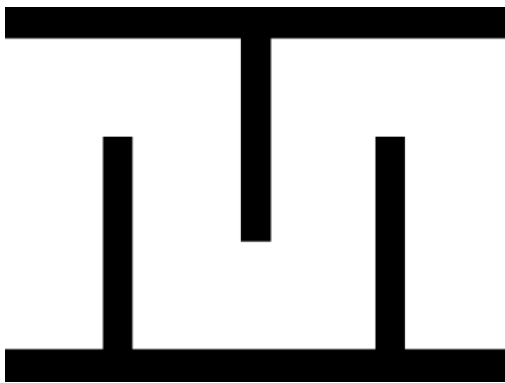
Krok 2: Załaduj rozszerzenie Mowa na tekst  
(STT - tylko dla przeglądarek Google Chrome)

Krok 3: Usuń duszka Cat, klikając na ikonę kosza obok duszka.

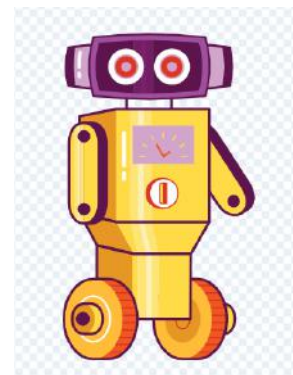


Krok 4: Pobierz plik maze.png z:

<https://drive.google.com/file/d/11YBBhQclhVfHYMWeLkhgAYKSfwv33pT5/view?usp=sharing> i załaduj go do Scratch jako niestandardowego duszka.



Krok 5: Z galerii duszków wybierz Retro Robot i użyj drugiego kostiumu (Retro Robot b).



Krok 6: Z galerii duszków wybierz przycisk Home (Home Button).





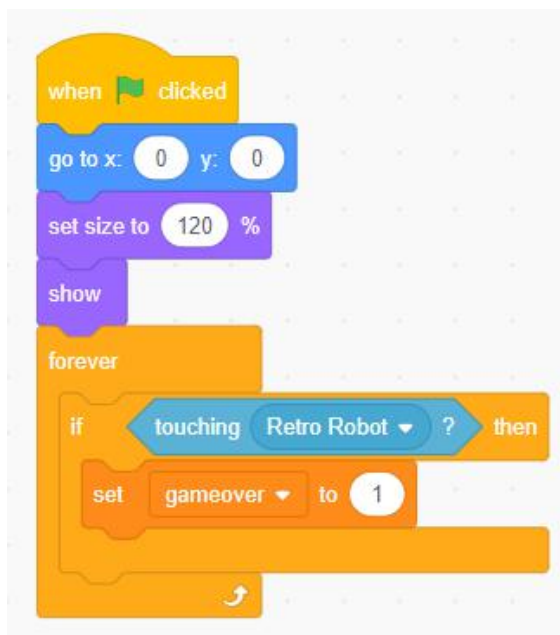
Krok 7: Stwórz trzy zmienne (dla wszystkich duszków):

- **gameover** (pokazuje, jak zakończyła się gra)
- **xm** (pozycja x robota)
- **ym** (pozycja y robota)

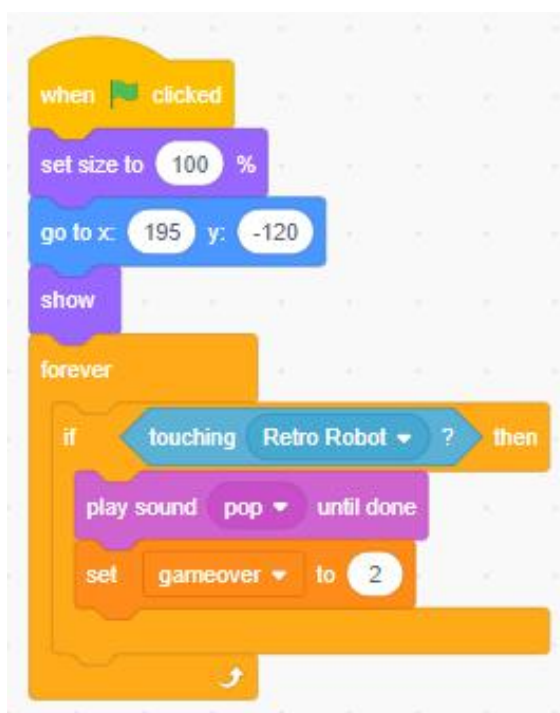
Krok 8: Stwórz listę i zmień jej nazwę na: **endtalk** (wyświetla komunikat zakończenia gry, w zależności od sposobu zakończenia gry)

Krok 9: Przełącz się na duszka maze i rozpocznij kodowanie. Pierwsze kilka bloków ustawia pozycję, rozmiar i widoczność labiryntu. Następnie umieść w pętli "Zawsze" (forever) blok sprawdzający kolizję. W przypadku kolizji, zmienna **gameover** jest ustawiana na 1.

Krok 10: Przełącz się na duszka **Home Button**. Kod jest prawie taki sam, z tą różnicą, że w przypadku kolizji odtwarzany jest dźwięk **pop** a zmienna **game over** jest ustawiana na 2.



```
when green flag clicked
  go to x: 0 y: 0
  set size to 120 %
  show
  forever loop
    if touching Retro Robot ? then
      set gameover to 1
```



```
when green flag clicked
  set size to 100 %
  go to x: 195 y: -120
  show
  forever loop
    if touching Retro Robot ? then
      play sound pop until done
      set gameover to 2
```



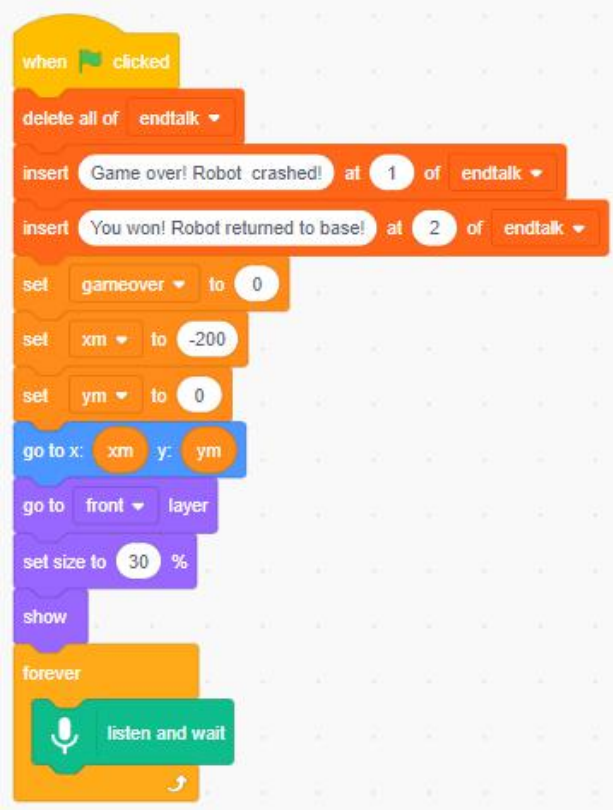


Krok 11: Główny kod jest przypisany do kodu duszka Retro Robot i składa się z 5 wątków, które startują jednocześnie.

11.1. Wątek Flag (start) ustawia zmienne, wpisy na liście (i usuwa wszystkie poprzednie wpisy), pozycję i widoczność. Na końcu znajduje się blok nasłuchu (z rozszerzenia STT) w pętli.

11.2. Kolejne 4 wątki są uruchamiane przez rozpoznawanie mowy, a każda rozpoznana komenda głosowa jest obsługiwana przez sekwencję bloków.

Jeśli nie wykryto kolizji (gameover wynosi 0), robot porusza się w określonym kierunku (poprzez zmianę wartości xm lub ym i przeniesienie robota na obliczoną pozycję). Jeśli wykryto kolizję (gameover wynosi 1 lub 2), wyświetlana jest wiadomość o zakończeniu gry (z listy), a program zostanie zatrzymany.







```
when I hear up
  if (gameover = 0) then
    change ym by 20
    glide 0.25 secs to x: xm y: ym
  else
    start sound computer beeps1
    say item gameover of endtalk for 3 seconds
    stop all

when I hear left
  if (gameover = 0) then
    change xm by -20
    glide 0.25 secs to x: xm y: ym
  else
    start sound computer beeps1
    say item gameover of endtalk for 3 seconds
    stop all

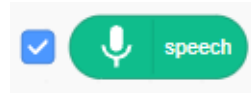
when I hear right
  if (gameover = 0) then
    change xm by 20
    glide 0.25 secs to x: xm y: ym
  else
    start sound computer beeps1
    say item gameover of endtalk for 3 seconds
    stop all

when I hear down
  if (gameover = 0) then
    change ym by -20
    glide 0.25 secs to x: xm y: ym
  else
    start sound computer beeps1
    say item gameover of endtalk for 3 seconds
    stop all
```





Krok 12: Spróbuj użyć komend głosowych w swoim ojczystym języku i zobacz, jak to działa. Omów, w jaki sposób (nie)poprawna pisownia wpływa na rozpoznawanie mowy. Włącz (sprawdź) wyniki rozpoznawania mowy.



Roboty śledzące obiekty, jeśli można nimi inteligentnie sterować za pomocą głosu, mogą być ogromną pomocą dla osób niepełnosprawnych fizycznie. System rozpoznawania mowy jest używany do rozpoznawania zestawu wcześniej zdefiniowanych poleceń, takich jak do przodu, do tyłu, w lewo, w prawo i obrót pod określonym kątem. Robot porusza się zgodnie z sygnałem z komendy głosowej, jednocześnie śledząc pożądaną obiekt. Przetwarzanie sygnału komend głosowych odbywa się w czasie rzeczywistym za pomocą serwera w chmurze, który przekształca je na formę tekstową. Tekstowy sygnał komend jest następnie przesyłany do robota poprzez sieć Bluetooth w celu kontrolowania jego napędu różnicowego. Prototypowy smart robot składa się z trzech podsystemów: systemu rozpoznawania mowy, systemu śledzenia obiektów i systemu sterowania ruchem opartego na napędzie różnicowym. Dokładność i wydajność systemu rozpoznawania mowy są badane za pomocą zestawu eksperymentów. Przebadane są także wpływające na nie czynniki, takie jak hałas i odległość, z obiecującymi wynikami. Prototypowy robot jest w stanie rozpoznawać polecenia głosowe w zasięgu Bluetooth, czyli na odległość 10 m. Omawiane są również możliwe rozbudowy, które mogą prowadzić do szerokiego zakresu dalszych zastosowań.

## PODSUMOWANIE

Możemy kontrolować obiekty w programie komputerowym za pomocą naszego głosu.





## Scenariusz lekcji 12

Czas (minuty):  
90

### Tematyka

Wprowadzenie do sprzętu  
- mikrokontroler, kamera i  
sterownik silnika

### Cele

Uczniowie poznają sprzęt,  
który będzie używany w  
naszym głównym projekcie

### Oczekiwane efekty

Zapoznanie się ze  
sprzętem używanym w  
głównym projekcie

## Wprowadzenie do sprzętu - mikrokontroler, kamera i sterownik silnika

Część tego programu nauczania dotyczy rzeczywistych, fizycznych urządzeń z możliwościami sztucznej inteligencji. W tym celu Chorwackie Stowarzyszenie Robotyki opracowało małego mobilnego robota do zastosowania w praktycznym programowaniu omawianym na poprzednich lekcjach. Na początku zastanówmy się, z czego powinien składać się nasz robot. Oczywiście jest, że mechaniczne części - obudowa, koła i silniki prądu stałego - będą wykorzystane jako napęd. A co z częściami elektronicznymi? Podobnie jak większość żywych istot, nasz robot będzie potrzebował mózgu (mikrokontrolera), oczu (kamery AI) oraz nerwów/impulsów (sterownika silników prądu stałego). Przyjrzyjmy się tym częściom.

Istnieją 2 opcje budowy robota:

Zestaw robota Maqueen Plus + kamera AI HuskyLens (łatwy - odpowiedni dla początkujących)

Micro:Maqueen Plus to zaawansowany edukacyjny robot STEM dla Micro:bita. Jest mocny i inteligentny, wyposażony w zoptymalizowane zarządzanie energią i wysokowydajne źródło zasilania: w pełni kompatybilny z czujnikiem wizyjnym AI HuskyLens, co czyni go dostępnym narzędziem dydaktycznym.

Micro:Maqueen Plus posiada dużą i stabilną obudowę, wbudowane funkcje oraz wiele portów rozszerzeń. Nadaje się nie tylko do nauczania w klasie, ale także do rozszerzonych zajęć pozalekcyjnych i zawodów robotycznych.

Arduino UNO + Sterownik silnika + kamera AI HuskyLens (bardziej zaawansowane - tylko dla doświadczonych użytkowników)

# CZĘŚĆ GŁÓWNA



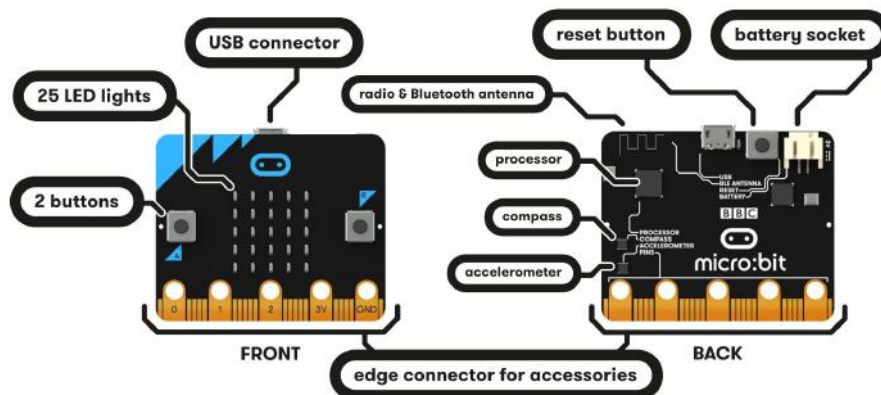
ARTIE



## Mózg 1 - micro:bit

W elektronicznym świecie "mózg" nazywany jest procesorem lub, w tym konkretnym przypadku, mikrokontrolerem.

Micro:bit to łatwy w użyciu, potężny i ekonomiczny mikrokontroler o kieszonkowych rozmiarach, zaprojektowany do nauki programowania dla dzieci i początkujących. Pozwala im łatwo realizować swoje pomysły w postaci własnych gier cyfrowych, interaktywnych projektów i robotyki DIY.

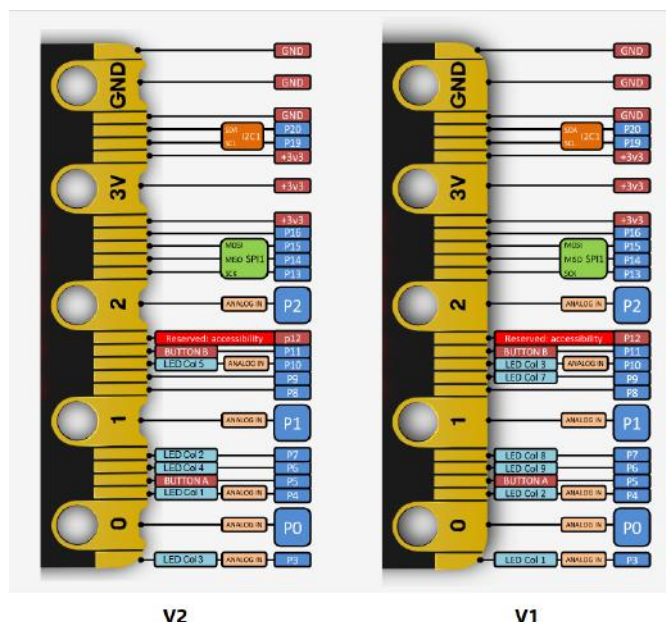


Dzięki swoim portom wejścia/wyjścia i wsparciu sprzętowemu, Micro:bit doskonale nadaje się do różnorodnych zadań związanych z nauką i rozwojem robotyki.

## Układ i specyfikacje techniczne Micro:bita

Na rynku dostępne są dwie wersje Micro:bita - sprawdź ten artykuł, aby dowiedzieć się, która wersja jest w Twoim posiadaniu:

<https://kitronik.co.uk/blogs/resources/explore-micro-bit-v1-microbit-v2-differences>



<http://erasmus-artie.eu>



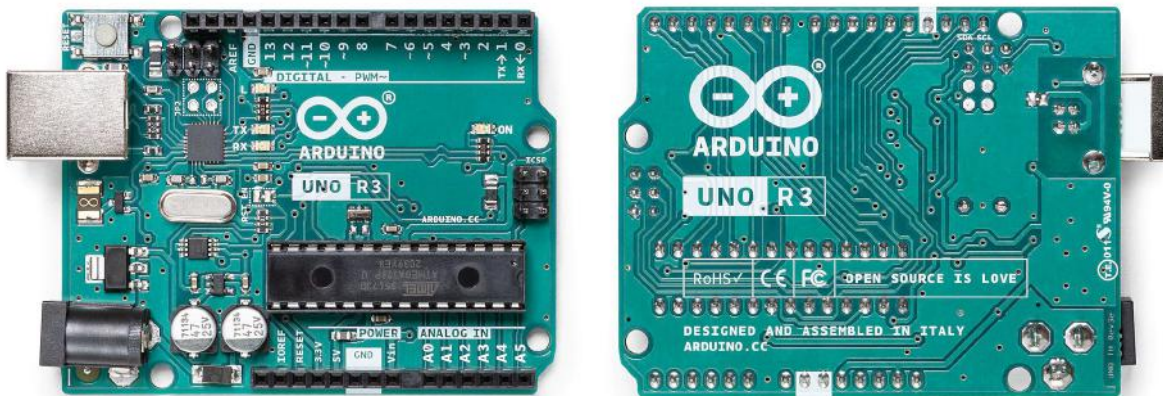
- Niewielka płytka o podobnym rozmiarze do karty kredytowej (4 cm x 5 cm)
- Zintegrowane moduły, takie jak akcelerometr, kompas i moduł Bluetooth® Smart
- Kieszonkowy mikrokontroler
- Macierz LED 5x5 (obsługuje również wykrywanie światła)
- Czujniki światła, temperatury i inne powszechne czujniki

Wyposażony w procesor ARM M0, micro:bit może wykonywać większość podstawowych funkcji robota.

Wprowadzenie do micro:bita [https://www.youtube.com/watch?v=POkel\\_2NXMo](https://www.youtube.com/watch?v=POkel_2NXMo)

## Mózg 2 - Arduino UNO

Arduino UNO to najpopularniejszy mikrokontroler na świecie, posiadający ogromną społeczność online i wiele dostępnych projektów w Internecie. Na rynku dostępnych jest wiele klonów Arduino UNO. Klon oznacza, że architektura rdzenia elektronicznego jest podobna do Arduino UNO, ale wprowadzono pewne modyfikacje, aby dodać dodatkowe funkcje do płytki. Te modyfikacje zostały zaprojektowane i opracowane specjalnie dla uczniów, aby uczyć ich programowania i architektury mikrokontrolera. Przyjrzyjmy się teraz Arduino UNO.



Każdy z 14 cyfrowych pinów na płytce Uno może być używany jako wejście lub wyjście i działa przy napięciu 5 woltów. Każdy pin może dostarczać lub odbierać prąd o wartości 20 mA jako zalecane warunki pracy i ma wewnętrzny rezystor pull-up (domyślnie rozłączony) o wartości 20-50 kΩ. Maksymalna wartość 40 mA nie powinna być przekroczona na żadnym pinie wejścia/wyjścia, aby uniknąć trwałego uszkodzenia mikrokontrolera.

Dodatkowo, niektóre piny mają specjalne funkcje:

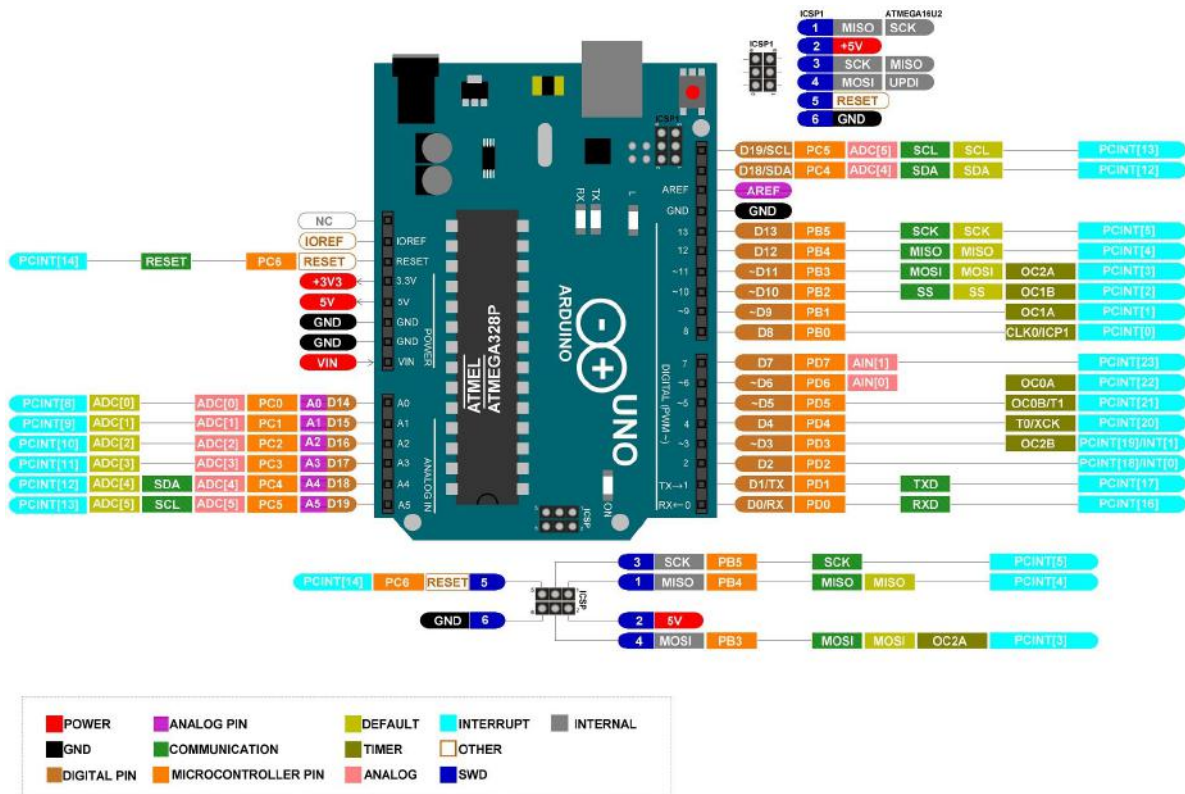
- Serial: 0 (RX) i 1 (TX). Służą do odbierania (RX) i przesyłania (TX) danych szeregowych TTL. Te piny są połączone z odpowiadającymi pinami układu ATmega8U2 USB-to-TTL Serial.
- Przerwania zewnętrzne: 2 i 3. Te piny można skonfigurować tak, aby wywoływały przerwanie przy niskiej wartości, narastającym lub opadającym zboczach lub zmianie wartości.
- PWM: 3, 5, 6, 9, 10 i 11. Zapewniają funkcję 8-bitowego wyjścia PWM.



- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Te piny obsługują komunikację SPI za pomocą biblioteki SPI.
  - LED: 13. Wbudowana dioda LED sterowana jest przez cyfrowy pin 13. Gdy pin ma wartość HIGH, dioda jest włączona, gdy pin ma wartość LOW, dioda jest wyłączona.
  - TWI: Pin A4 lub SDA i pin A5 lub SCL. Obsługują komunikację TWI za pomocą biblioteki Wire.
- Uno posiada 6 wejść analogowych oznaczonych jako A0 do A5, z których każde dostarcza 10-bitową rozdzielczość (czyli 1024 różne wartości). Domyślnie mierzą od zera do 5 woltów, choć możliwe jest zmienienie górnego zakresu za pomocą pinu AREF. Na płytce znajduje się jeszcze kilka innych pinów:
- REF. Napięcie referencyjne dla wejść analogowych.
  - Reset. Niski stan na tej linii powoduje zresetowanie mikrokontrolera. Zwykle używany do dodania przycisku resetującego na shieldach, które blokują ten na płytce.

### Układ płytki Arduino UNO oraz specyfikacje techniczne

Oto wyjaśnienie pinów na płytce:





MICROCONTROLLER	ATmega328P
OPERATING VOLTAGE	5V
INPUT VOLTAGE (RECOMMENDED)	7-12V
INPUT VOLTAGE (LIMIT)	6-20V
DIGITAL I/O PINS	14 (of which 6 provide PWM output)
PWM DIGITAL I/O PINS	6
ANALOG INPUT PINS	6
DC CURRENT PER I/O PIN	20 mA
DC CURRENT FOR 3.3V PIN	50 mA
FLASH MEMORY	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
CLOCK SPEED	16 MHz
LED_BUILTIN	13
LENGTH	68.6 mm
WIDTH	53.4 mm
-----	--

Główne zasilanie dla Arduino UNO odbywa się przez połączenie USB, ale możliwe jest również zasilanie z baterii podłączonych za pomocą pinów złącza (VIN i GND).

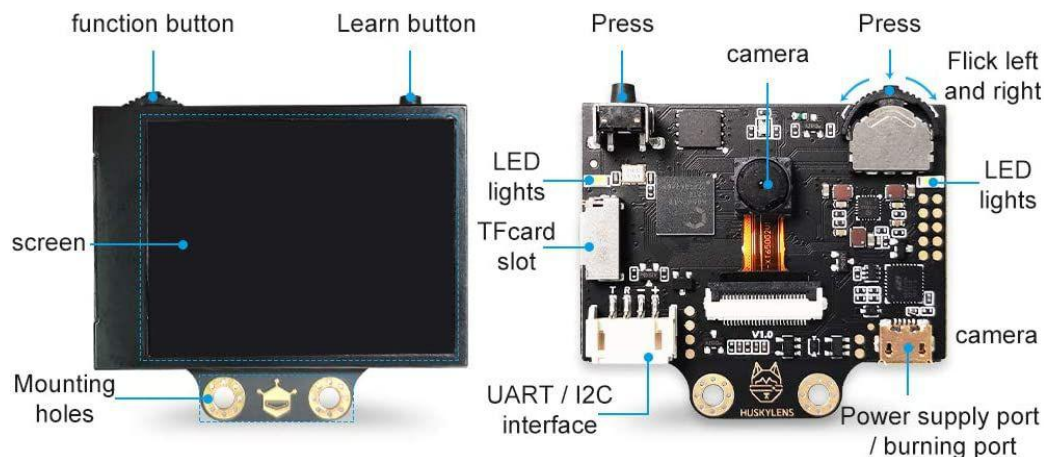
Wprowadzenie do Arduino UNO :<https://www.youtube.com/watch?v=bniUECtJkeU>

### Oczy - HuskyLens Kendryte K210 (we wszystkich scenariuszach)

Nasz mobilny robot będzie w stanie postrzegać otoczenie za pomocą tej kamery. HuskyLens to łatwy w użyciu czujnik sztucznej inteligencji i komputerowego widzenia, który posiada 7 wbudowanych funkcji: rozpoznawanie twarzy, śledzenie obiektów, rozpoznawanie obiektów, śledzenie linii, rozpoznawanie kolorów, rozpoznawanie etykiet i klasyfikacja obiektów.

Może być łatwo podłączony do dowolnego urządzenia Arduino lub Arduino-kompatybilnego oraz do micro:bita. Teraz możesz tworzyć bardzo kreatywne projekty nawet bez znajomości zaawansowanych algorytmów uczenia maszynowego.

Spójrzmy teraz na kamerę.



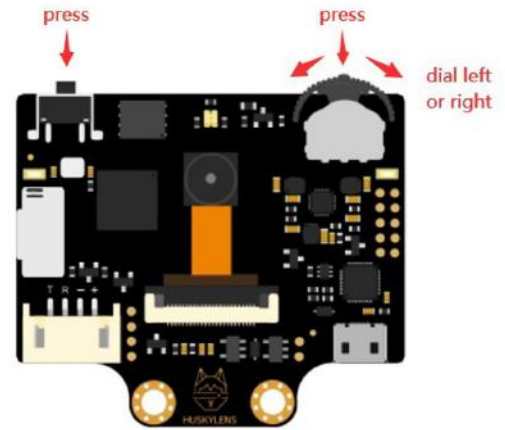


## Przyciski

Na HuskyLens znajdują się dwa przyciski: przycisk funkcji (function button) i przycisk nauki (learning button).

Podstawowe operacje tych dwóch przycisków są przedstawione poniżej:

- Przekręć przycisk "funkcji" w lewo lub prawo, aby przełączać między różnymi funkcjami.
- Krótkie naciśnięcie przycisku "nauki" służy do nauki określonego obiektu. Długie naciśnięcie przycisku "nauki" pozwala na ciągłe uczenie określonego obiektu z różnych kątów i odległości. Jeśli HuskyLens już nauczył się danego obiektu, krótkie naciśnięcie przycisku "nauki" spowoduje jego zapomnienie.
- Długie naciśnięcie przycisku "funkcji" powoduje wejście do drugiego poziomu menu (ustawienia parametrów) w bieżącej funkcji. Przekręć w lewo, prawo lub krótko naciśnij przycisk "funkcji", aby ustawić odpowiednie parametry.



## System współrzędnych

Kiedy HuskyLens wykryje obiekt, cel zostanie automatycznie zaznaczony na ekranie za pomocą kolorowej ramki.

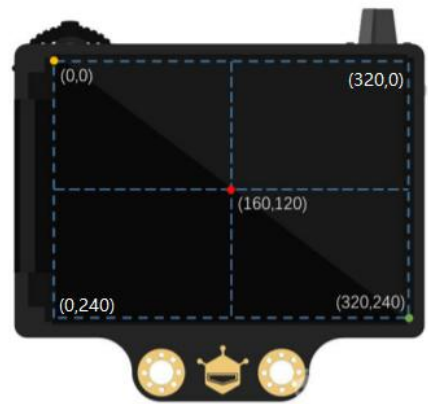
Współrzędne pozycji ramki koloru x i y są przypisane zgodnie z następującym systemem współrzędnych. Po otrzymaniu współrzędnych z portu UART / I2C, można poznać pozycję obiektu. Format: (x, y)

Funkcje:

- 1) Rozpoznawanie twarzy
- 2) Śledzenie obiektów
- 3) Rozpoznawanie obiektów
- 4) Śledzenie linii
- 5) Rozpoznawanie kolorów
- 6) Rozpoznawanie etykiet
- 7) Klasyfikacja obiektów

## Instrukcje kolorów

W każdej funkcji definicje kolorów ramki oraz symbolu "+" na środku ekranu są takie same, co pomaga w określeniu aktualnego statusu HuskyLens..



Colour	Status
From orange to yellow, then from yellow to orange	Have not learned the object yet but ready to learn
Yellow	Learning the new object
Blue	Have learned the object and recognized it





Wskaźnik LED RGB służy do wskazywania statusu funkcji rozpoznawania twarzy. Jego kolory są zdefiniowane w następujący sposób.

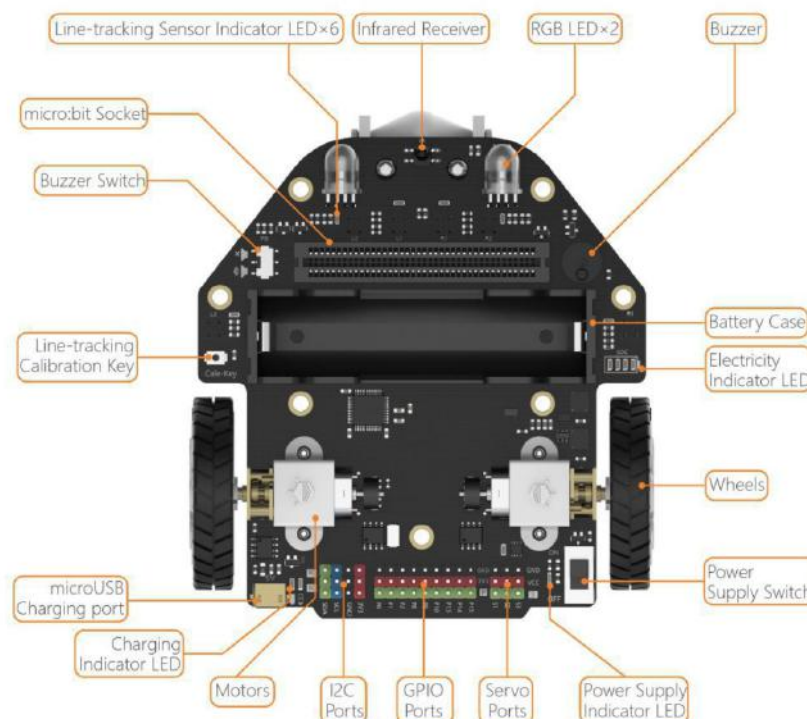
Colour	Status
Blue	Have not learned the face yet, but detected the face
Yellow	Learning the new face
Green	Have learned the face and recognized it

Kompletna instrukcja: [https://wiki.dfrobot.com/HUSKYLENS\\_V1.0\\_SKU\\_SEN0305\\_SEN0336](https://wiki.dfrobot.com/HUSKYLENS_V1.0_SKU_SEN0305_SEN0336)

## Nerwy/impulsy/mięśnie 1 - Płyta robota Maqueen Plus

Maqueen Plus to inteligentny programowalny robot edukacyjny zaprojektowany dla początkujących. Może być programowany za pomocą platform Mind+ i MakeCode. Został zoptymalizowany pod względem zarządzania energią i wyposażony w większe źródło zasilania. Jest idealny do użytku z czujnikiem wizyjnym HuskyLens AI i posiada większą i bardziej stabilną obudowę. Oferuje także więcej wbudowanych funkcji oraz większą ilość portów rozszerzeń. Nadaje się do nauczania w klasach, a także do zajęć pozalekcyjnych i konkursów robotycznych.

Oto główne części na płycie Maqueen Plus:





## Specyfikacje techniczne Maqueen Plus:

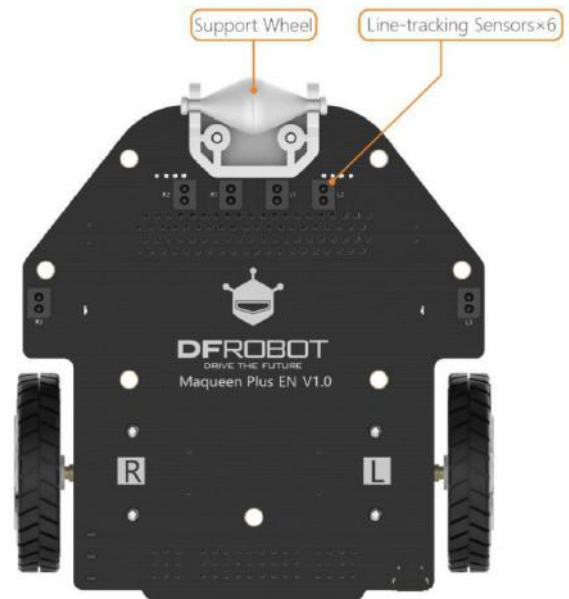
- Zasilanie: bateria litowa 3,7V-18650
- Napięcie ładowania: 5V
- Prąd ładowania: 900mA, czas ładowania: 4h
- Wskaźnik baterii: 4 diody LED
- Silnik napędowy: silnik N20, 260 obr./min
- Buzzer \* 1
- RGB-LED \* 2
- Porty rozbudowy GPIO: P0, P1, P2, P8, P12, P13, P14, P15, P16
- Porty rozbudowy I2C \* 3
- Porty rozbudowy serwomechanizmów \* 3
- Czujniki śledzenia linii \* 6
- Dane wyjściowe czujników śledzenia linii: analogowe + cyfrowe
- Kalibracja czujników linii: obsługa
- Czujnik odbiornika podczerwieni \* 1
- Czujnik ultradźwiękowy: URM10
- Metalowa płyta górna \* 1
- Połączenia gwintowane M3 \* 12
- Wielkość mapy: 50cm x 50cm
- Wymiary: 107 x 100mm (4.21 x 3.94")

Obecnie Maqueen Plus jest dostępny poprzez platformę programistyczną Mind+. Mind+ to platforma programowania graficznego oparta na Scratch 3.0, stworzona przez DFRobot, która obsługuje język Python, Arduino i inne platformy programistyczne. Obecnie Mind+ został zastosowany do różnych czujników, modułów i powiązanych produktów edukacyjnych.

Dowiedz się więcej o Maqueen Plus:

[https://github.com/DFRobot/Maqueen\\_Plus\\_Basic\\_Tutorial/blob/master/MBT0021-EN-Maqueen%20Plus%20Basic%20Tutorial.pdf](https://github.com/DFRobot/Maqueen_Plus_Basic_Tutorial/blob/master/MBT0021-EN-Maqueen%20Plus%20Basic%20Tutorial.pdf)

W trakcie realizacji tego scenariusza na rynku dostępna jest wersja V2 tego robota.



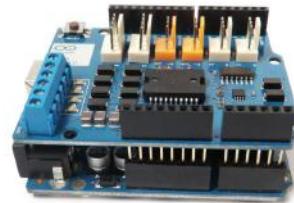
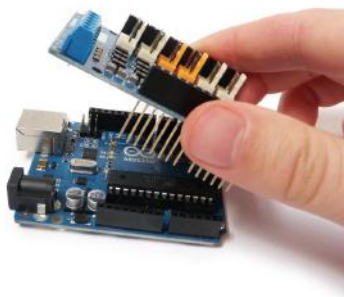
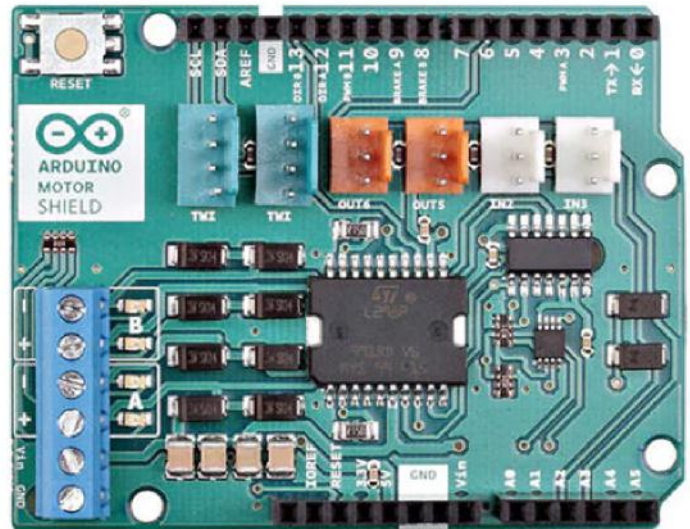


## Nerwy/impulsy/mięśnie 2 - Arduino Motor Shield Rev 3

Istnieje kilka sposobów sterowania silnikiem prądu stałego; najprostszym z nich jest zastosowanie zasilania. Wczesne wynalazki wykorzystujące silnik prądu stałego działały w ten sposób: podłącz źródło zasilania, a silnik zacznie się obracać; zmień polaryzację, a zmienisz kierunek obrotów.

Jednak jeśli chcemy zrobić coś więcej niż tylko obracać silnikiem w pełnej prędkości w dwóch kierunkach, potrzebujemy układu sterującego silnikiem. Konkretnie chodzi o układ dwukanałowego sterownika mostka H L298P (układ scalony), który znajduje się na Motor Shield Rev3.

Motor Shield jest stosowalny (ang. stackable) do Arduino UNO, co oznacza, że nie trzeba prowadzić połączeń przewodami do mikrokontrolera. Wystarczy podłączyć wtyczki (male) płytki Motor Shield do gniazd (female) mikrokontrolera.



Płytki Motor Shield ma 2 kanały, co umożliwia kontrolę dwóch silników prądu stałego lub jednego silnika krokowego.

Posiada również 6 gniazd do podłączenia wejść, wyjść i linii komunikacyjnych Tinkerkit. Użycie tych pinów jest nieco ograniczone i nie jest objęte tym scenariuszem.

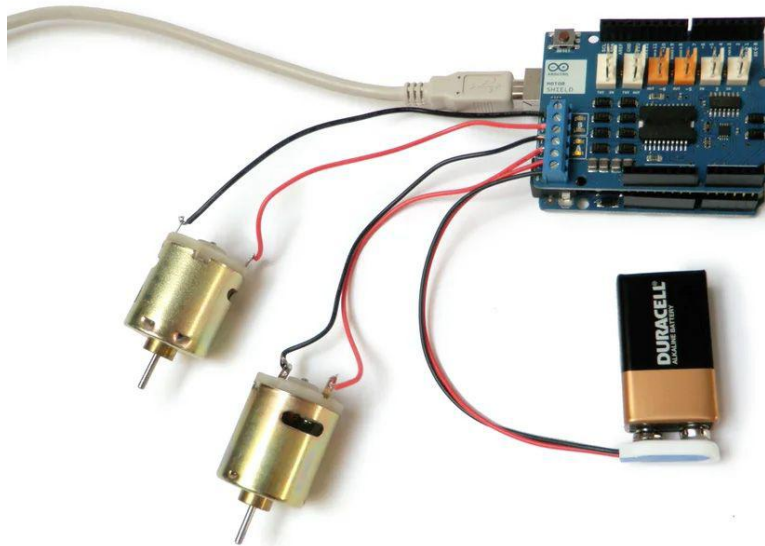
Z zewnętrznym zasilaniem płytki Motor Shield może bezpiecznie dostarczać napięcie do 12V i 2A na kanał silnika (lub 4A dla pojedynczego kanału).

Na Arduino są piny, które zawsze są używane przez tę płytkę. Poprzez manipulowanie tymi pinami możemy wybrać kanał silnika do inicjalizacji, ustawić kierunek obrotów silnika (polarność), regulować prędkość silnika (sygnał PWM), zatrzymać i uruchomić silnik oraz monitorować pobór prądu każdego kanału.



Function	Channel A	Channel B
Direction	Digital 12	Digital 13
Speed (PWM)	Digital 3	Digital 11
Brake	Digital 9	Digital 8
Current Sensing	Analog 0	Analog 1

Po podłączeniu powinno to wyglądać następująco:



<http://erasmus-artie.eu>

Dowiedz się więcej na stronie Instructables:

<https://www.instructables.com/Arduino-Motor-Shield-Tutorial/>

## PODSUMOWANIE

ARTIEBot, podobnie jak większość istot żywych, posiada mózg (mikrokontroler), oczy (kamerę sztucznej inteligencji) oraz nerwy/impulsy (sterownik silników prądu stałego).



## Scenariusz lekcji 13

Czas (minuty):  
90

### Tematyka

Podłączanie mikrokontrolera do kamery i komputera i rozpoczęcie pracy

### Cele

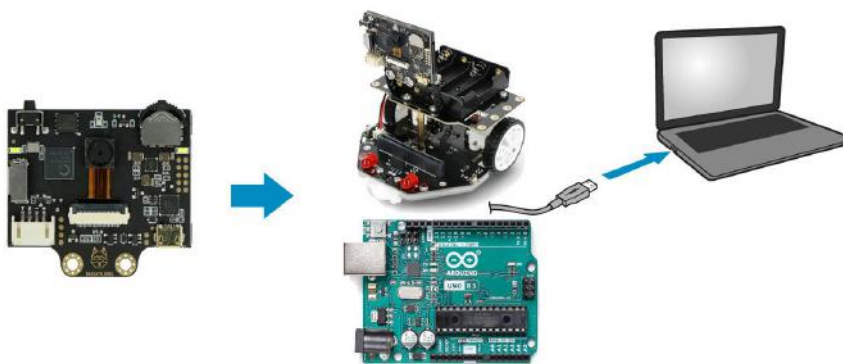
Uczniowie podłączają mikrokontroler do kamery i komputera i obserwują działanie urządzenia

### Oczekiwane efekty

Zrozumienie, jak działa kamera

### Podłączanie mikrokontrolera do kamery i komputera i rozpoczęcie pracy

Zanim przystąpimy do programowania, musimy podłączyć kamerę do mikrokontrolera (micro:bit lub Arduino UNO), a mikrokontroler do komputera.



Opcja 1: Aparat > I2Connection > micro:bit/Maqueen plus > Kabel USB > Laptop lub PC

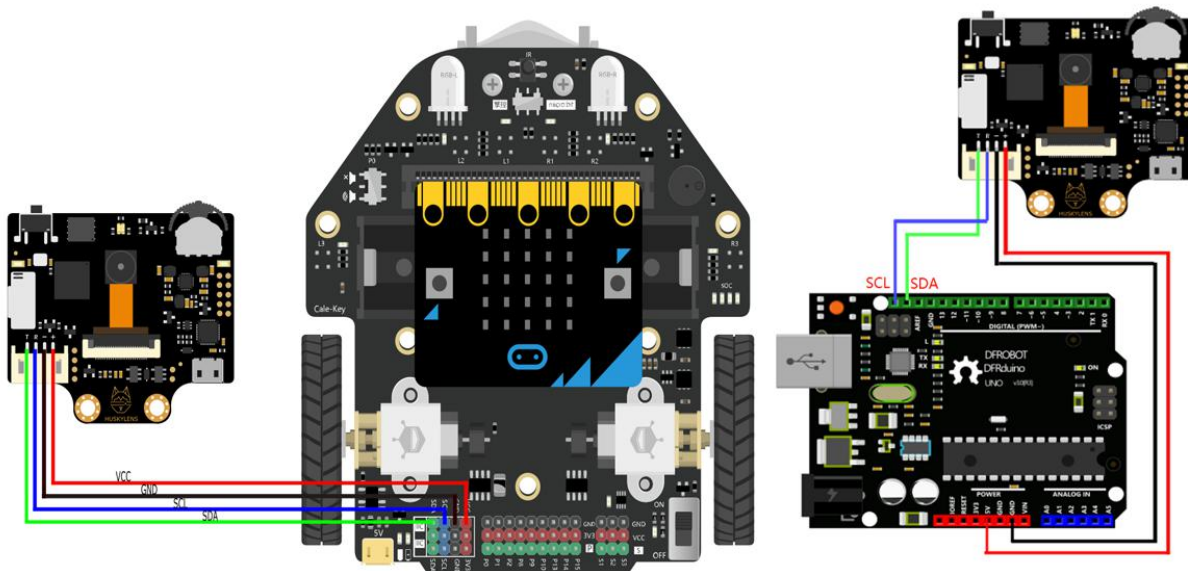
Opcja 2: Kamera > I2Connection > Arduino UNO > Kabel USB > Laptop lub PC



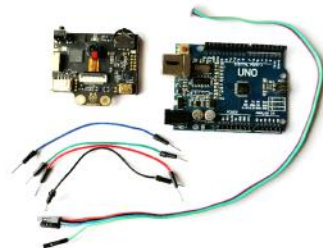
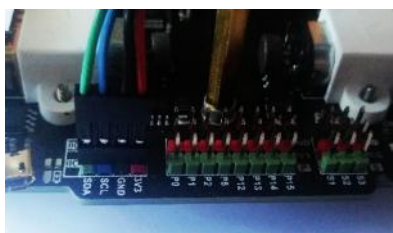
# CZĘŚĆ GŁÓWNA

Nie ma wiele nowego do powiedzenia na temat połączenia USB, ale prawdopodobnie nie słyszeliście jeszcze o I2C.

Magistrala Inter-Integrated Circuit (I2C) to dwuprzewodowy interfejs szeregowy, pierwotnie opracowany przez firmę Phillips do użytku w produktach konsumenckich. Jest hierarchią typu master/slave, gdzie master to urządzenie sterujące zegarem magistrali, adresujące urządzenia slave oraz zapisujące lub odczytujące dane z rejestrów urządzeń slave. Slaves to urządzenia, które odpowiadają tylko na zapytania mastera, poprzez ich unikalny adres. Magistrala I2C wykorzystuje tylko dwie dwukierunkowe linie: linię danych szeregowych (SDA) i linię zegara szeregowego (SCL).

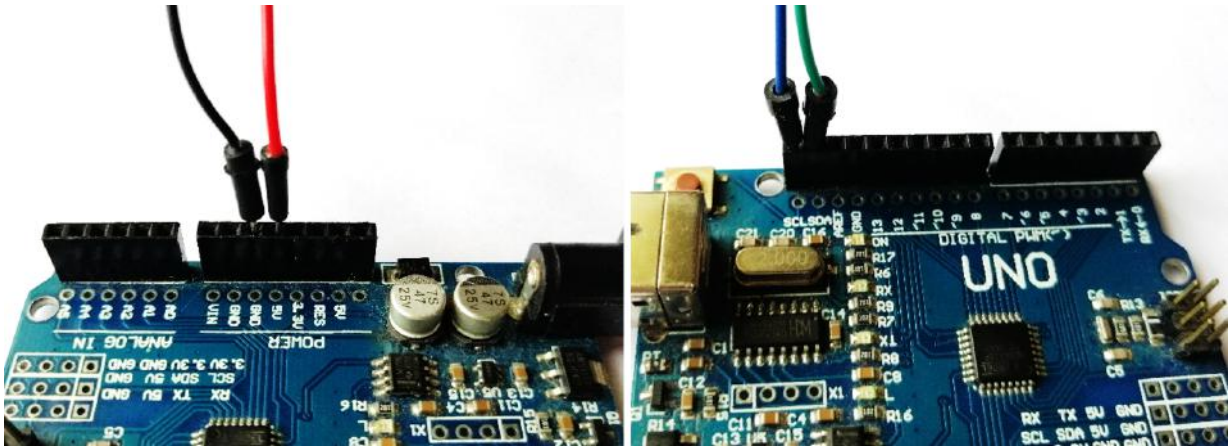


W Opcji 1 - rozwiązanie jest bardzo proste i polega na dopasowaniu kolorów na przewodach i złączu.



W Opcji 2 może być trochę trudniej połączyć kabel kamery z Arduino UNO, ponieważ oba złącza są typu żeńskiego (female). Będziemy potrzebować 4 przewodów połączeniowych typu male to male (męsko-męskie), które można znaleźć w niemal każdym sklepie elektronicznym sprzedającym części do projektów DIY.

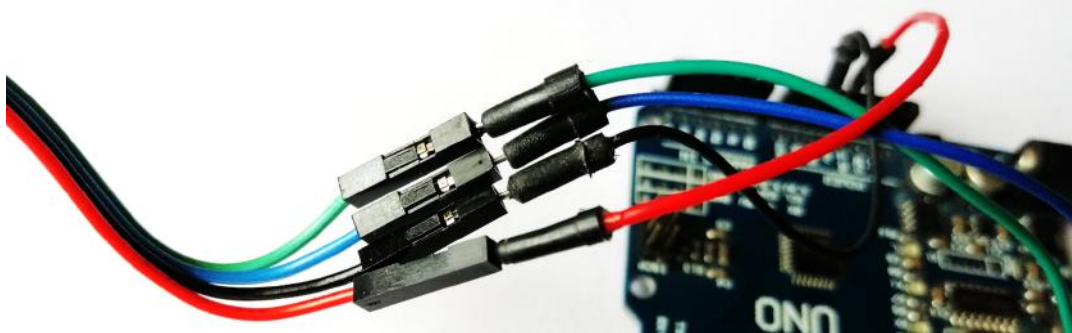




Wybierz 4 przewody połączeniowe (czerwony, czarny, zielony i niebieski), aby dopasować je do odpowiednich pinów kamery.

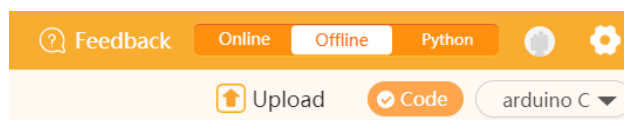
Najpierw podłącz czerwony przewód do pinu 5V na Arduino UNO, czarny przewód do pinu GND, niebieski przewód do pinu SCL, a zielony przewód do pinu SDA, jak pokazano na obrazkach.

Jedynie, co pozostało do zrobienia, to podłączyć drugą stronę tych przewodów do złącza kamery. Dopasuj kolory przewodów do przewodów złącza kamery (czarny do czarnego, czerwony do czerwonego, itd...).



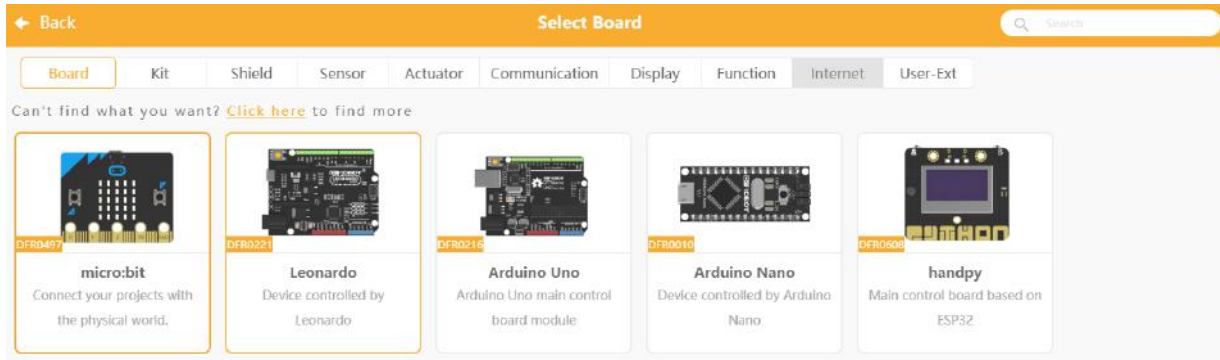
Teraz masz kamerę podłączoną do mikrokontrolera. Podłącz kabel USB do robota Maqueen Plus lub płytki Arduino UNO i połącz go z laptopem lub komputerem.

Przejdź do: <http://mindplus.cc/download-en.html> i pobierz odpowiednią wersję programu Mind+ dla systemu operacyjnego Twojego komputera. Zainstaluj program i uruchom go. Na początek przełącz się w tryb offline



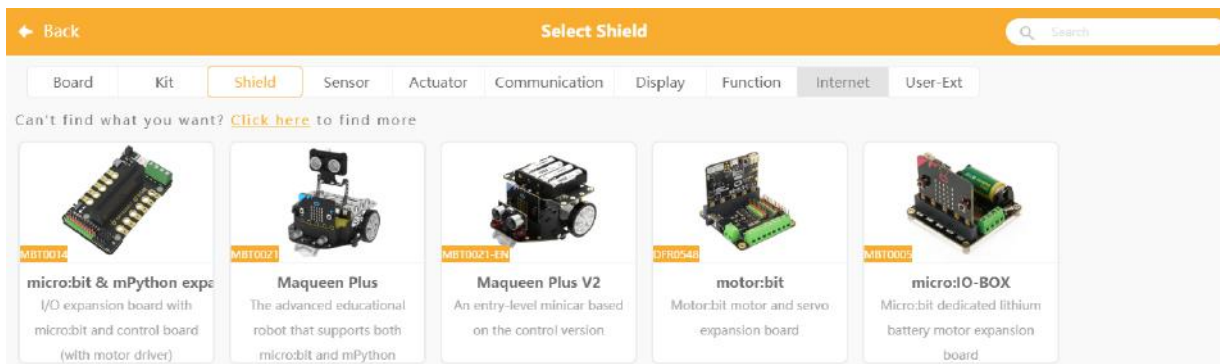


Otwórz zakładkę "Extensions" (Rozszerzenia) i wybierz płytkę (Board):  
Opcja 1) micro:bit (jeśli pracujesz z Maqueen Plus) Opcja 2) Arduino UNO



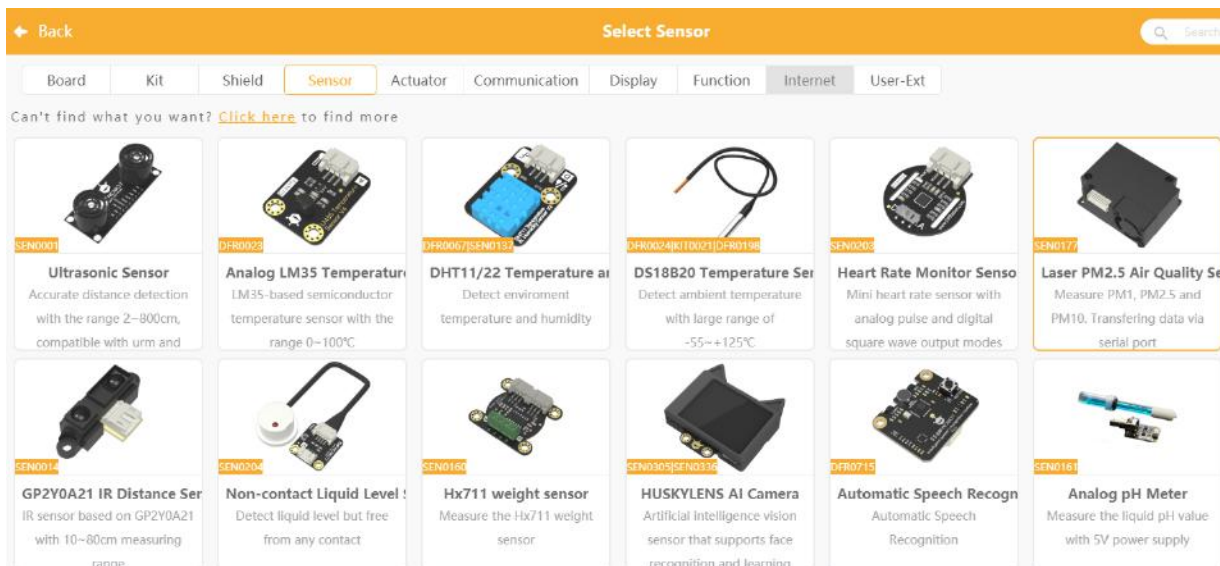
Tylko dla Opcji 1)

Przejdź do zakładki "Shield" (Płytki rozszerzeń) i wybierz Maqueen Plus lub Maqueen Plus V2 (w zależności od Twojej wersji).



Dla obu Opcji 1) i 2)

Przejdź do zakładki "Sensor" (Czujnik) i wybierz czujnik - HUSKYLENS AI Camera.

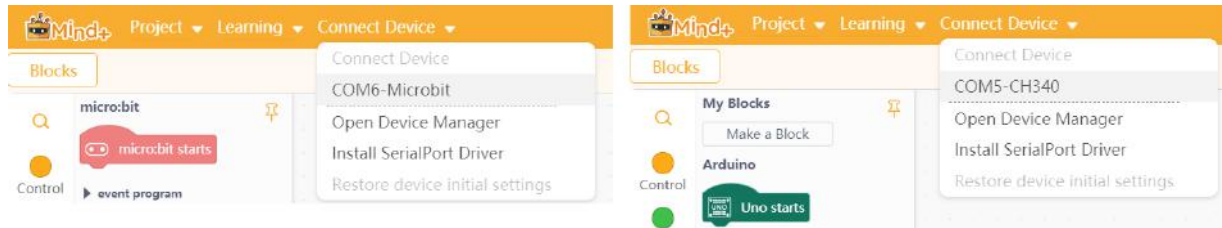






Po wyborze kliknij na **<- Wstecz** i jesteś gotowy do użycia wybranej płytki/roboty i kamery. Przejdźmy do testu, aby sprawdzić, czy wszystko działa.

Przed tym musisz podłączyć urządzenie. Kliknij na Przycisk "Podłącz urządzenie" (Connect Device) i wybierz swój port i urządzenie.

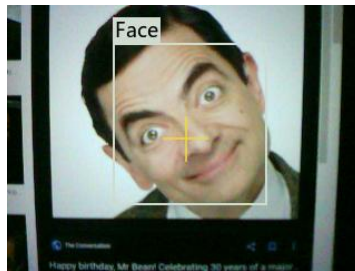


Weź kamerę i przekręć przycisk funkcji w lewo, aż na górze ekranu pojawi się napis "Rozpoznawanie twarzy".



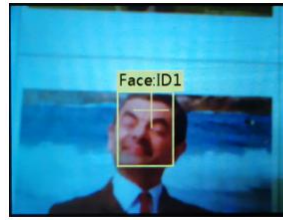
## Nauka i Wykrywanie

1. Wykrywanie Twarzy: Skieruj kamerę HuskyLens na dowolne twarze. Po wykryciu twarzy, zostanie ona automatycznie zaznaczona białą ramką z napisem "Twarz" na ekranie.



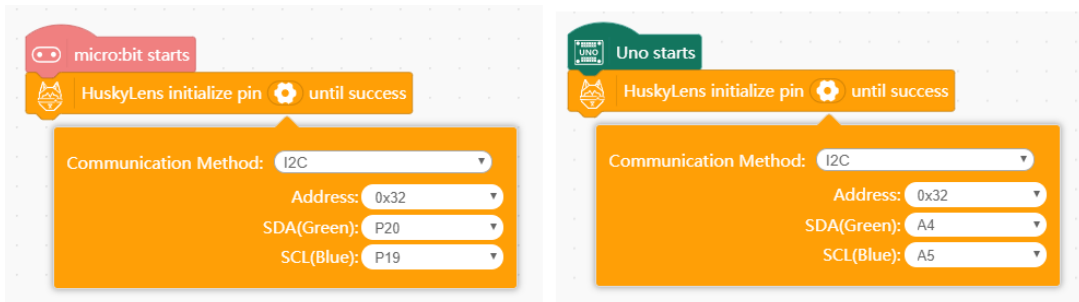
Wskazówki: Jeśli chcesz, aby HuskyLens nauczył się lub rozpoznał Twoją twarz, czyli zrobił selfie, nie będziesz w stanie widzieć ekranu w tym czasie. Możesz jednak określić status na podstawie różnych kolorów wskaźnika RGB.

2. Nauka Twarzy: Skieruj symbol "+" na twarz, wykonaj krótkie naciśnięcie "przycisku nauki" (learning button), aby nauczyć kamerę danej twarzy. Jeśli ta sama twarz zostanie wykryta przez HuskyLens, na ekranie pojawi się niebieska ramka z napisem "Twarz: ID0", co oznacza, że HuskyLens nauczył się twarzy i teraz potrafi ją rozpoznawać.



Odwiedź: [https://wiki.dfrobot.com/HUSKYLENS\\_V1.0\\_SKU\\_SEN0305\\_SEN0336#target\\_15](https://wiki.dfrobot.com/HUSKYLENS_V1.0_SKU_SEN0305_SEN0336#target_15) aby uzyskać pełne informacje dotyczące rozpoznawania twarzy.

Przejdź do Mind+ i rozpocznij programowanie. Po uruchomieniu mikro:bit/Uno użyj bloku "HuskyLens initialize pin until success" i skonfiguruj go dla interfejsu I2C, jak pokazano na obrazku poniżej dla każdej opcji.



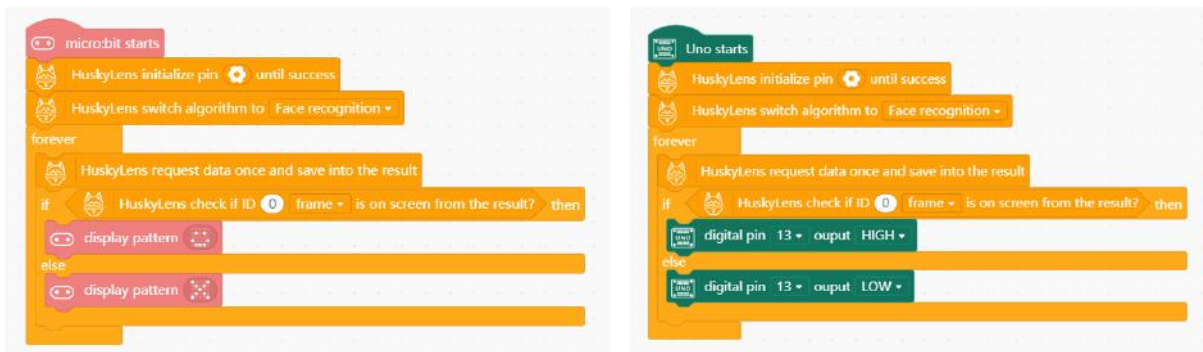
Następnie użyj bloku "HuskyLens switch algorithm to Face recognition" dla obu opcji.



Następnie użyj bloku pętli "forever loop", który sprawdza, czy twarz na kamerze jest rozpoznawana jako Twarz ID0.

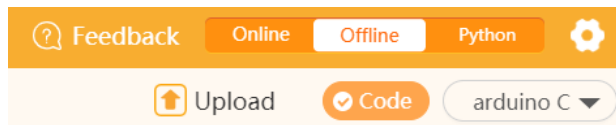
Jeśli twarz zostanie wykryta w Opcji 1), zobaczysz twarz na wyświetlaczu mikro:bit. Jeśli nie, zobaczysz znak X.

Jeśli twarz zostanie wykryta w Opcji 2), zostanie włączona wbudowana dioda LED (D13). Jeśli nie, dioda LED zostanie wyłączona.





Naciśnij przycisk **"Upload"** (Prześlij), aby przesłać ten kod do mikro:bita lub Arduino UNO.



Po przesłaniu kodu, skieruj kamerę na "nauczoną" twarz, a dioda LED na D13 powinna się zapalić. Jeśli oddalisz kamerę od twarzy, dioda na D13 zostanie wyłączona.

Jeśli wszystko działa zgodnie z opisem - wszystko jest w porządku i gotowe do użycia w naszym mobilnym robocie ze sztuczną inteligencją.

## PODSUMOWANIE

HuskyLens to łatwy w użyciu czujnik sztucznej inteligencji do komputerowego widzenia maszynowego, który posiada 7 wbudowanych funkcji: rozpoznawanie twarzy, śledzenie obiektów, rozpoznawanie obiektów, śledzenie linii, rozpoznawanie kolorów, rozpoznawanie tagów i klasyfikacja obiektów. Poprzez port UART / I2C, HuskyLens może być podłączony do Arduino i mikro:bit, co pozwala na tworzenie bardzo kreatywnych projektów bez konieczności zagłębiania się w skomplikowane algorytmy.





## Scenariusz lekcji 14

Czas (minuty):  
90

### Tematyka

Składanie robota

### Cele

Nauka składania robota

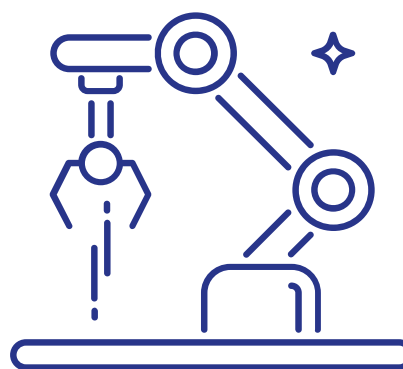
### Oczekiwane efekty

Pomyślnie zmontowany robot - ARTIEbot

## Budowanie robota – 2 opcje

Część tego programu nauczania jest związana z rzeczywistym fizycznym urządzeniem z możliwościami sztucznej inteligencji. Chorwackie Stowarzyszenie Robotyczne opracowało niewielką, przystępną cenowo platformę robota mobilnego, aby wdrożyć wiedzę programistyczną z poprzednich lekcji.

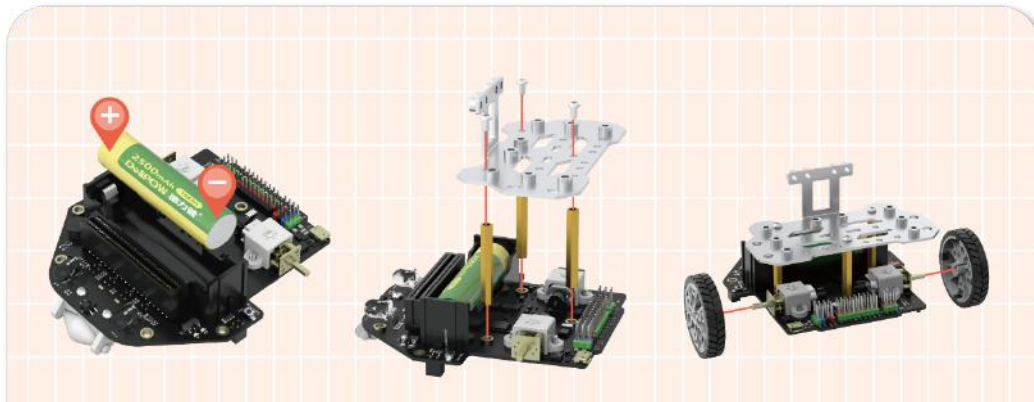
Przedstawienie celu głównego lekcji:  
Składanie robota - ARTIEbot.





# CZĘŚĆ GŁÓWNA

Opcja 1a - Montaż robota Maqueen Plus V1 (zalecany dla początkujących)



1) Włóż baterię 18650

2) Zamontuj wspornik

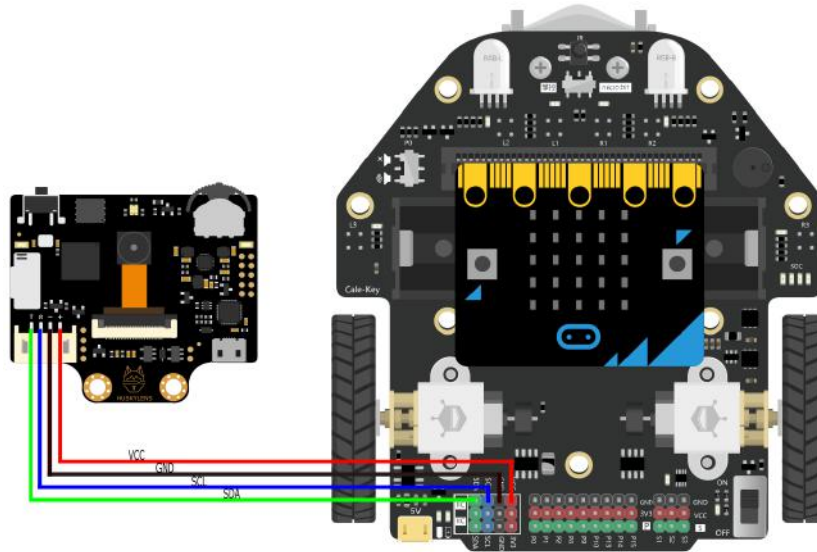
3) Przymocuj koła



4) Zamontuj kamerę HuskyLens

5) Umieść płytkę micro:bit w gnieździe





#### 6) Połącz płytę Maqueen Plus z kamerą HuskyLens

Do ładowania akumulatora 18650 użyj mikrozłącza USB (port ładowania) z tyłu robota:



Battery indicator

Charging port

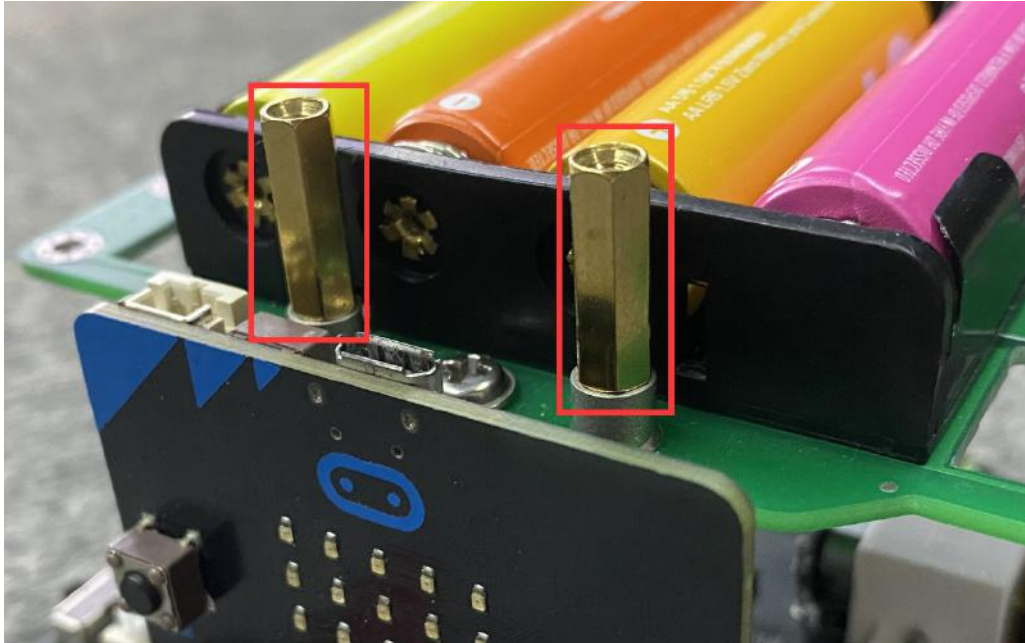
Gdy bateria jest w pełni naładowana, wszystkie diody LED będą się świeciły. Diody LED będą stopniowo gasły w miarę spadania poziomu naładowania baterii. Jeśli wszystkie światła zgasną, oznacza to, że bateria musi zostać ponownie naładowana. Aby to zrobić, użyj ładowarki do telefonu i podłącz ją do portu ładowania.

Proszę pamiętać, że port ładowania jest przeznaczony tylko do ładowania. Do programowania robota Maqueen Plus użyj złącza USB typu Micro na płycie mikrokontrolera micro:bit.

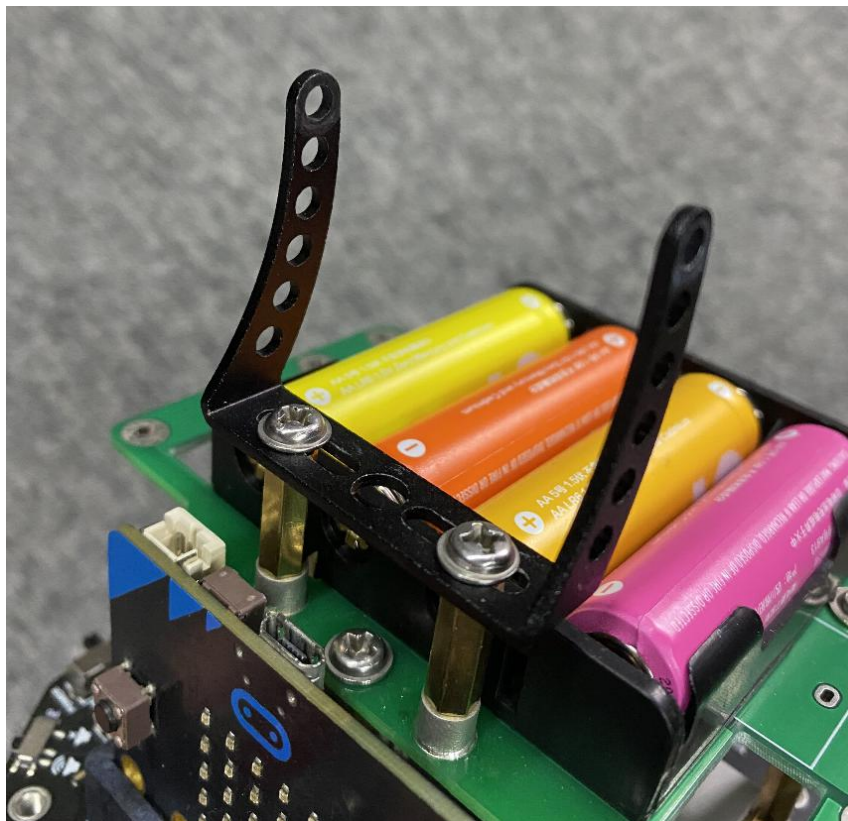
Opcja 1b - Montaż robota Maqueen Plus V2 (zalecany również dla początkujących)

Robot Maqueen plus V2 jest dostępny w postaci wstępnie zmontowanej i wystarczy zamontować na nim kamerę HuskyLens

1. Zainstaluj dwa miedziane słupki dostarczone wraz z produktem w pozycji pokazanej na rysunku.

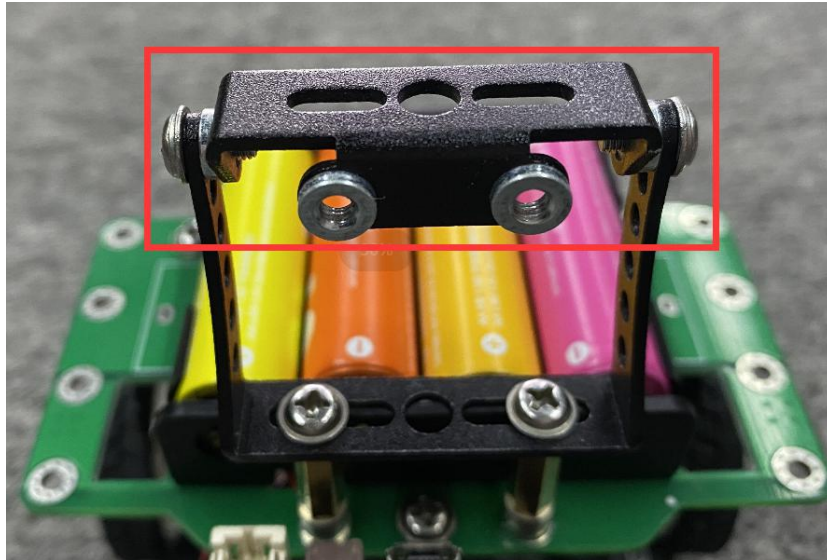


2. Zamocuj uchwyt w kształcie łuku (uchwyt i śruby montażowe są dostarczane wraz z produktem HuskyLens) na kolumnie miedzianej za pomocą śrub.

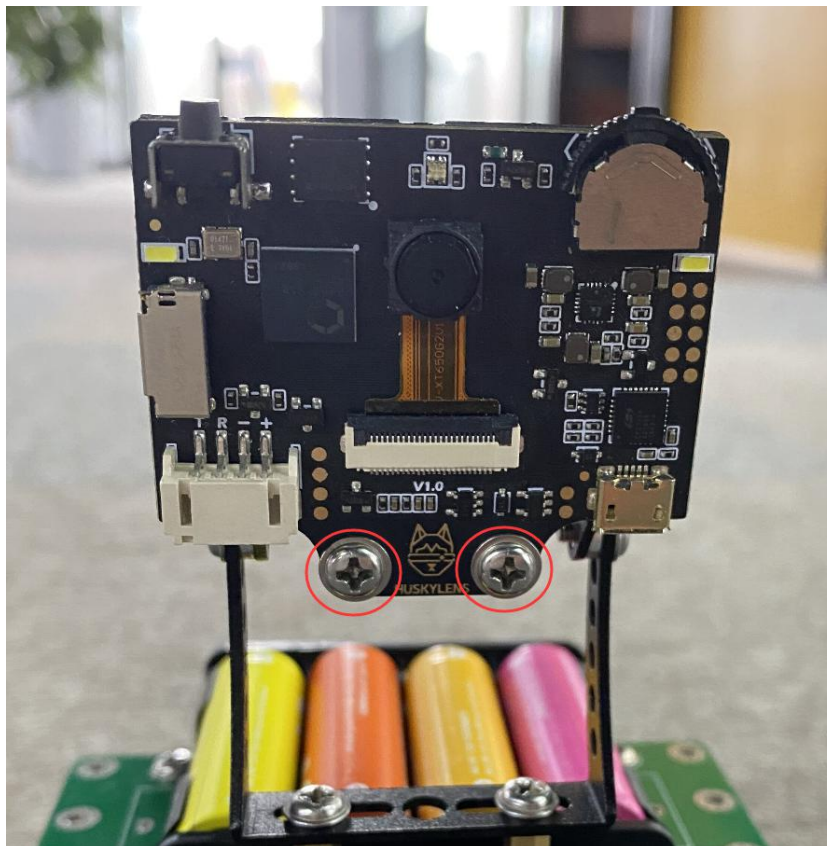




3. Zamontuj drugi uchwyt (uchwyt i śruby montażowe są dostarczane wraz z produktem HuskyLens).



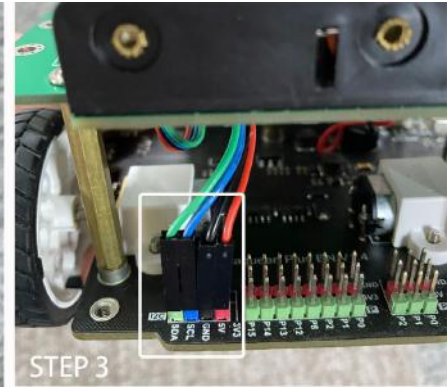
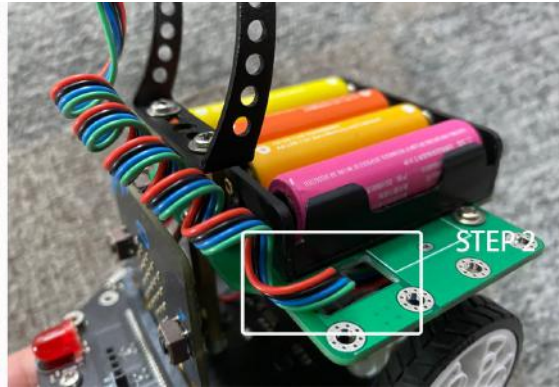
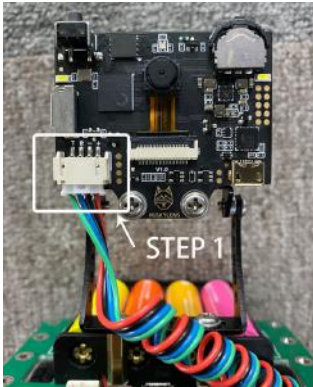
4. Zainstaluj kamerę HuskyLens AI



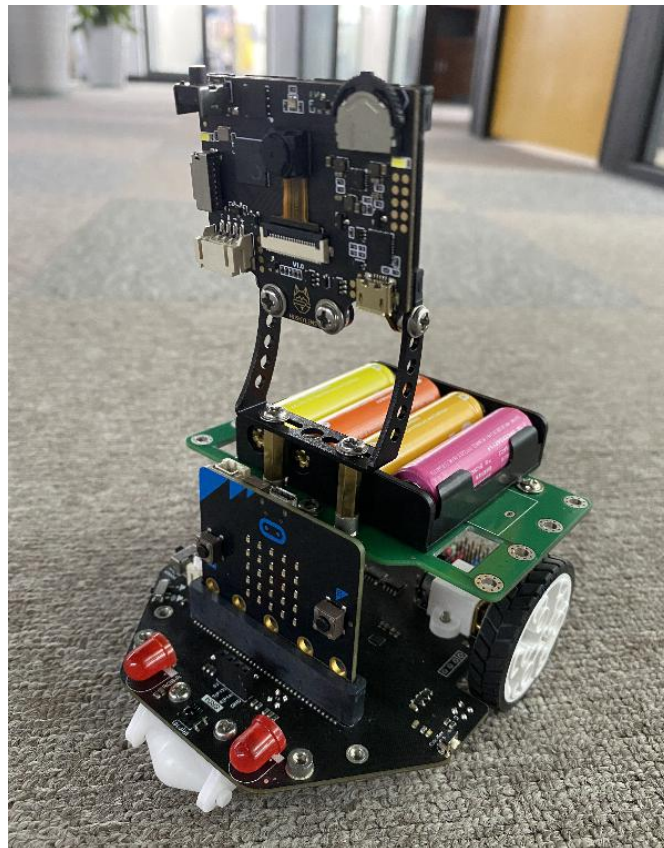




### 5. Podłącz złącze kamery AI



### 6. Instalacja zakończona.





## Opcja 2 - Montaż robota Arduino (tylko dla doświadczonych użytkowników DIY)

### Zestawienie materiałów (BOM)

- 1 x Inteligentny zestaw robota samochodowego dla Arduino  
<https://botland.store/chassis-for-robots/7283-chassis-rectangle-2wd-2-wheel-robot-chassis-with-dc-motor-drive-5904422335649.html>
- 1 x Zestaw przewodów mostkujących  
<https://botland.store/various-wires/1022-connecting-cables-male-male-65pcs.html>
- 1 x Dodatkowy uchwyt na baterie (4 x AA)  
<https://botland.store/battery-holders/173-battery-holder-4-x-aa-r6-5904422329389.html>
- Różne śruby i nakrętki  
<https://botland.store/screws-and-nuts/637-screws-nuts-and-washers-set-330pcs-5410329304478.html>
- 1 x Arduino UNO (oryginał lub kopia)  
<https://store.arduino.cc/collections/boards/products/arduino-uno-rev3>  
<https://botland.store/arduino-compatible-boards-dfrobot/2683-dfduino-uno-v3-compatible-with-arduino.html>
- 1 x Osłona silnika Arduino R3  
<https://store.arduino.cc/collections/shields/products/arduino-motor-shield-rev3>
- 1 x Kamera HuskyLens AI  
<https://store.arduino.cc/collections/dfrobot/products/gravity-huskylens-ai-machine-vision-sensor>
- 8 x Baterie AA (spróbuj użyć akumulatorów)
- 8 x Baterie AA (spróbuj użyć akumulatorów)

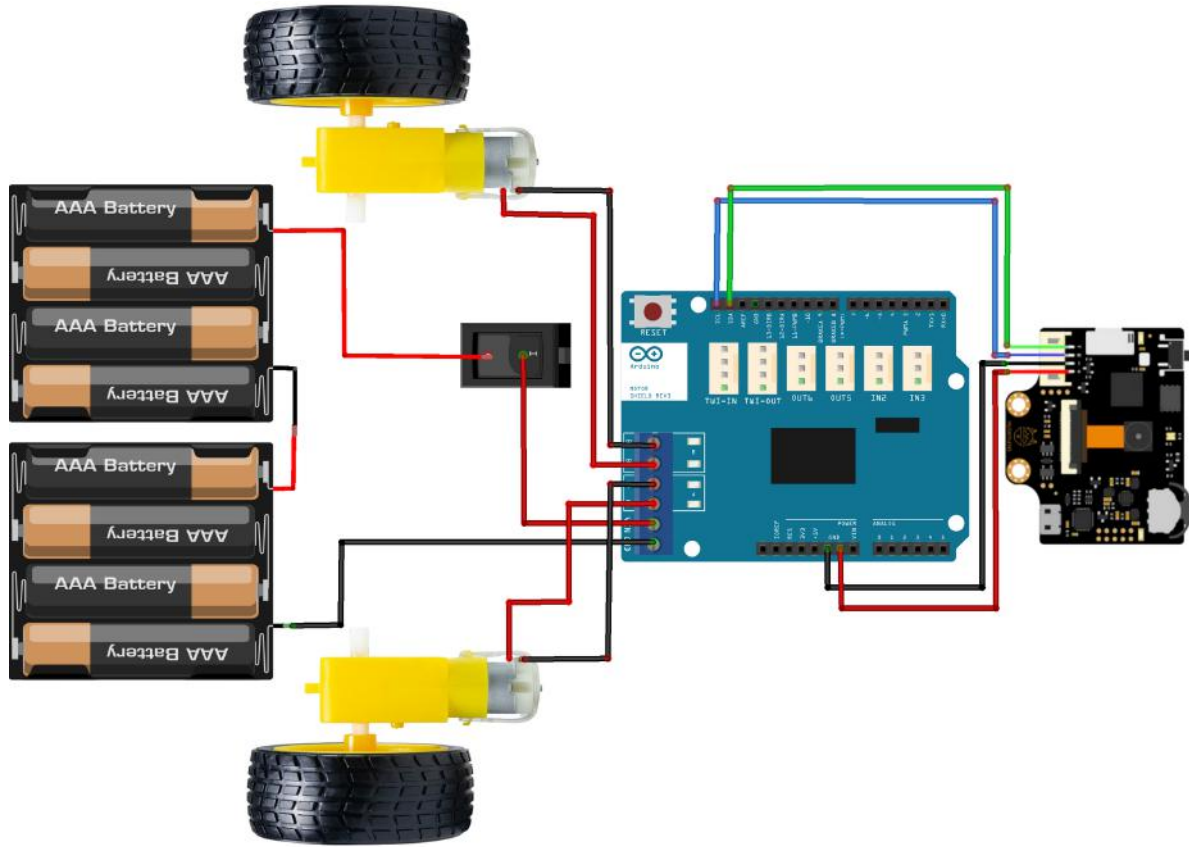
### Wymagane narzędzia

- Śrubokręty (będzie potrzebny jeden z gniazdem i jeden z gniazdem Philips)
- Lutownica i lut
- Taśma izolacyjna lub rurki termokurczliwe
- Szczypce tnące

### Krok 1) Schemat

Oto schemat połączeń. Zauważ, że będziemy używać dwóch uchwytów na baterie podłączonych szeregowo, aby każda bateria w uchwycie była połączona (+ z - i - z +). 8 baterii AA powinno dostarczyć większe napięcie do wejścia zasilania i zapobiec resetowaniu się Arduino lub HuskyLens, gdy silniki zaczynają pracować. Możesz także użyć 3 baterii typu 18650 połączonych szeregowo.





Krok 2) Inteligentny zestaw robota samochodowego dla Arduino - otwórz plastikową torebkę z częściami

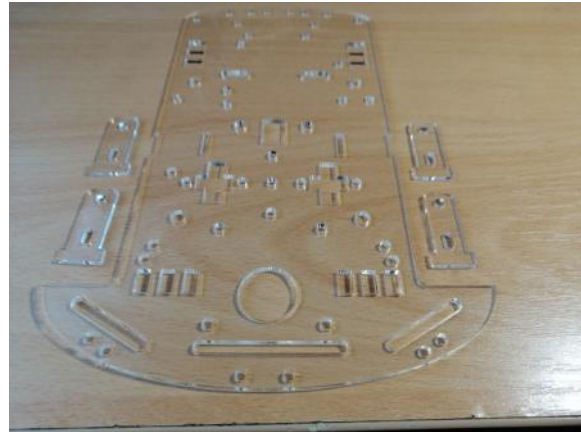


<http://erasmus-artie.eu>

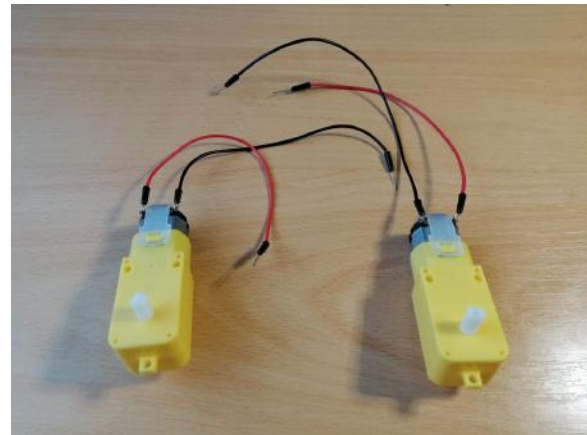




Krok 3) Zdejmij papier ochronny z dużej płyty i wsporników silnika



Krok 4) Przylutuj przewody (lub jumpery) do silników



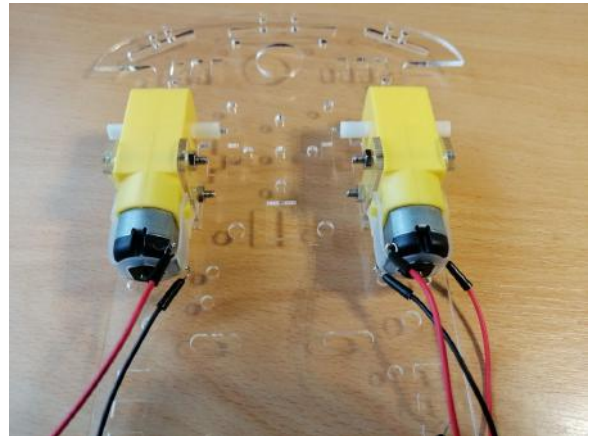
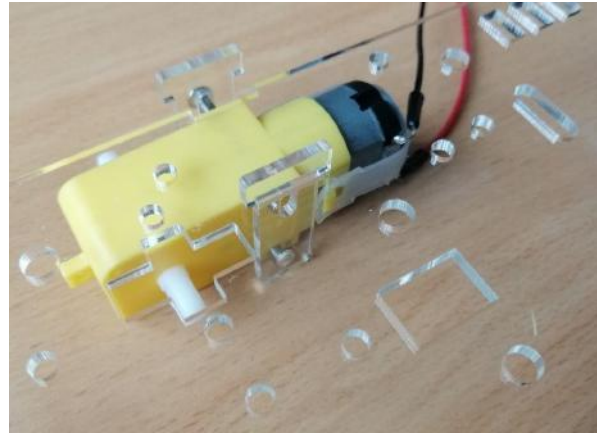
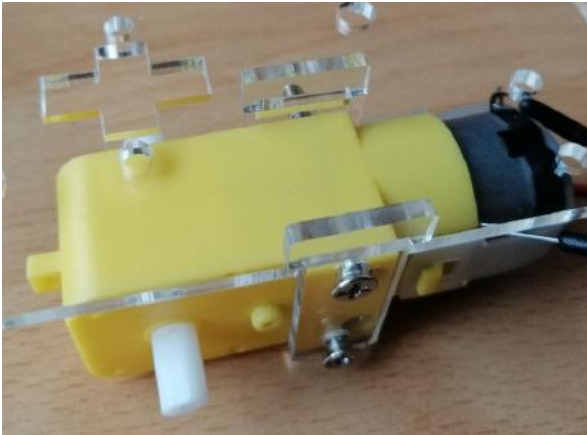
Obejrzyj ten krótki film, aby zobaczyć, jak to jest zrobione:

<https://www.youtube.com/watch?v=xSWKnnvGWBs>

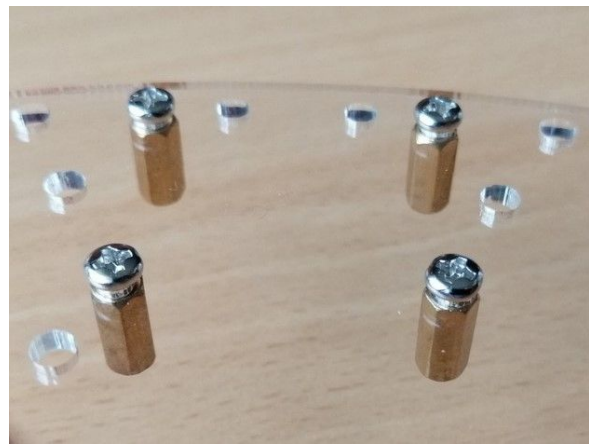
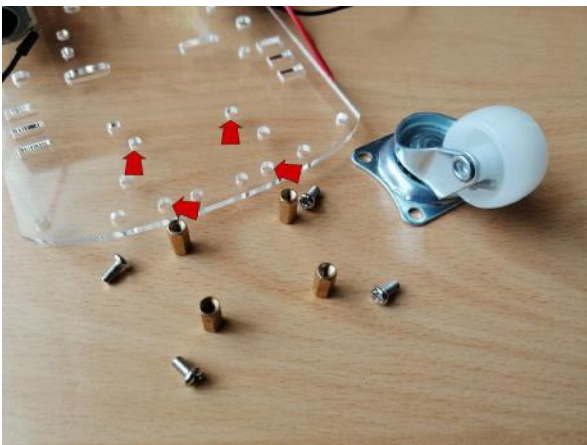
Poproś nauczyciela lub rodzica, aby Ci w tym pomógł, jeśli nie masz wcześniejszego doświadczenia lub narzędzi.

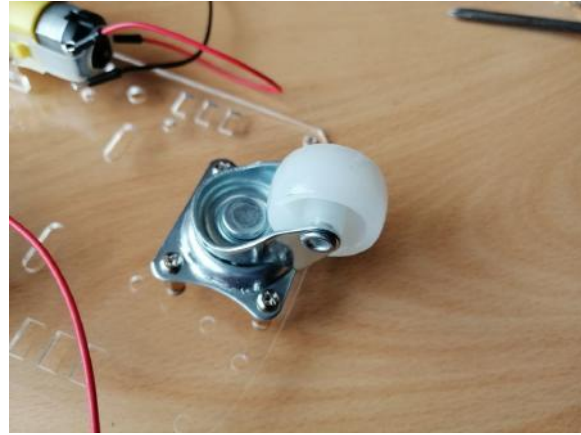
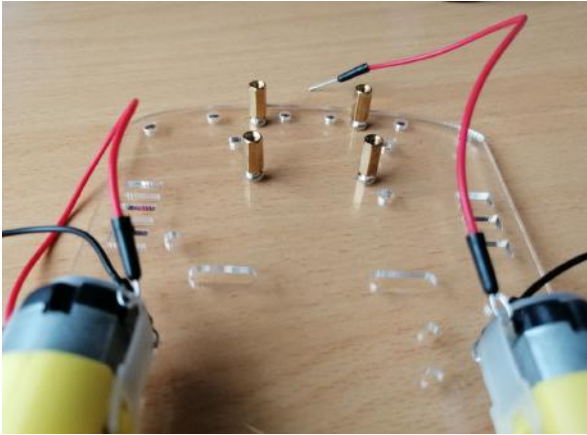


Krok 5) Zamontuj silniki za pomocą „wsporników T”

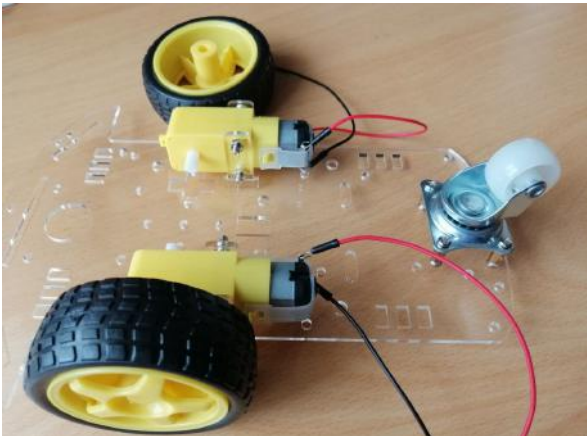


Krok 7) Zamontuj małe (tylne) koło za pomocą śrub dystansowych

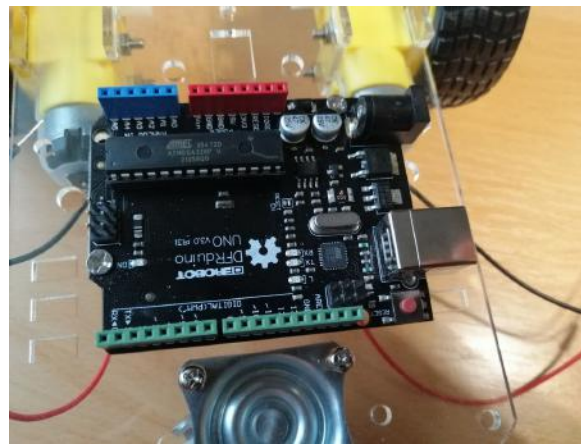
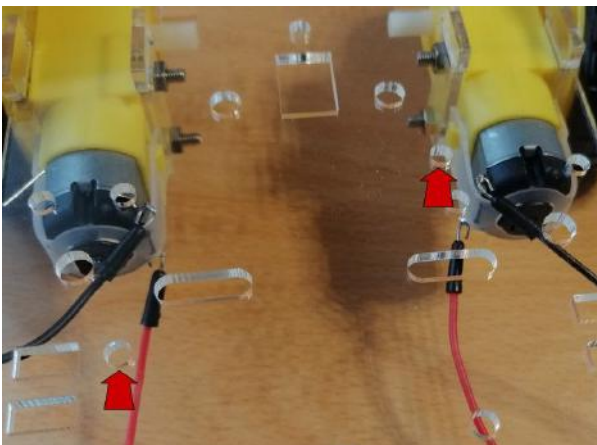


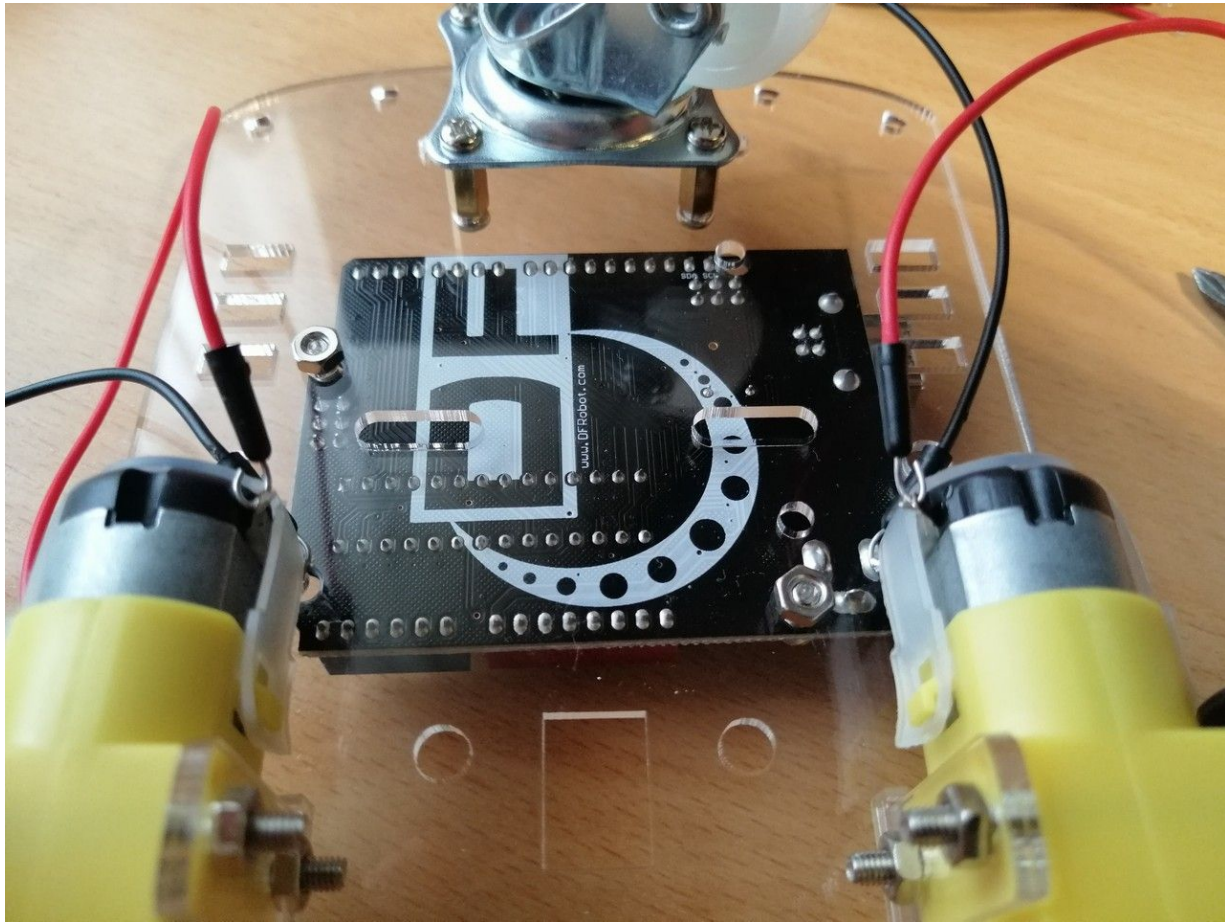


Krok 8) Zamocuj koła na wałkach silnika - sprawdź kształt wałka i kształt otworów na koła



Krok 9) Zamontuj Arduino UNO zgodnie z zaznaczonymi otworami





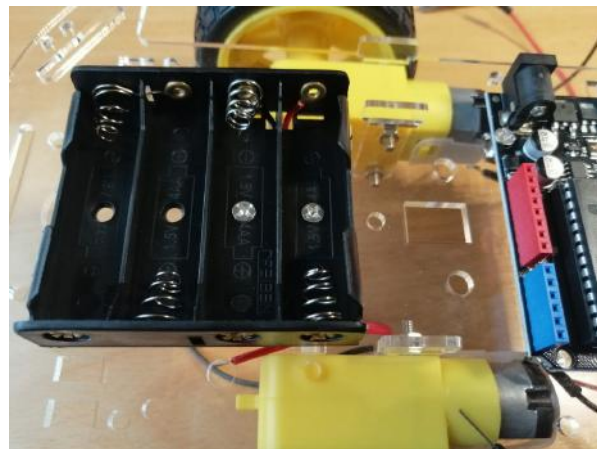
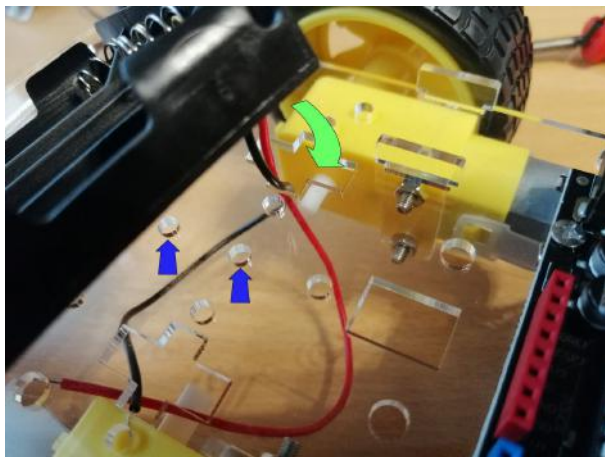
<http://erasmus-artie.eu>

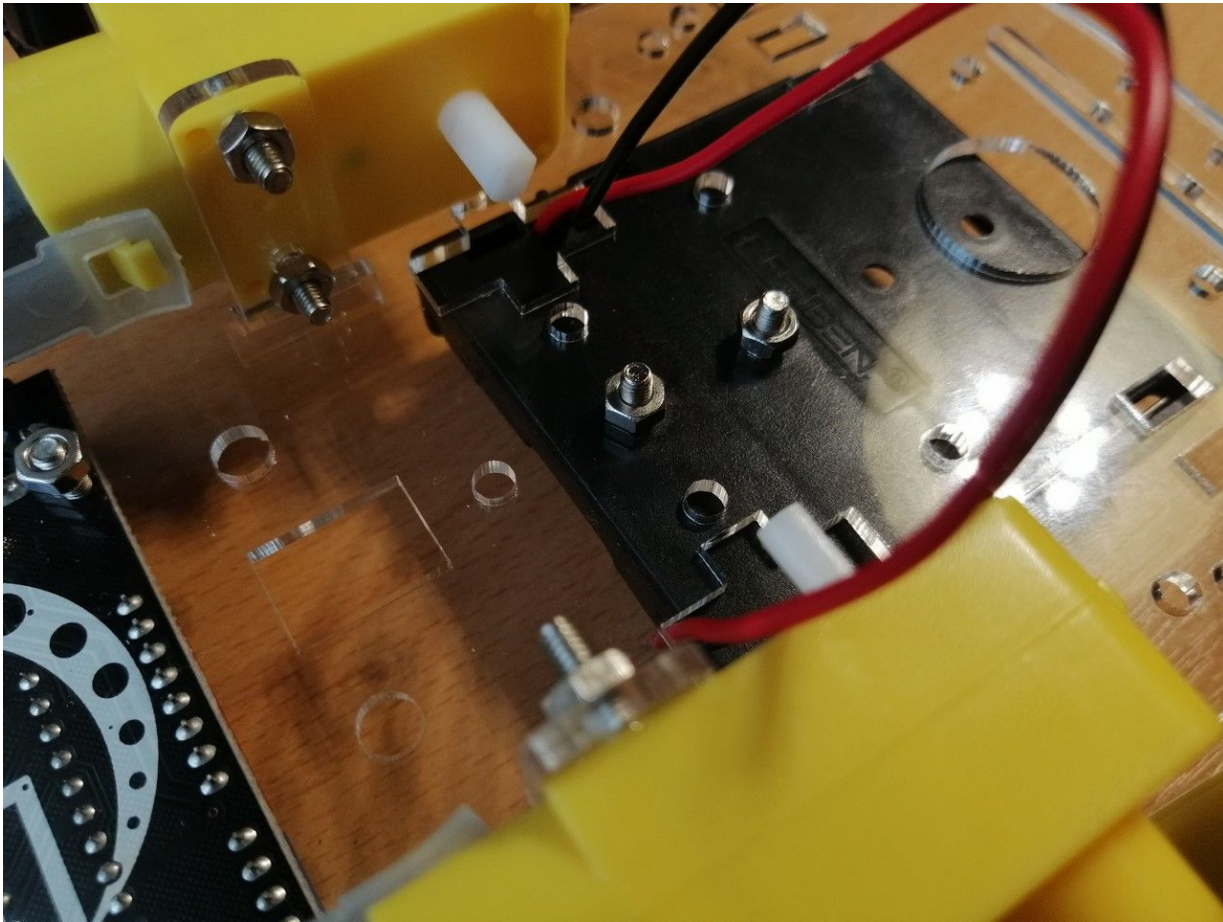
111

Możesz sprawdzić ten film, aby uzyskać szczegółowe informacje, chociaż stosujemy nieco inne podejście:

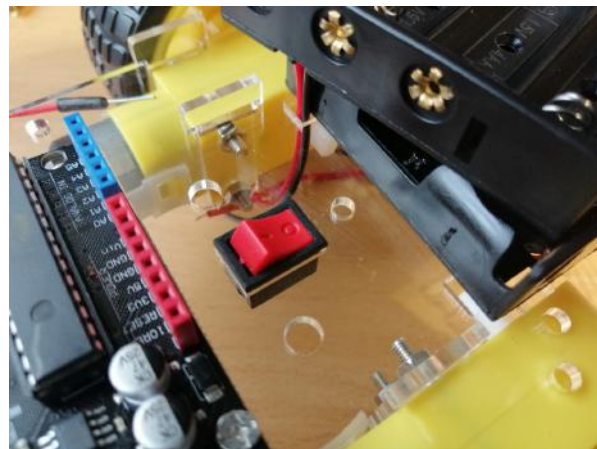
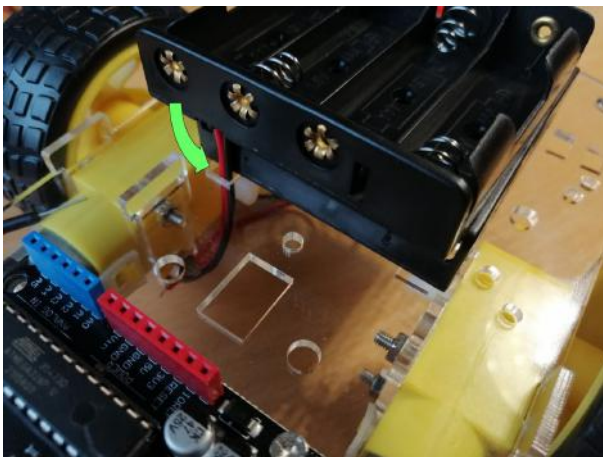
<https://www.youtube.com/watch?v=3a-bE1VlaU8>

Krok 10) Zamontuj pierwszy (dolny) uchwyt baterii za pomocą otworów (niebieskich)  
Najpierw przeciągnij kable uchwytu przez otwór (zielony)





**Krok 11) Przeciągnij drugie (górne) kable uchwytu baterii przez zaznaczony otwór (zielony)**  
Umieść przełącznik w otworze między Arduino UNO a uchwytem baterii

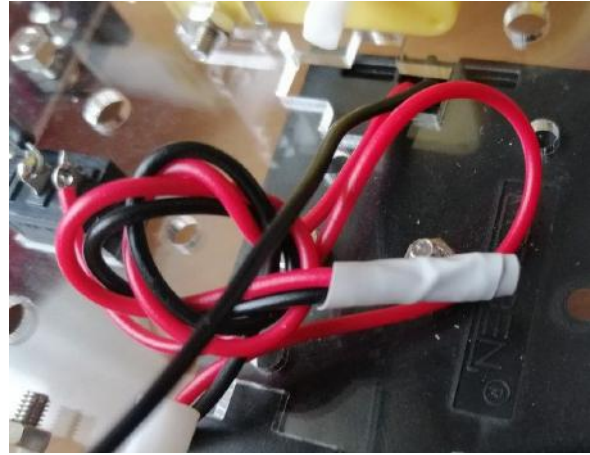
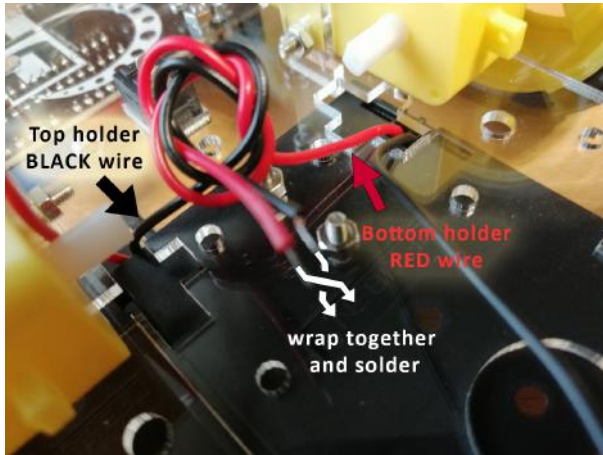






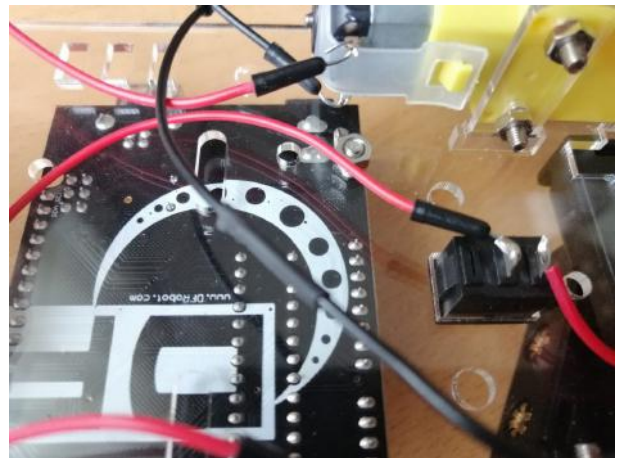
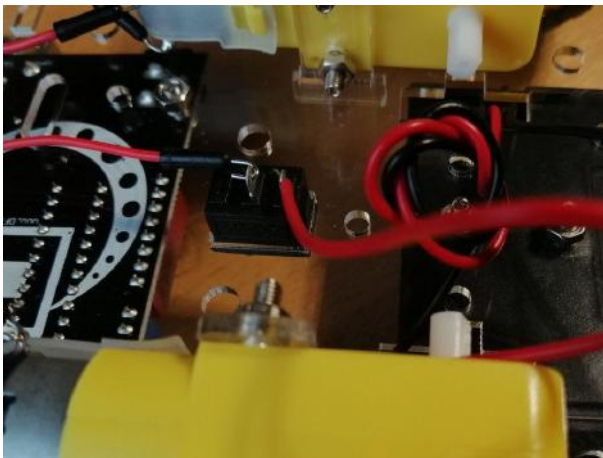
Krok 12) Połącz CZERWONY przewód z dolnego uchwyty baterii z CZARNYM przewodem z górnego uchwyty baterii

Zaizoluj złącze rurką termokurczliwą lub taśmą izolacyjną.



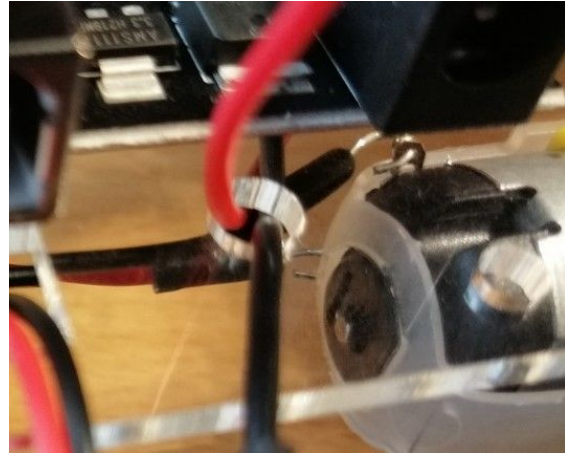
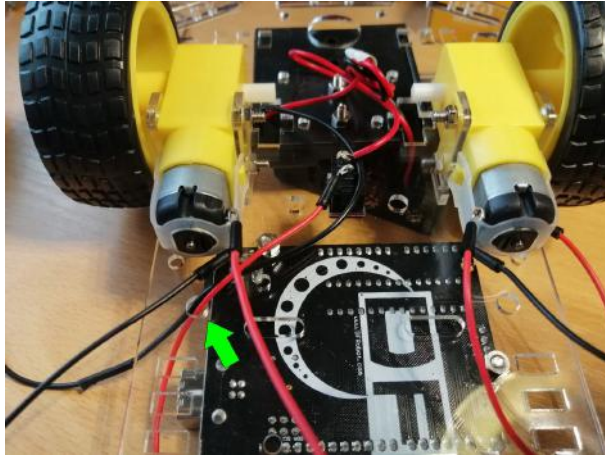
Krok 13) Przylutuj CZERWONY przewód od dolnego uchwyty do przełącznika

Następnie przylutuj czerwony jumper do drugiego styku przełącznika. Przylutuj czarny jumper do dolnego uchwyty CZARNEGO przewodu. Zaizoluj złącze rurką termokurczliwą lub taśmą izolacyjną.

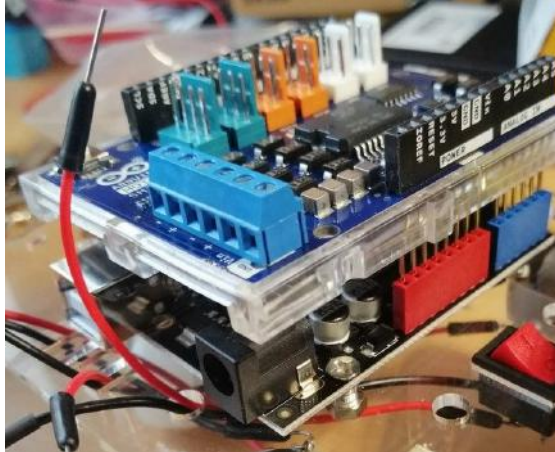




Krok 14) Przeciągnij czarny i czerwony przewód zasilający przez zaznaczony otwór (zielony)



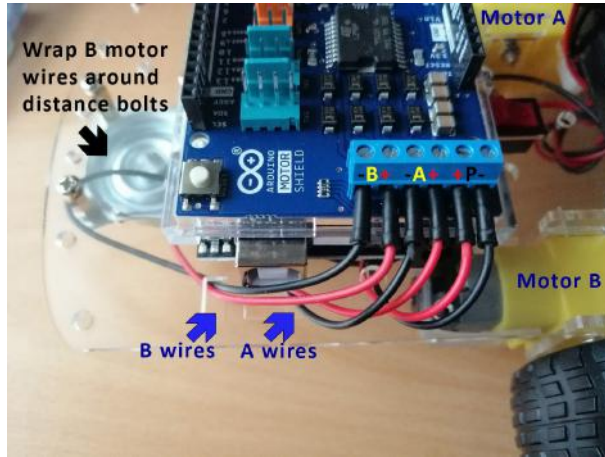
Krok 15) Umieść osłonę silnika Arduino na płycie Arduino UNO  
Wyrównaj piny i mocno je dociśnij.





**Krok 16) Przeciagnij przewody silnika przez zaznaczone otwory.**

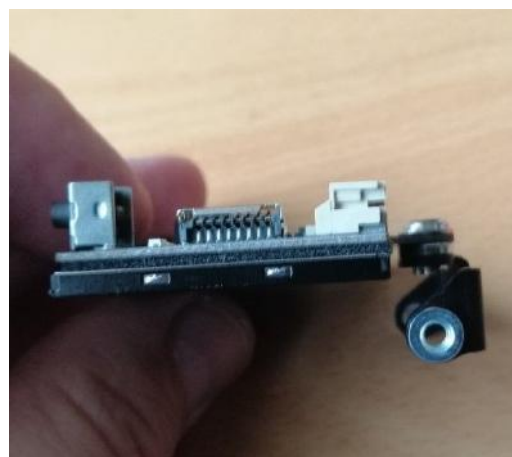
Skróć piny przewodów mostkujących za pomocą szczypiec tnących i podłącz je do złącz śrubowych osłony silnika zgodnie z oznaczeniem (- Silnik B + - Silnik A + + Zasilanie -).



**Krok 17) Rozpakuj kamerę HuskyLens AI**

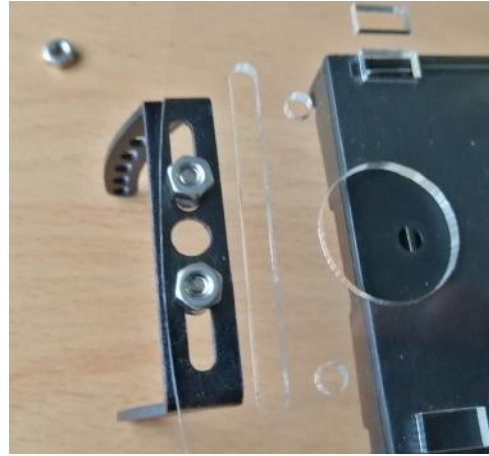


**Krok 18) Zamontuj mały wspornik na kamerze - użyj śrub z pudełka HuskyLens**

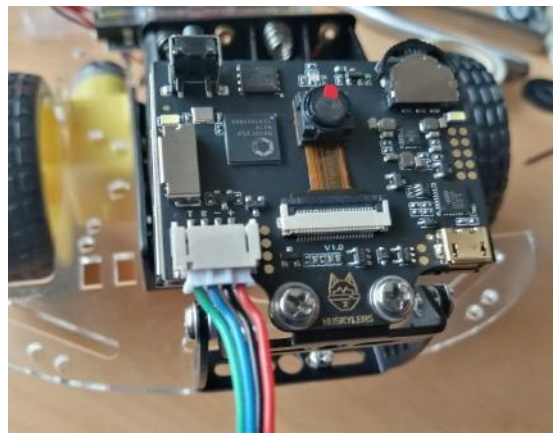




Krok 19) Zamontuj większy wspornik kamery na obudowie robota



Krok 20) Połącz dwa wsporniki za pomocą śrub (niezbyt mocno) i podłącz kabel kamery

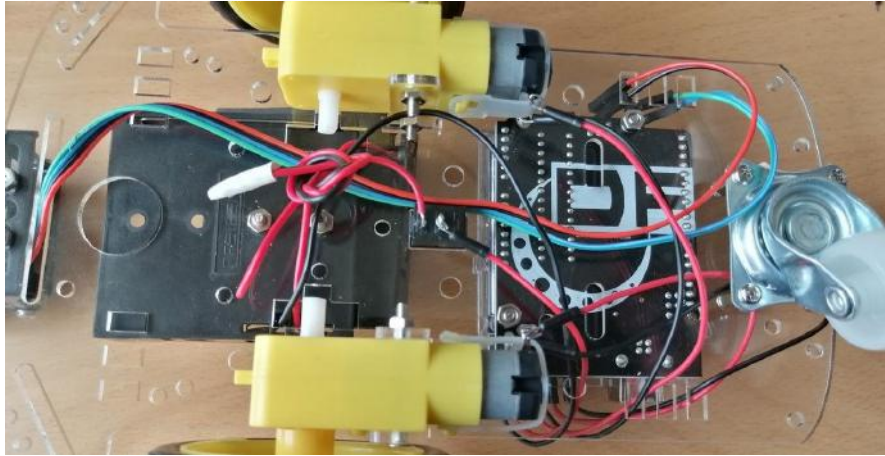


Krok 21) Przeciągnij kabel kamery przez otwór

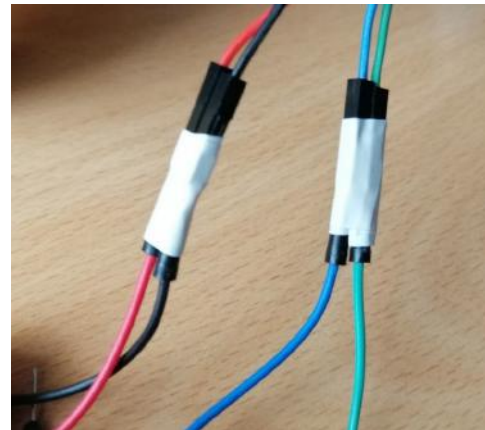
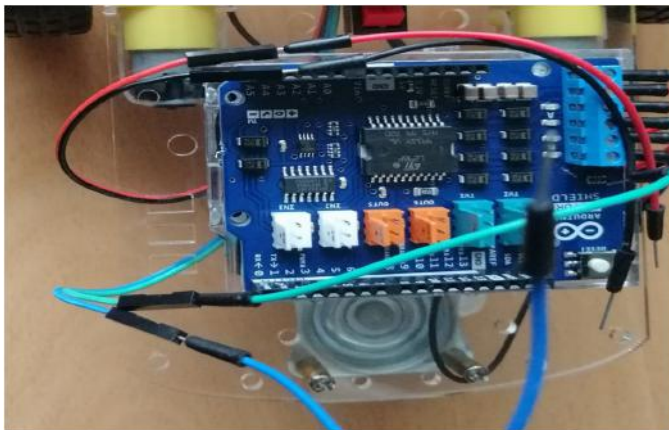




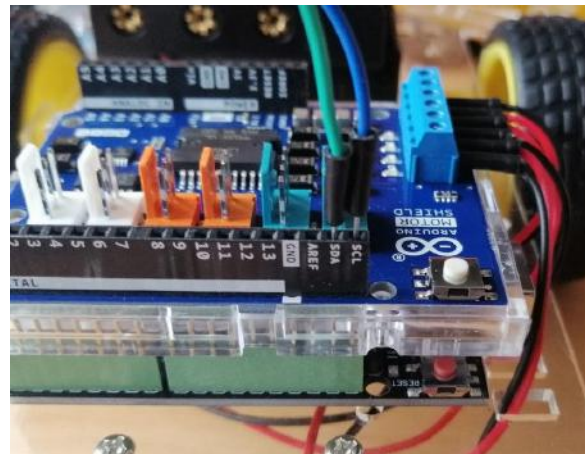
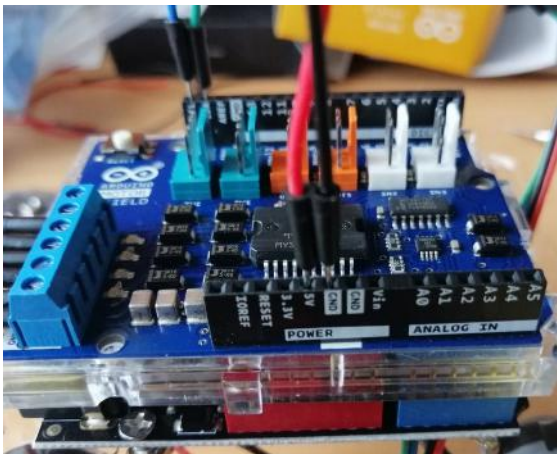
Krok 22) Odwróć robota i przeciągnij kabel kamery przez otwory, jak na poniższym obrazku



Krok 23) Podłącz przewody mostkujące (jumpery) do kabla kamery - dopasuj kolor jumperów  
Napraw połączenia za pomocą taśmy izolacyjnej.



Krok 24) Włóż przewody mostkujące zasilania (power jumpers) oraz przewody I2C do pinów tarczy silnikowej. Czerwony przewód do pinu 5V, czarny do pinu GND, niebieski do pinu SCL i zielony do pinu SDA.



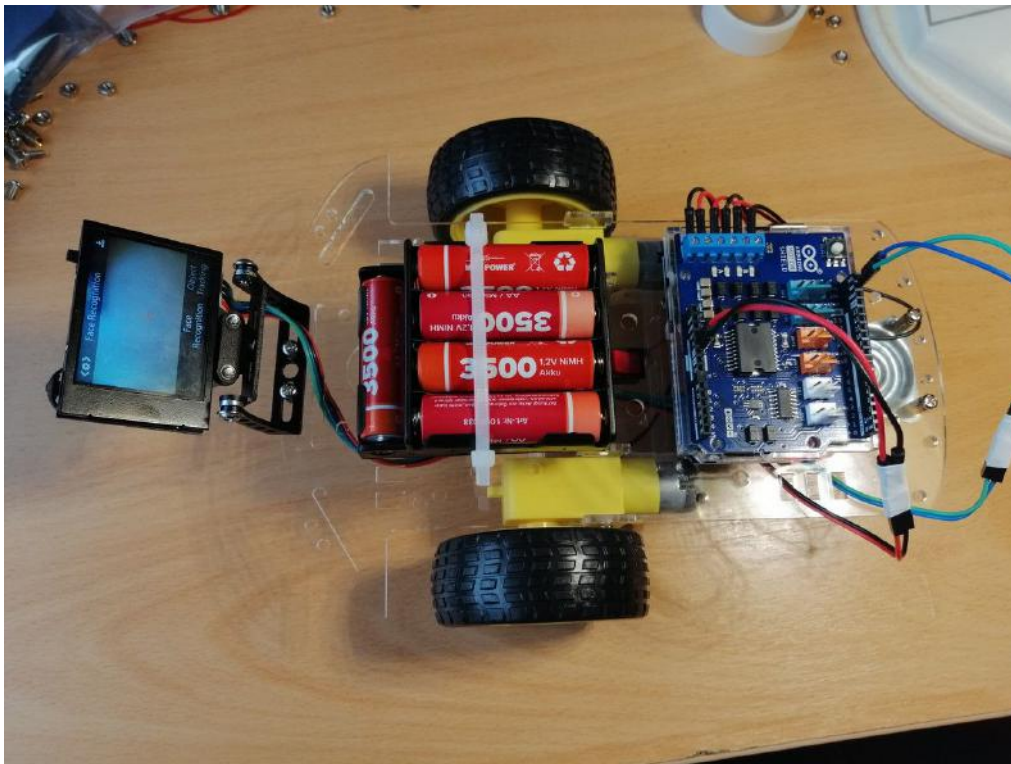


### Krok 25) Umieść baterie w obu uchwytach

Użyj 2 napinaczy kabli, aby zamocować górne i dolne uchwyty baterii (nie za mocno – górny uchwyt powinien być w stanie poruszać się w przód iw tył)



Krok 26) Włącz przełącznik - jeśli działa zgodnie z oczekiwaniami - kamera HuskyLens również się włączy.





## Scenariusz lekcji 15

Czas (minuty):  
90

### Tematyka

Programowanie robota do poruszania się

### Cele

Nauka programowania robota do poruszania się

### Oczekiwane efekty

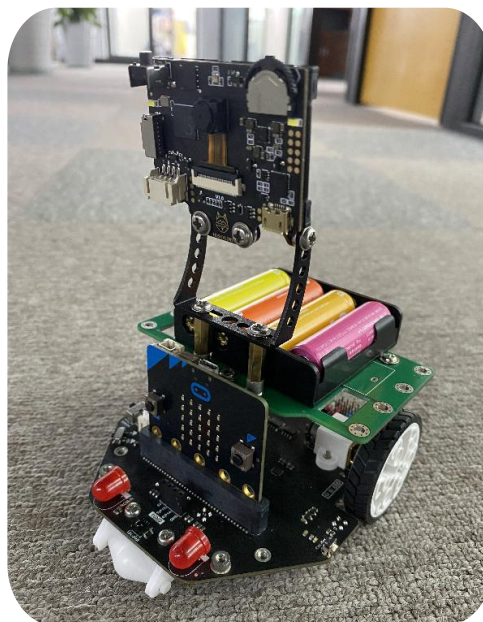
Wiedza, jak napisać program, aby robot się poruszał

## Programowanie robotów - poruszanie się

Roboty są fajne. W tej lekcji pokazujemy krok po kroku, łatwe do zrozumienia przykłady programowania ruchów autonomicznego robota mobilnego.

### Przedstawienie celu głównego lekcji:

Napisz nasz pierwszy program, za pomocą którego uruchomimy robota.

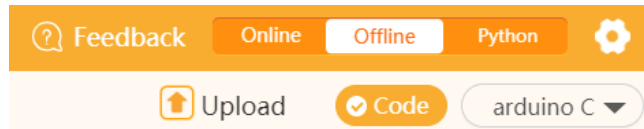




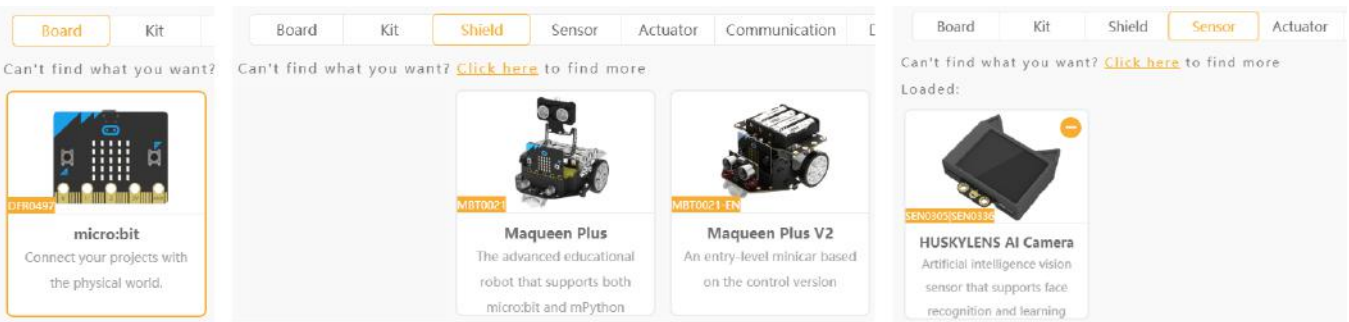
# CZĘŚĆ GŁÓWNA

Nauczyciel pomaga uczniowi w napisaniu kodu złożonego przez nas robota i umożliwieniu mu poruszania się.

Krok 1 (obie opcje): Przejdź do strony: <http://mindplus.cc/download-en.html> i pobierz wersję programu odpowiednią dla systemu operacyjnego Twojego komputera. Zainstaluj program i uruchom Mind+. Po uruchomieniu, przełącz się w tryb Offline.



eśli pracujesz z Arduino, przejdź bezpośrednio do kroku 2. -> Maqueen Plus



<http://erasmus-artie.eu>

Kliknij na "Extensions" (Rozszerzenia) i w zakładce "Board" (Płyta) wybierz "micro:bit", w zakładce "Shield" wybierz "Maqueen Plus" lub "Maqueen Plus V2", a w zakładce "Sensor" wybierz "HUSKYLENS AI Camera". Kliknij na przycisk "Back" (Wstecz), aby wrócić, i oprogramowanie jest gotowe do użycia wybranych modułów.







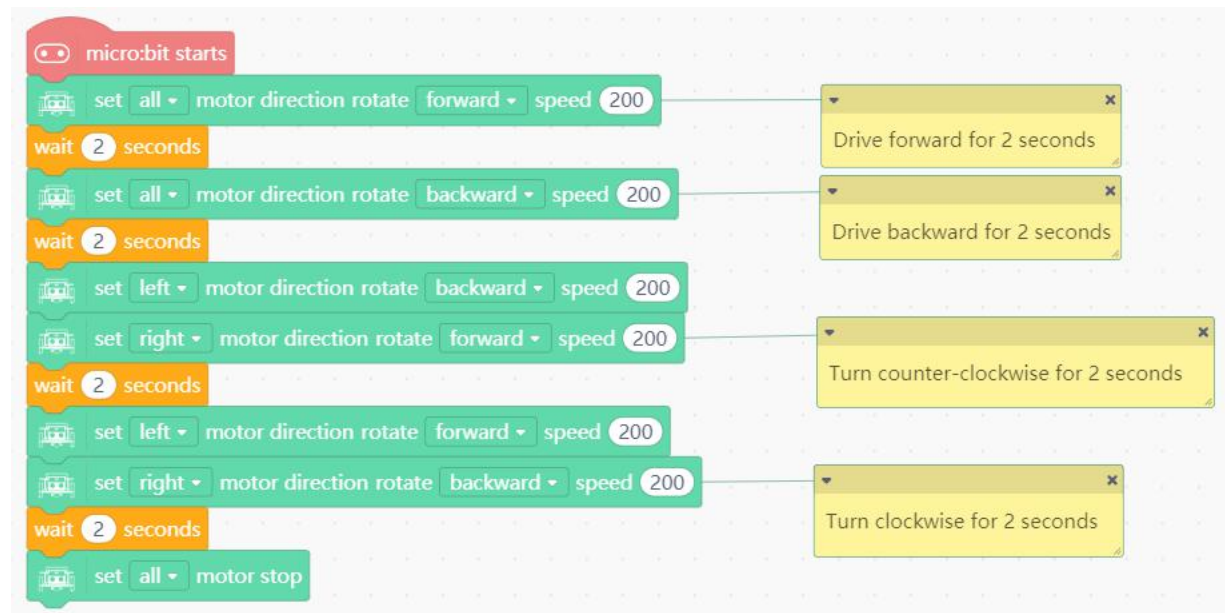
Podłącz mikro:bit do komputera za pomocą kabla USB micro, a wskaźnik zasilania się zaświeci. Przetestuj sekwencję przesyłania za pomocą tego kodu:



Kliknij przycisk "Upload" (Prześlij).

Robot powinien poruszać się do przodu przez jedną sekundę, a następnie zatrzymać się.

Wypróbuj tę sekwencję - robot powinien poruszać się zgodnie z opisem w komentarzach.



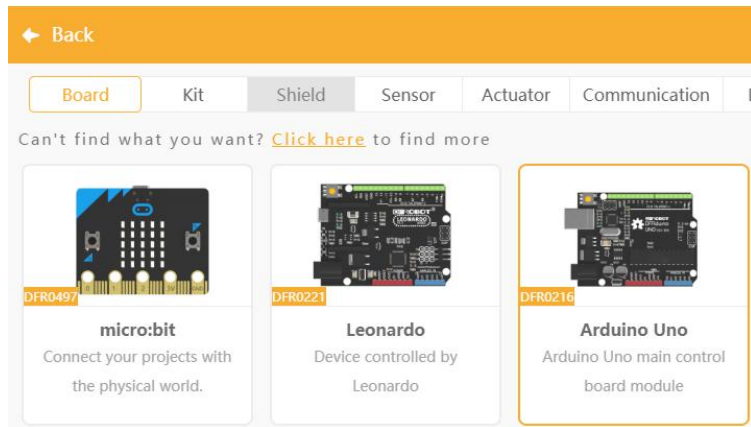
Kliknij przycisk "Upload" (Prześlij).

Robot powinien poruszać się zgodnie z opisem w komentarzach (żółte bloki).

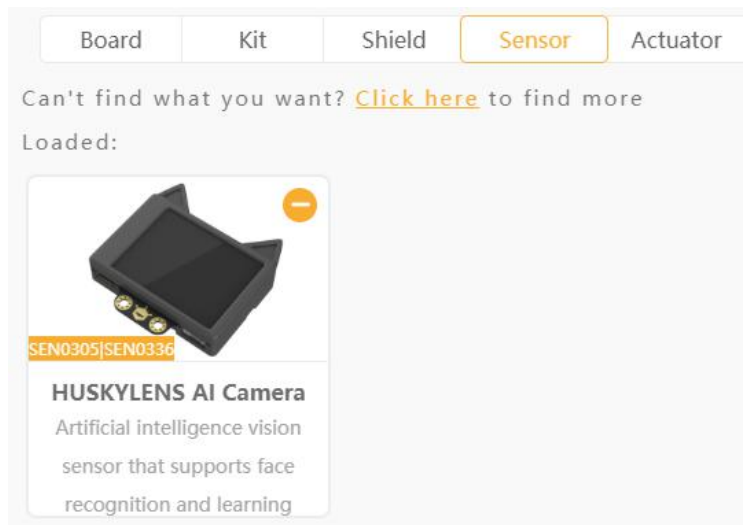
Spróbuj różnych prędkości, aby sprawić, by Twój robot poruszał się szybciej lub wolniej.



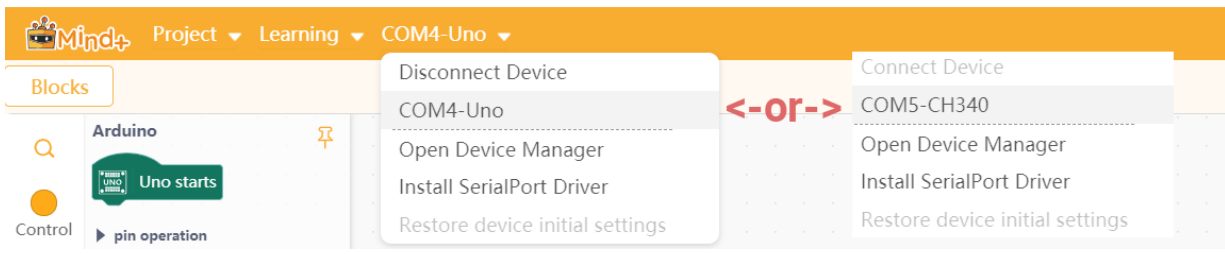
Krok 2: Otwórz zakładkę "Extensions" (Rozszerzenia) i wybierz płytkę - Arduino UNO.



Krok 3: Przejdź do zakładki "Sensor" (Czujnik) i wybierz czujnik - HUSKYLENS AI Camera.



Krok 4: Po wyborze czujnika, kliknij na przycisk "<- Back" (Wstecz), a następnie jesteś gotowy do użycia bloków Arduino i Sensor. Przejdźmy do testu, aby sprawdzić, czy wszystko działa. Przed tym musisz podłączyć urządzenie. Podłącz swoją płytkę Arduino UNO za pomocą kabla USB i wybierz odpowiedni port COMX-Uno (lub CH340), w zależności od producenta Twojej płytki Arduino.





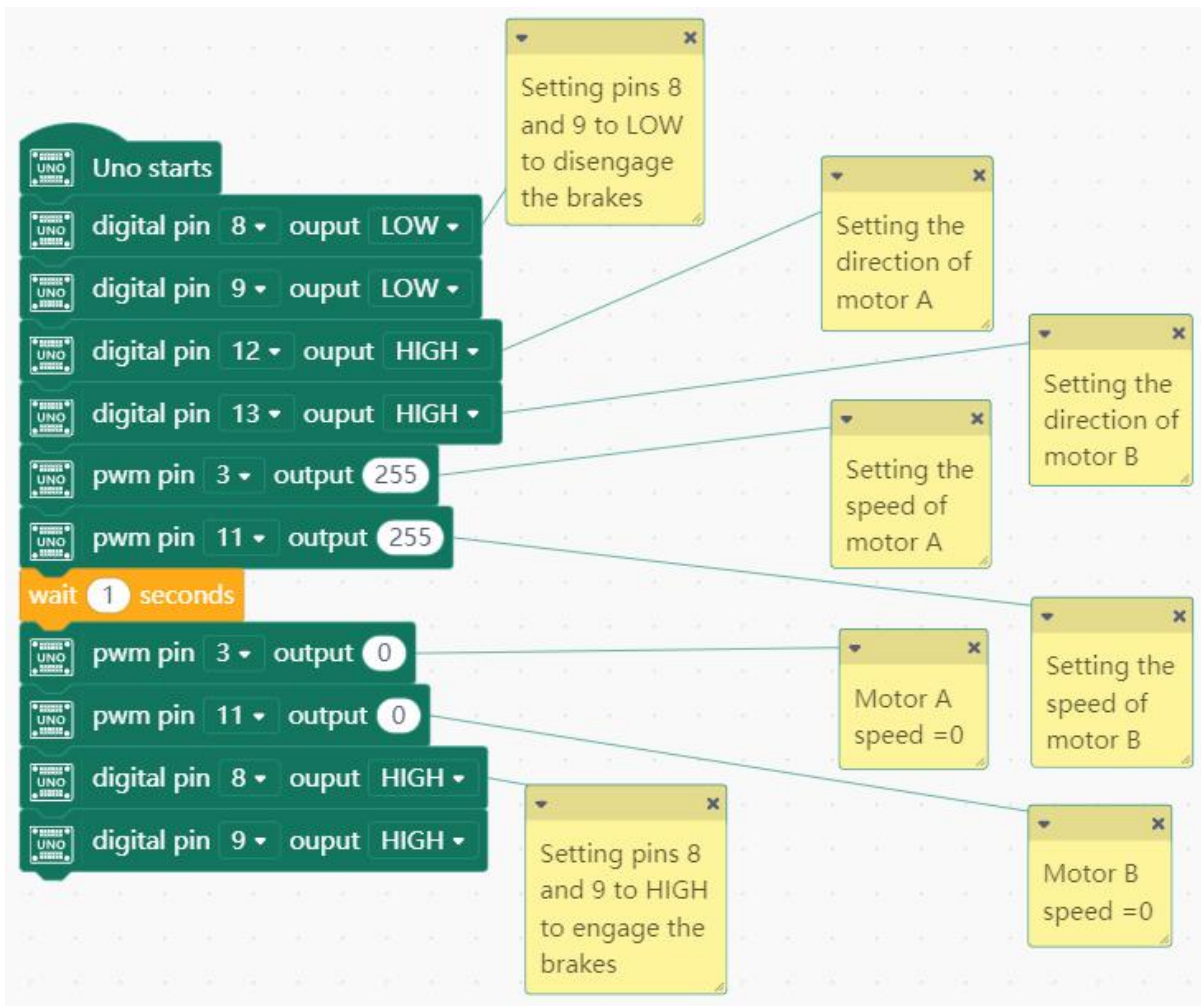
### Podstawowy ruch

Pamiętasz tę tabelę ze scenariusza wprowadzenia sprzętu?

Function	Channel A	Channel B
Direction	Digital 12	Digital 13
Speed (PWM)	Digital 3	Digital 11
Brake	Digital 9	Digital 8
Current Sensing	Analog 0	Analog 1

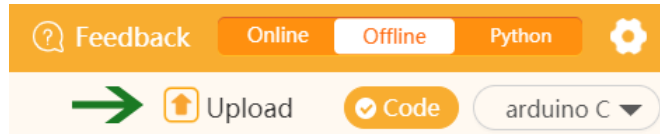
Dwa silniki (A i B) są lewym i prawym silnikiem. Wyprowadzenia cyfrowe 12 i 13 są używane do zmiany kierunku (HIGH - jeden kierunek, LOW - przeciwny kierunek), a wyprowadzenia PWM 3 i 11 są używane do ustawiania prędkości (0-255). Wyprowadzenia 9 i 8 włączają/wyłączają hamulce (HIGH - hamulce włączone, LOW - hamulce wyłączone).

Poniżej możecie zobaczyć przykładowy kod z komentarzami po prawej stronie, które pomogą Wam zrozumieć, jak to działa.





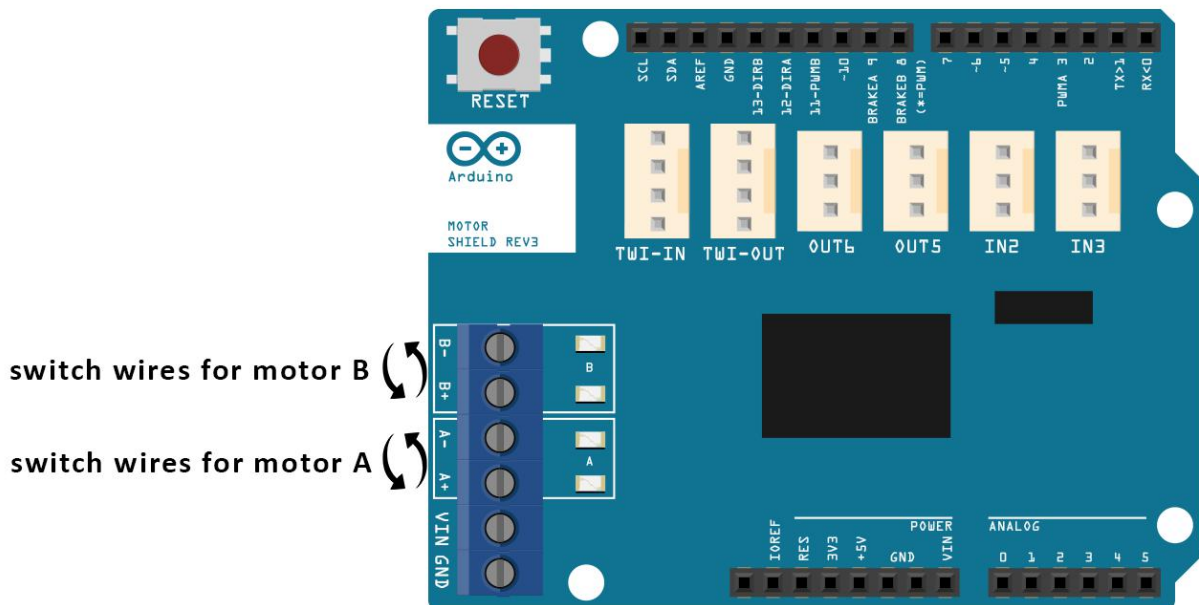
Krok 1a: Naciśnij przycisk "Upload" (Prześlij), aby przesłać ten kod do płytki Arduino UNO.



Krok 2a: Bądź ostrożny, robot zacznie się poruszać natychmiast po zakończeniu przesyłania kodu! Robot powinien poruszać się do przodu przez jedną sekundę, a następnie zatrzymać się.

Rozwiązywanie problemów - jeśli robot:

- porusza się do tyłu - zamień miejscami czerwone i czarne przewody w terminalach obu silników.
- obraca się zgodnie z ruchem wskazówek zegara - zamień miejscami czerwone i czarne przewody w terminalach silnika B.
- obraca się przeciwnie do ruchu wskazówek zegara - zamień miejscami czerwone i czarne przewody w terminalach silnika A.



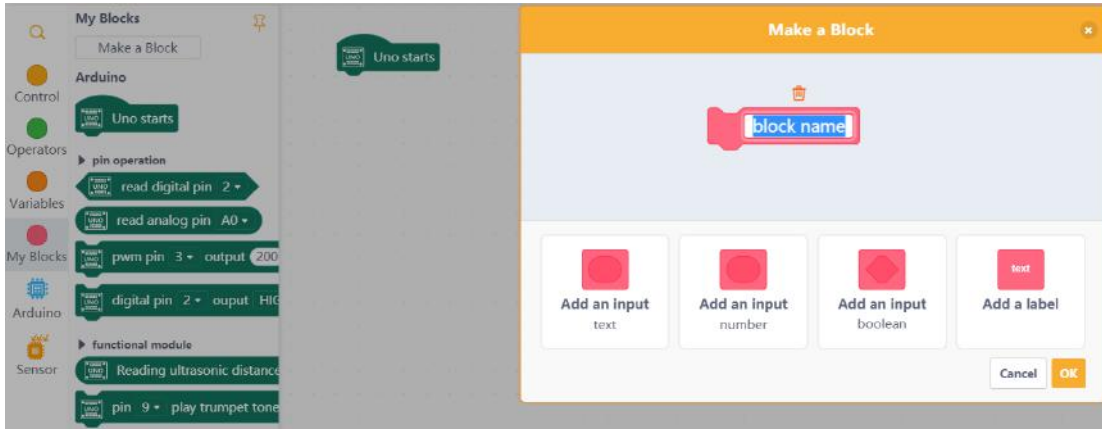
<http://erasmus-artie.eu>

Teraz twój robot powinien poruszać się do przodu, przy użyciu stanu HIGH na pinach 12 i 13. Istnieje także prostszy sposób programowania ruchu - będziemy używać naszych własnoręcznie wykonanych bloków zamiast powtarzać całą serię bloków ustawiających piny.

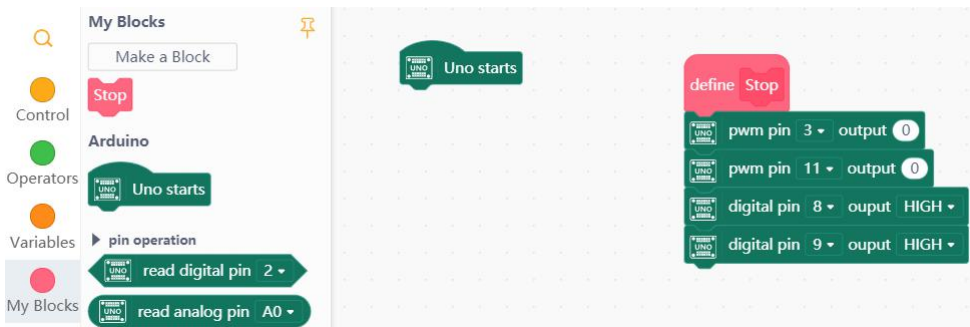
Krok 1b: Kliknij na grupę "My Blocks" (Czerwona).

Krok 2b: Kliknij przycisk "Make a Block" (Utwórz blok) i zmień "nazwę bloku" na "Stop".

Krok 3b: Kliknij OK.

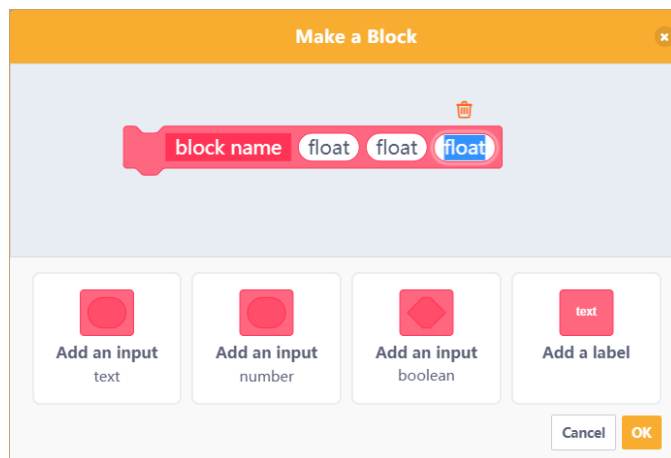


Masz już swój pierwszy blok, ale obecnie nie jest on jeszcze powiązany z żadnym działaniem, więc musimy go najpierw zdefiniować.



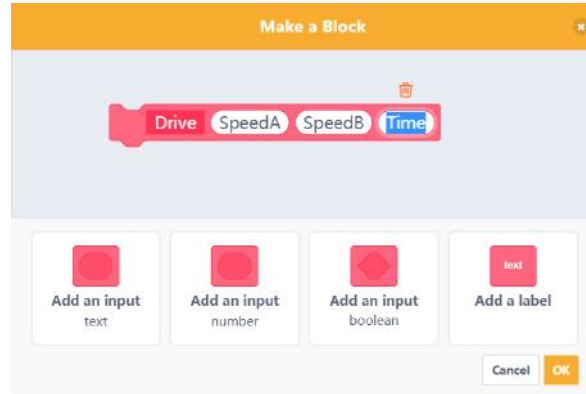
Blok "Stop" zawiera 4 bloki. Pierwsze dwa bloki ustawiają prędkość obu silników na 0, a ostatnie dwa bloki włączają hamulce. Jeśli chcesz użyć trybu bezhamulcowego, ustaw stan pinów cyfrowych 8 i 9 na LOW.

Teraz nadszedł czas, aby utworzyć blok wejściowy z 3 parametrami liczbowymi, które będą przechowywać prędkość obu silników oraz czas trwania. Kliknij ponownie przycisk "Make a Block" (Utwórz blok), a następnie kliknij "Add an input" (Dodaj wejście) - "number" - 3 razy, a rezultat powinien być następujący:





Zmień nazwę bloku na "Drive", pierwszy parametr "float" na "SpeedA", drugi parametr "float" na "SpeedB", a trzeci parametr "float" na "Time". Następnie kliknij OK.

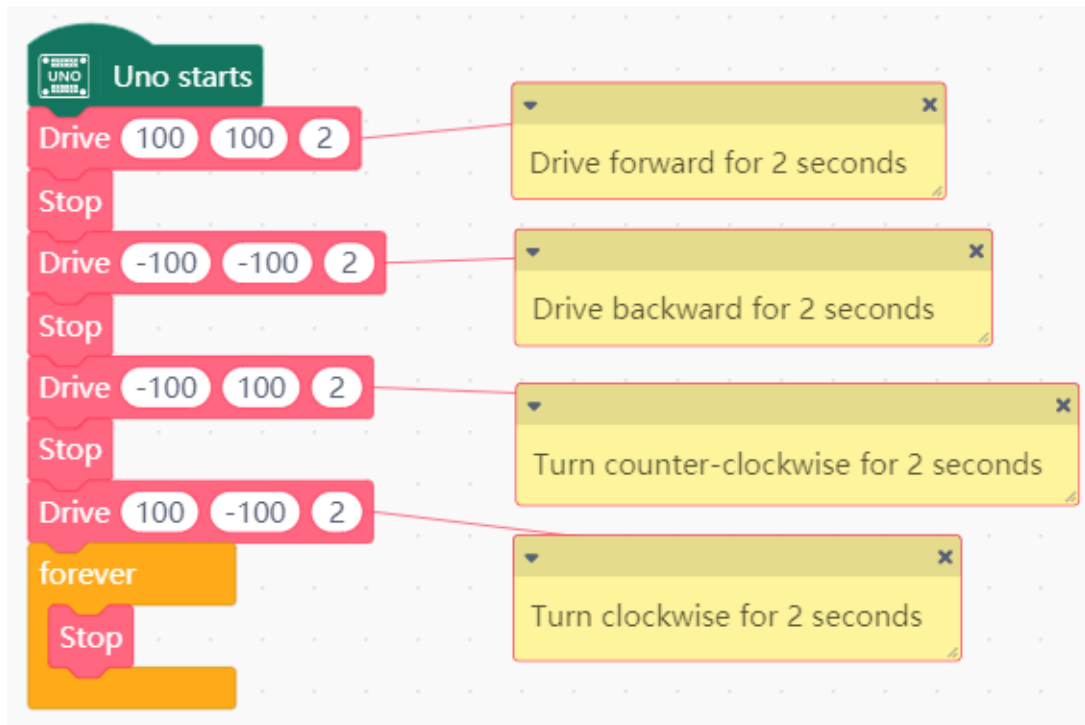


Podstawowym pomysłem jest pobranie wartości SpeedA i SpeedB (akceptowalny zakres: -255 do 255), sprawdzenie, czy którakolwiek z nich jest ujemna (lub obie), i jeśli tak - odwrócenie kierunku poprzez ustawienie odpowiedniego pinu kierunku. Jednak do sterowania silnikiem musisz używać wartości bezwzględnej na pinie PWM.

```
define Drive SpeedA SpeedB Time
  digital pin 8 output LOW
  digital pin 9 output LOW
  if SpeedA > 0 then
    digital pin 12 output HIGH
  else
    digital pin 12 output LOW
  if SpeedB > 0 then
    digital pin 13 output HIGH
  else
    digital pin 13 output LOW
  pwm pin 3 output abs of SpeedA
  pwm pin 11 output abs of SpeedB
  wait Time seconds
```



Spróbujmy teraz poruszyć robotem przy użyciu naszych bloków.



Ostatni blok "Stop" znajduje się w pętli nieskończonej, aby zakończyć ruch.

Zmień wartości w bloku "Drive", prześlij program na Arduino UNO i sprawdź, jak szybko porusza się Twój robot.

Teraz jesteś gotowy/-a do korzystania z ARTIEbota w bardziej zaawansowanych projektach, włączając w to korzystanie z kamery HuskyLens.

## PODSUMOWANIE

Nie ma wątpliwości, że roboty są niesamowite. Kiedyś nawet mogą rządzić światem, ale miejmy nadzieję, że wtedy zlitują się nad swoimi biednymi miękkimi twórcami (czyli programistami robotów) i pomogą nam zbudować raj w kosmosie. To oczywiście żart, ale tylko trochę.





## Scenariusz lekcji 16

**Czas (minuty):**  
90

### Tematyka

Programowanie robota do śledzenia obiektów z wykorzystaniem śledzenia i rozpoznawania obiektów

### Cele

Nauka programowania robota do śledzenia obiektów

### Oczekiwane efekty

Umiejętność pisania programu dla robota do śledzenia obiektów

## Programowanie robota – śledzenie obiektu

Nauczyliśmy się, jak poruszać robotem w poprzedniej lekcji.

Sprawdźmy, czy uczniowie dobrze zrozumieli wszystko z poprzedniej lekcji i czy są gotowi na kolejny krok.

Teraz przesuniemy nasze roboty w kierunku określonego obiektu. Najpierw musimy wykryć ten obiekt i śledzić go. Śledzenie poruszającego się obiektu wymaga technologii wizyjnego śledzenia obiektów oraz ręcznego działania.

#### Przedstawienie celu głównego lekcji:

Jak przygotować robota do śledzenia obiektów?



<http://erasmus-artie.eu>





# CZĘŚĆ GŁÓWNA

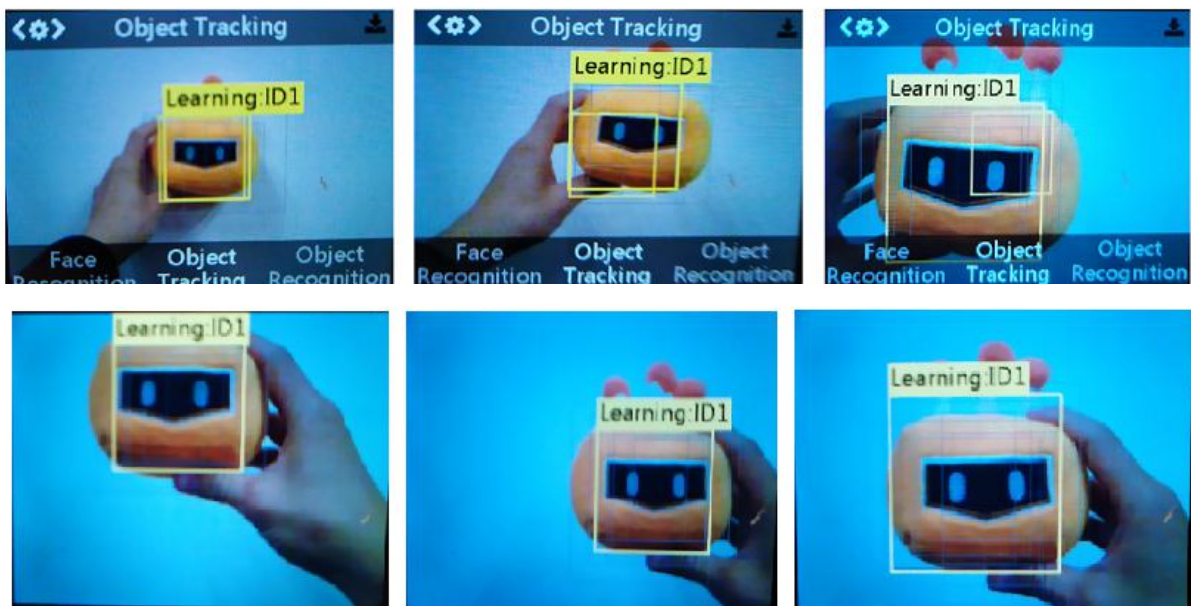
**Śledzenie obiektów** to ważne zadanie w zakresie komputerowego przetwarzania obrazu. Polega ono na ciągłym wnioskowaniu o stanie obiektów na sekwencjach wideo. Obraz jest rejestrowany przez pojedynczą kamerę, a informacje obrazowe są przekazywane do mikrokontrolera. Po analizie i przetwarzaniu obliczana jest względna pozycja poruszającego się obiektu. Jednocześnie sterowane jest obracanie robota z kamerą w czasie rzeczywistym w celu śledzenia obiektu. Podczas wykonywania funkcji śledzenia obiektów przez system, proces ten można podzielić na 4 kroki:

- rozpoznawanie obiektu,
- śledzenie obiektu,
- analiza ruchu obiektu,
- sterowanie robotem (lub innym systemem) za pomocą kamery.

## Rozpoznawanie obiektów - uczenie

Podłącz mikro:bit lub Arduino UNO z kamerą HuskyLens do komputera stacjonarnego lub laptopa. Skieruj kamerę HuskyLens na obiekt docelowy, dostosowując odległość, aż obiekt znajdzie się w pomarańczowym prostokącie na środku ekranu. Akceptowalne jest również, jeśli tylko część obiektu znajduje się wewnątrz prostokąta, ale posiada wyraźne cechy.

Następnie przytrzymaj "przycisk nauki" przez dłuższy czas, aby nauczyć kamerę obiektu z różnych kątów i odległości. W trakcie procesu nauki na ekranie pojawi się pomarańczowy prostokąt z napisem "Nauka: ID1"



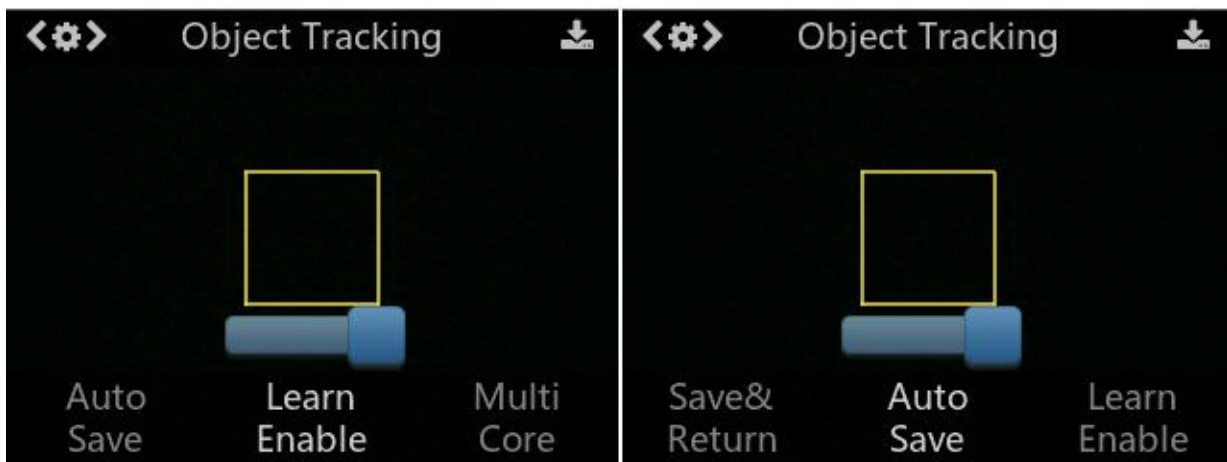


Kiedy HuskyLens jest w stanie śledzić obiekt z różnych kątów i odległości, puść przycisk "nauki", aby zakończyć proces uczenia. Jeśli na środku ekranu nie ma pomarańczowego prostokąta, oznacza to, że HuskyLens już nauczył się obiektu. Jeśli chcesz śledzić inny obiekt, wybierz opcję "Zapomnij nauczony obiekt" i rozpocznij naukę od nowa.

W ramach funkcji śledzenia obiektów, HuskyLens może nadal uczyć się, czyli dopóki kamera widzi nauczony obiekt, będzie kontynuować naukę aktualnego stanu obiektu, co ułatwia śledzenie obiektów dynamicznych. Metoda działania: Przytrzymaj przycisk funkcji, aby wejść w ustawienia parametrów funkcji śledzenia obiektu.

**f**

Przesuń przycisk funkcji w prawo, aby wybrać opcję "Learn Enable", następnie krótko naciśnij przycisk funkcji i przekręć go w prawo, aby włączyć opcję "Learn Enable". Ikona kwadratu na pasku postępu powinna przesunąć się w prawo. Następnie krótko naciśnij przycisk funkcji, aby potwierdzić ten parametr.



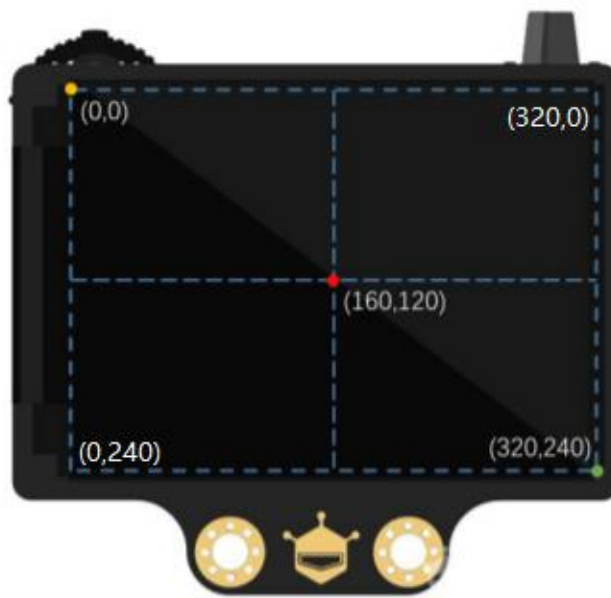


Po ponownym uruchomieniu HuskyLens, domyślnie nie zapisuje się ostatnio nauczony obiekt, jednak możesz włączyć przełącznik, aby automatycznie zapisywać modele.

Metoda działania: taka sama jak wcześniej, po wejściu w ustawienia parametrów, przełącz opcję "Auto Save" na ON. W ten sposób wystarczy nauczyć obiekt raz. Po ponownym uruchomieniu kamery, obiekt nauczony wcześniej zostanie automatycznie zapisany.

### Śledzenia obiektu

Rozdzielczość ekranu czujnika HuskyLens wynosi 320\*240, jak pokazano na poniższym rysunku



Wartości współrzędnych otrzymane przez program również mieszczą się w tym zakresie. Na przykład, jeśli otrzymane wartości współrzędnych to (160, 120), oznacza to, że śledzony obiekt znajduje się na środku ekranu.

"Współrzędne X" i "Współrzędne Y" odnoszą się do pozycji środka prostokąta na ekranie.

"Szerokość obiektu" i "Wysokość obiektu" odnoszą się do rozmiaru ramki. W ramach funkcji śledzenia obiektów, ramka jest kwadratowa, więc szerokość i wysokość są sobie równe.

### Przetestuj śledzenie obiektów – opcja 1 (Maqueen Plus/HuskyLens)

Otwórz swój Mind+ i załaduj rozszerzenia do pracy z kamerą Maqueen Plus i HuskyLens. Użyj tego kodu:



```

micro:bit starts
HuskyLens initialize pin until success
HuskyLens switch algorithm to Object tracking
set motor All stop

forever
  HuskyLens request data once and save into the result
  if HuskyLens check if ID 1 is learned from the result? then
    if HuskyLens check if ID 1 frame is on screen from the result? then
      serial output join "X: " HuskyLens get X center of ID 1 frame from the result in string , No-Wrap
      serial output join ", Y: " HuskyLens get Y center of ID 1 frame from the result in string , No-Wrap
      serial output join ", W: " HuskyLens get width of ID 1 frame from the result in string , No-Wrap
      serial output join ", H: " HuskyLens get height of ID 1 frame from the result in string , Wrap
    end if
  end if
  wait 1 seconds
end forever

```

Przejdź do **Sprawdzanie wyników**

### Przetestuj śledzenie obiektów – Opcja 2 (Arduino UNO/HuskyLens)

Otwórz swój Mind+ i załaduj rozszerzenia do pracy z Arduino UNO i kamerą HuskyLens. Użyj tego kodu z Arduino/HuskyLens:

```

Uno starts
HuskyLens initialize pin until success
HuskyLens switch algorithm to Object tracking
forever
  HuskyLens request data once and save into the result
  if HuskyLens check if ID 1 is learned from the result? then
    if HuskyLens check if ID 1 frame is on screen from the result? then
      serial output join "X: " HuskyLens get X center of the ID 1 No. 1 frame from the result in string , No-Wrap
      serial output join ", Y: " HuskyLens get Y center of the ID 1 No. 1 frame from the result in string , No-Wrap
      serial output join ", W: " HuskyLens get width of the ID 1 No. 1 frame from the result in string , No-Wrap
      serial output join ", H: " HuskyLens get height of the ID 1 No. 1 frame from the result in string , Wrap
    end if
  end if
  wait 0.5 seconds
end forever

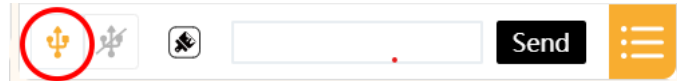
```



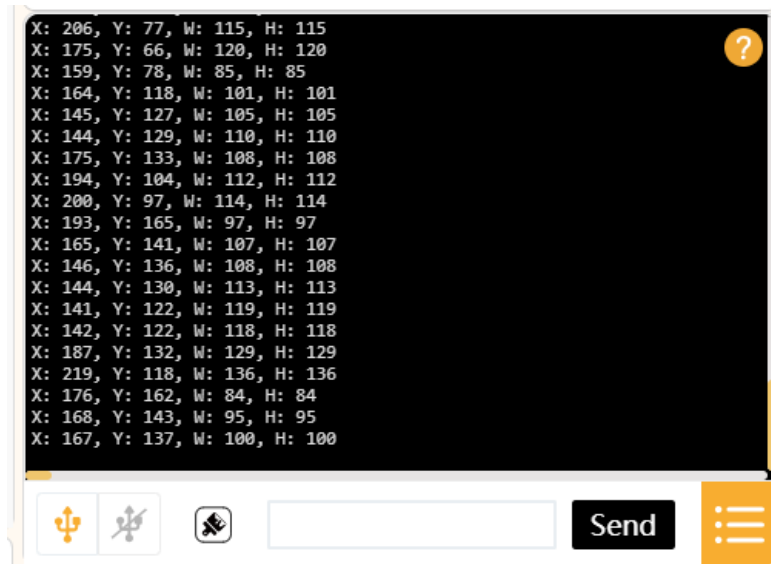


## Sprawdzanie wyników na monitorze szeregowym (obie opcje)

Otwórz monitor szeregowy, klikając ikonę USB w prawej dolnej części ekranu Mind+.

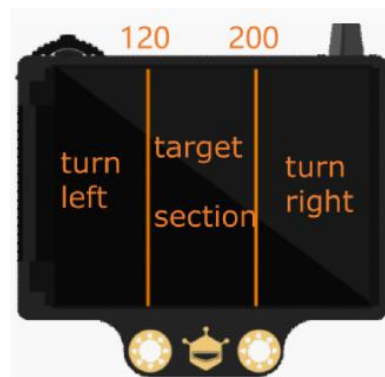


Spróbuj poruszać obiektem w lewo i prawo, aby obserwować numeryczną zmianę współrzędnej X środka. Przesuń obiekt w górę i w dół, aby obserwować numeryczną zmianę współrzędnej Y środka. Przesuń obiekt do przodu i do tyłu, aby obserwować numeryczną zmianę szerokości i wysokości.



## Analiza ruchu obiektów

Jak pokazano na poniższym obrazku, ekran jest podzielony na 3 sekcje zgodnie z osią X układu współrzędnych ekranu kamery, a środkowa sekcja jest naszą sekcją docelową.

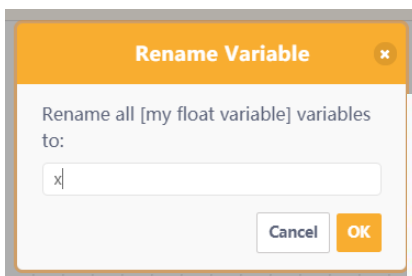




Gdy kamera ciągle wykrywa stan obiektu docelowego na obrazie, jeśli środek obiektu ma wartość X od 120 do 200, oznacza to, że obiekt jest w centrum pola widzenia i robot nie musi dostosowywać swojej pozycji; jeśli środek obiektu ma wartość X od 0 do 120, nasz robot musi dostosować się, skręcając w prawo; jeśli środek obiektu ma wartość X od 200 do 320, ARTIEbot musi skręcić w lewo, aby dostosować swoją pozycję.

Teraz nadszedł czas, aby napisać główną część kodu, która pozwoli skierować robota w stronę obiektu.

**W obu opcjach** - Zmień nazwę **mojej zmiennej typu float** na x. Kliknij prawym przyciskiem myszy na zmiennej -> Zmień nazwę zmiennej liczbowej.



```
micro:bit starts
HuskyLens initialize pin until success
HuskyLens switch algorithm to Object tracking
set motor All stop
forever
  HuskyLens request data once and save into the result
  if HuskyLens check if ID 1 is learned from the result? then
    if HuskyLens check if ID 1 frame is on screen from the result? then
      set x to HuskyLens get X center of ID 1 frame from the result
      if x >= 120 and x <= 200 then
        set motor All stop
      else if x > 0 and x < 120 then
        set motor Left move by 20 speed Forward
        set motor Right move by 80 speed Forward
      else if x > 200 and x < 320 then
        set motor Left move by 80 speed Forward
        set motor Right move by 20 speed Forward
    else
      set motor All stop
  else
    set motor All stop
```



**Opcja 1 - śledź obiekt za pomocą Maqueen Plus**

Użyj i skonfiguruj bloki jak na poniższym obrazku:



## Opcja 2 - śledź obiekt za pomocą ArtieBota

Najpierw zdefiniuj bloki Drive i Stop, jak opisano w poprzedniej lekcji (Programowanie robota)

```
define Drive SpeedA SpeedB Time
  digital pin 8 output LOW
  digital pin 9 output LOW
  if SpeedA > 0 then
    digital pin 12 output HIGH
  else
    digital pin 12 output LOW
  if SpeedB > 0 then
    digital pin 13 output HIGH
  else
    digital pin 13 output LOW
  pwm pin 3 output abs of SpeedA
  pwm pin 11 output abs of SpeedB
  wait Time seconds

define Stop
  pwm pin 3 output 0
  pwm pin 11 output 0
```





Użyj i skonfiguruj bloki jak na obrazku poniżej:

```

Uno starts
HuskyLens initialize pin until success
HuskyLens switch algorithm to Object tracking
forever
  HuskyLens request data once and save into the result
  if HuskyLens check if ID 1 is learned from the result? then
    if HuskyLens check if ID 1 frame is on screen from the result? then
      set x to HuskyLens get X center of the ID 1 No. 1 frame from the result
      if x >= 120 and x <= 200 then
        Stop
      else if x > 0 and x < 120 then
        Drive 20 80 0.25
      else if x > 200 and x < 320 then
        Drive 80 20 0.25
      else
        Stop
    else
      Stop
  else
    Stop

define Stop
  pwm pin 3 output 0
  pwm pin 11 output 0

define Drive SpeedA SpeedB Time
  digital pin 8 output LOW
  digital pin 9 output LOW
  if SpeedA > 0 then
    digital pin 12 output HIGH
  else
    digital pin 12 output LOW
  if SpeedB > 0 then
    digital pin 13 output HIGH
  else
    digital pin 13 output LOW
  pwm pin 3 output abs of SpeedA
  pwm pin 11 output abs of SpeedB
  wait Time seconds
  
```



<http://erasmus-artie.eu>

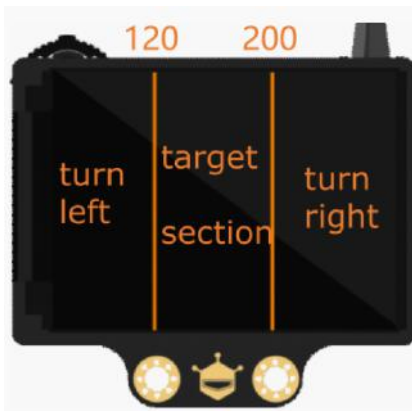
### OBIE opcje - Sprawdź, jak to działa.

Prześlij program do swojego robota.

Dokonaj ewentualnych dostosowań prędkości MotorA lub MotorB, jeśli jest to konieczne.

Gdy pole zidentyfikowanego obiektu znajduje się w centrum ekranu, robot zatrzymuje się.

Gdy pole znajduje się po lewej lub prawej stronie ekranu, robot automatycznie dostosowuje swoje położenie w lewo lub w prawo, aż pole zostanie umieszczone w docelowym obszarze ekranu.







## Śledź obiekt

Udało nam się skierować nasze roboty w stronę obiektu, ale wciąż nie śledzą go. Aby to osiągnąć, musimy wykryć rozmiar obiektu, aby dowiedzieć się, czy jest duży (blisko kamery) czy mały (daleko od kamery).

Utwórz nową zmienną (liczbową) i nazwij ją *h*. Będzie przechowywać wysokość śledzonego obiektu. Jeśli wysokość obiektu mieści się w zakresie od 60 do 100, robot będzie utrzymywał obecną pozycję. Jeśli jest mniejsza niż 60, oznacza to, że jest daleko i robot powinien jechać do przodu. Jeśli jest wyższa niż 100, robot powinien się cofać.

### Opcja 1 - Kod dla Maqueen Plus

```
micro:bit starts
HuskyLens initialize pin until success
HuskyLens switch algorithm to Object tracking
set motor All stop

forever
  HuskyLens request data once and save into the result
  if HuskyLens check if ID 1 is learned from the result? then
    if HuskyLens check if ID 1 frame is on screen from the result? then
      set x to HuskyLens get X center of the ID 1 No. 1 frame from the result
      set h to HuskyLens get height of the ID 1 No. 1 frame from the result
      if x >= 120 and x <= 200 and h >= 60 and h <= 100 then
        set motor All stop
      else if x > 0 and x < 120 then
        set motor Left move by 20 speed Forward
        set motor Right move by 80 speed Forward
      else if x > 200 and x < 320 then
        set motor Right move by 80 speed Forward
        set motor Left move by 20 speed Forward
      else if h > 0 and h < 60 then
        set motor All move by 80 speed Forward
      else if h > 100 and h < 240 then
        set motor All move by 80 speed Backward
    else
      set motor All stop
```





## Opcja 2 – Kod dla Arduino (ArtieBot):

```

Uno starts
HuskyLens initialize pin until success
HuskyLens switch algorithm to Object tracking
forever
  HuskyLens request data once and save into the result
  if HuskyLens check if ID 1 is learned from the result? then
    if HuskyLens check if ID 1 frame is on screen from the result? then
      set x to HuskyLens get X center of the ID 1 No. 1 frame from the result
      set h to HuskyLens get height of the ID 1 No. 1 frame from the result
      if x >= 120 and x <= 200 and h >= 60 and h <= 100 then
        Stop
      else if x > 0 and x < 120 then
        Drive 20 80 0.25
      else if x > 200 and x < 320 then
        Drive 80 20 0.25
      else if h > 0 and h < 60 then
        Drive 80 80 0.25
      else if h > 100 and h < 240 then
        Drive -80 -80 0.25
      else
        Stop
    else
      Stop
  else
    Stop

define Drive SpeedA SpeedB Time
  digital pin 8 output LOW
  digital pin 9 output LOW
  if SpeedA > 0 then
    digital pin 12 output HIGH
  else
    digital pin 12 output LOW
  if SpeedB > 0 then
    digital pin 13 output HIGH
  else
    digital pin 13 output LOW
  pwm pin 3 output abs of SpeedA
  pwm pin 11 output abs of SpeedB
  wait Time seconds

define Stop
  pwm pin 3 output 0
  pwm pin 11 output 0

```



<http://erasmus-artie.eu>

### Obydwie opcje - Sprawdź, jak to działa

Prześlij program do mikro:bita/Arduino UNO, aby sprawdzić, jak to działa.

Dokonaj poprawek, aby ruch był płynny, dostosowując prędkość silników i czas jazdy.

Po zakończeniu uczenia obiektu przez HuskyLens, roboty automatycznie będą śledzić obiekt i poruszać się do przodu, do tyłu, w lewo i w prawo, utrzymując pole obiektu w centrum ekranu i w odpowiedniej odległości.

Gdy robot jest używany jako robot śledzący, można go zaprogramować do lokalizowania dowolnego celu za pomocą kamery HuskyLens. Oznacza to, że można przekształcić ten projekt w śledzącego człowieka robota, który będzie podążał za osobami.



Śledzenie obiektów to zadanie polegające na wykrywaniu początkowego zestawu obiektów, nadawaniu każdemu z nich unikalnego identyfikatora, a następnie śledzeniu tych obiektów w kolejnych klatkach wideo, utrzymując przypisanie identyfikatora. Metody najnowocześniejsze polegają na łączeniu danych z kamer RGB i kamer opartych na zdarzeniach w celu uzyskania bardziej niezawodnego śledzenia obiektów.

Teraz rozumiemy podstawowe zasady śledzenia obiektów i jak korzystać z możliwości śledzenia obiektów w HuskyLens.

Wiemy także, jak używać HuskyLens, aby nasz robot podążał za wybranym celem.



## **PODSUMOWANIE**

Aby skutecznie śledzić obiekt, musimy najpierw go wykryć, a następnie śledzić. Śledzenie poruszającego się obiektu wymaga technologii wizualnego śledzenia obiektów, a także ręcznego sterowania.





## Scenariusz lekcji 17

Czas (minuty):  
90

### Tematyka

Programowanie robota do śledzenia linii za pomocą kamery światła widzialnego, kamery termowizyjnej na podczerwień i innych instrumentów wykrywających

### Cele

Nauka programowania robota do śledzenia linii

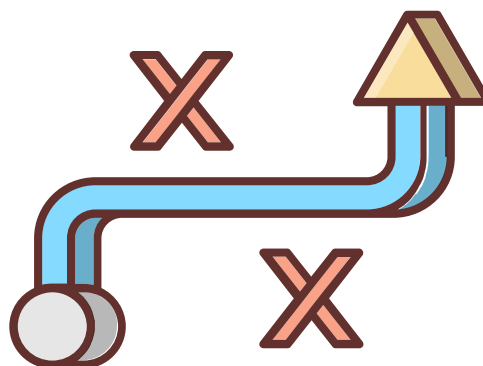
### Oczekiwane efekty

Umiejętność kodowania robota do śledzenia linii

## Programing a robot - line following

Nauczyliśmy się na poprzedniej lekcji programować roboty, aby mogły śledzić i podążać za obiektami. Upewnij się, że Twoi uczniowie zrozumieli wszystko z poprzedniej lekcji i są gotowi na kolejny krok. Teraz nauczmy się śledzić i podążać za linią.

Przedstawienie celu głównego lekcji:  
Jak przygotować robota do podążania za linią?



<http://erasmus-artie.eu>



# CZĘŚĆ GŁÓWNA

Śledzenie linii odnosi się do procesu poruszania się obiektu wzdłuż określonej trasy. W pełni funkcjonalny robot do śledzenia linii wykorzystuje mobilnego robota jako nośnik, kamerę widzialnego światła, termowizor podczerwieni i inne instrumenty detekcyjne jako system obciążenia, wielopolową fuzję informacji z dziedziny widzenia maszynowego, pola elektromagnetycznego, GPS i GIS jako system nawigacji do autonomicznego poruszania się i śledzenia robota, oraz wbudowany komputer jako platformę do rozwoju oprogramowania i sprzętu dla systemu sterowania.

## Czujniki śledzenia linii

Type	Infrared Line Tracking Sensor	Visual Sensor
Comparison Items		
Cost	Low	High
Range of Vision	The sensor has a small range of view; it needs to be close to the ground .	The range of vision is wide, and the movement state can be adjusted in advance according to the line changes.
Environment Adaptation	When the usage environment is changed, the sensitivity of the sensor needs to be adjusted, and the adjustment process is complicated.	When the use environment is changed, only the lines need to be relearned, and the operation is simple.
Map Adaptation	Generally, only suitable for simple maps with clear background lines, black and white lines or solid lines	Suitable for maps with clear background lines, multi-colour lines, solid lines, dotted lines and other complex conditions.



<http://erasmus-artie.eu>

## Algorytm śledzenia linii HuskyLens

Funkcja śledzenia linii w HuskyLens opiera się na Pixy, projekcie open-source Uniwersytetu Carnegie Mellon. Algorytm Pixy jest w stanie rozpoznawać kolory na zdjęciach. Jego podstawową ideą jest wykorzystanie przestrzeni kolorów do usunięcia tła, które nie jest interesujące dla użytkowników, oraz wyodrębnienie pierwszego planu (takiego jak linie).

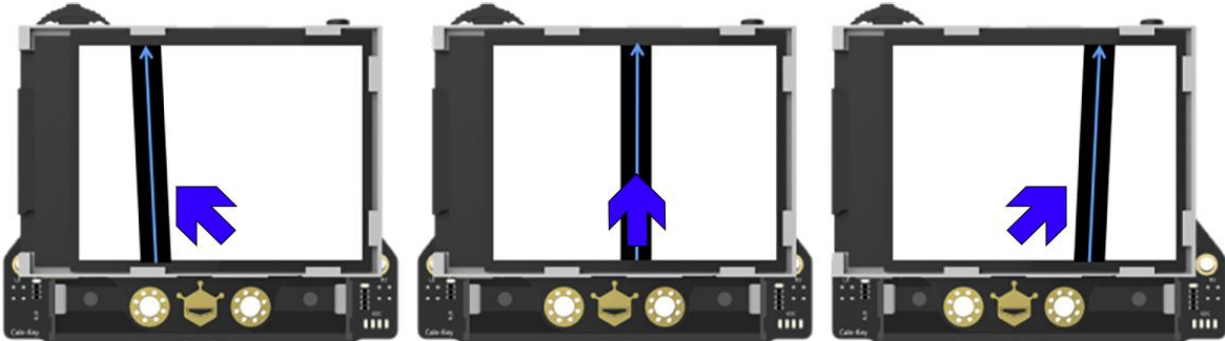


Jak robot może śledzić czarną linię na mapie śledzenia (która ma czarne linie na białym tle)? W rzeczywistości, musimy jedynie znać względną pozycję ARTIEbota względem czarnej linii.



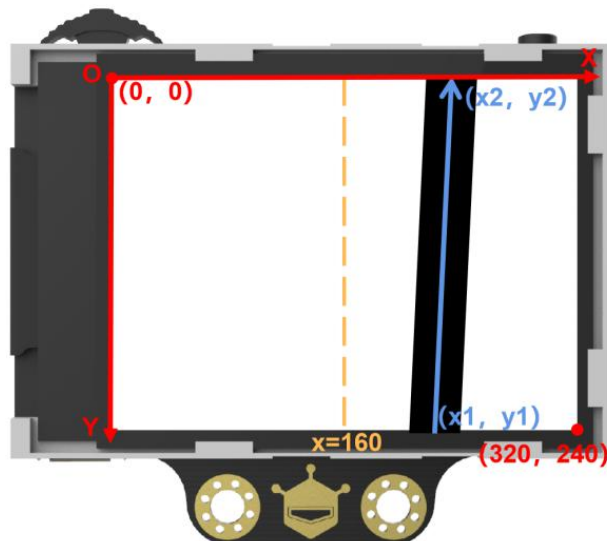
W zasadzie mamy trzy następujące sytuacje:

1. Gdy robot jest po prawej stronie czarnej linii, powinien skrócić w lewo
2. Gdy robot jest na środku, wyrównany z czarną linią, powinien jechać prosto
3. Gdy robot jest po lewej stronie czarnej linii, powinien skrócić w prawo



## Implementacja

Rozdzielczość ekranu HuskyLens wynosi  $320 \times 240$ . Punkt  $O$  w lewym górnym rogu ekranu jest początkiem układu współrzędnych ekranu  $(0, 0)$ , prawo to kierunek dodatni osi  $X$ , a w dół to kierunek dodatni osi  $Y$ . Oznacza to, że współrzędne w prawym dolnym rogu ekranu to  $(320, 240)$ . Przerwana pomarańczowa linia na obrazku to oś środkowa ekranu, a wartość  $x$  tej linii wynosi  $160$ . Czarna linia na poniższym obrazku to linia mapy "widziana" przez kamerę HuskyLens. Niebieska strzałka to kierunek linii obliczony przez HuskyLens. Współrzędne punktu początkowego niebieskiej strzałki to  $(x_1, y_1)$ , a współrzędne punktu końcowego to  $(x_2, y_2)$ .





W uproszczeniu, potrzebujemy tylko wiedzieć, jaki jest punkt początkowy (x1) niebieskiej strzałki względem osi środkowej ( $x=160$ ), aby zaimplementować śledzenie linii.

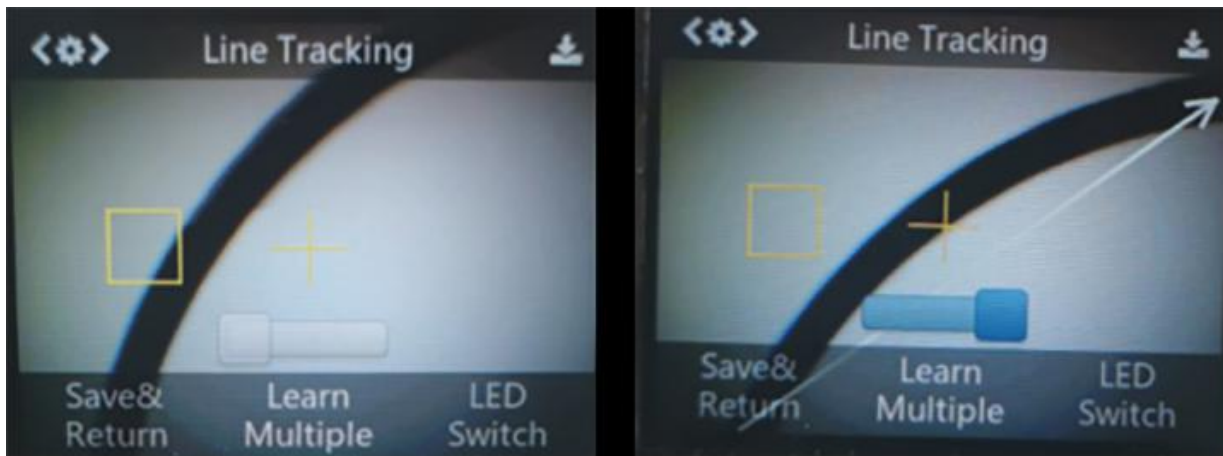
Ta funkcja może śledzić linie określonych kolorów i przewidywać ich ścieżkę. Domyślne ustawienie to śledzenie linii jednego koloru, a w tym projekcie będzie używane **śledzenie linii jednego koloru**.

### Ustawienia kamery

**Krok 1:** Przekręć przycisk funkcji w lewo lub w prawo, aż na górze ekranu pojawi się napis "Śledzenie linii".

**Krok 2:** Przytrzymaj przycisk funkcji, aby wejść w ustawienia parametrów funkcji śledzenia linii.

**Krok 3:** Przekręć przycisk funkcji w prawo lub w lewo, aż zostanie wybrana opcja "Ucz się wielokrotnie". Następnie krótko naciśnij przycisk funkcji, a następnie przekręć go w lewo, aby wyłączyć przełącznik "Ucz się wielokrotnie", czyli kwadratowy symbol na pasku postępu zostanie przekręcony w lewo. Następnie krótko naciśnij przycisk funkcji, aby zatwierdzić ten parametr.

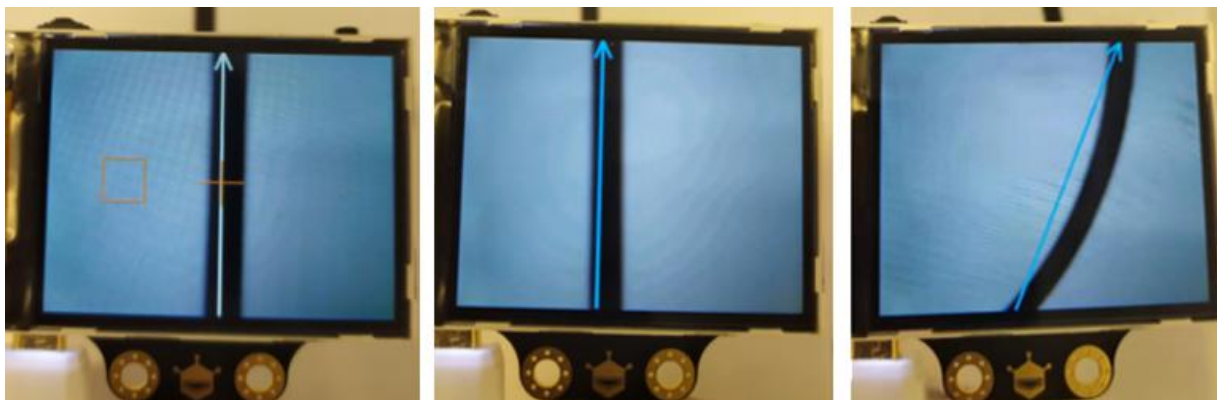


**Krok 4:** Możesz również włączyć diodę LED, ustawiając przełącznik "LED". Jest to bardzo przydatne w ciemnym otoczeniu. Korzystając z powyższej metody, włącz przełącznik "LED".

**Krok 5:** Przekręć przycisk funkcji w lewo, aż zostanie wybrana opcja "Zapisz i powrót", a następnie krótko naciśnij przycisk funkcji, aby zapisać parametry. Automatycznie zostanie wykonany powrót.

### Nauka i śledzenie

**Nauka linii:** Wskaż symbol "+" na linii, a następnie skieruj pomarańczowe pole na tło. Zaleca się, aby na ekranie nie było innych linii. Staraj się utrzymać HuskyLens równoległe do linii docelowej; HuskyLens automatycznie wykryje linię, a na ekranie pojawi się biała strzałka. Następnie krótko naciśnij przycisk "nauki", a biała strzałka zmieni się w niebieską strzałkę.



- Podczas uczenia linii musimy dostosować pozycję HuskyLens, aby była równoległa do linii.
- HuskyLens może uczyć się linii o dowolnym kolorze, który ma wyraźny kontrast kolorystyczny do tła, ale linia ta musi być jednokolorowa, aby proces śledzenia linii był stabilny.
- HuskyLens może uczyć się i śledzić wiele linii o różnych kolorach, ale wszystkie te linie muszą być jednokolorowe i w widocznym kontraście do tła. W tym przykładzie użyjemy czarnej linii (czarna taśma izolacyjna na białym tle, takim jak papier lub biała płyta MDF).
- Widoczność linii zależy w dużej mierze od oświetlenia otoczenia. Podczas śledzenia linii staraj się utrzymać oświetlenie otoczenia jak najstabilniejsze i użyj diody LED HuskyLens, jeśli to konieczne.



### Otwórz Mind+ i załaduj rozszerzenia

Otwórz Mind+ i załaduj rozszerzenia. **Gdy korzystasz z robota micro:Maqueen Plus, upewnij się, że wybierasz odpowiednią wersję (V1 lub V2).**

Zmień nazwę **mojej zmiennej typu float na x**. Kliknij prawym przyciskiem myszy na zmienną -> Zmień nazwę zmiennej numerycznej.

### Algorytm:

- Odczytaj wartość  $x_1$  (lub  $x$  początkowe) z funkcji śledzenia linii HuskyLens - jest to początkowy punkt niebieskiej strzałki
- Jeśli czarna linia znajduje się po lewej stronie ekranu ( $x_1 < 150$ ), robot powinien skręcić w lewo
- Jeśli czarna linia znajduje się po prawej stronie ekranu ( $x_1 > 170$ ), robot powinien skręcić w prawo
- Jeśli czarna linia znajduje się w środku ekranu ( $150 \leq x_1 \leq 170$ ), robot powinien jechać prosto





## Opcja 1 - Śledzenie linii z Maqueen Plus

Użyj tego kodu:

```
micro:bit starts
HuskyLens initialize pin until success
HuskyLens switch algorithm to Line tracking
forever
  HuskyLens request data once and save into the result
  if HuskyLens check if ID 1 is learned from the result? then
    if HuskyLens check if ID 1 arrow is on screen from the result? then
      set my float variable to HuskyLens get X beginning of ID 1 arrow from the result
      if my float variable >= 150 and my float variable <= 170 then
        set motor All move by 60 speed Forward
      else if my float variable < 150 then
        set motor Right move by 90 speed Forward
        set motor Left move by 30 speed Forward
      else if my float variable > 170 then
        set motor Left move by 90 speed Forward
        set motor Right move by 30 speed Forward
      else
        set motor All stop
```





## Opcja 2 - Śledzenie linii z użyciem Arduino (ArtieBot)

Najpierw zdefiniuj bloki Drive i Stop, jak opisano w lekcji Programowanie robota.

Dobrym pomysłem jest zebranie często używanych skryptów i łatwe przenoszenie ich między projektami.

The image shows two Scratch code blocks for an Arduino Uno. The 'define Drive' block (pink) contains the following logic: set digital pins 8 and 9 to LOW; if SpeedA > 0, set pin 12 to HIGH, otherwise LOW; if SpeedB > 0, set pin 13 to HIGH, otherwise LOW; set PWM pins 3 and 11 to the absolute values of SpeedA and SpeedB respectively; and wait for 'Time' seconds. The 'define Stop' block (pink) contains: set PWM pins 3 and 11 to 0; and set digital pins 8 and 9 to HIGH.





```

Uno starts
HuskyLens initialize pin until success
HuskyLens switch algorithm to Line tracking
define Stop
  pwm pin 3 output 0
  pwm pin 11 output 0
forever
  HuskyLens request data once and save into the result
  if HuskyLens check if ID 1 is learned from the result? then
    if HuskyLens check if ID 1 arrow is on screen from the result? then
      set x to HuskyLens get X beginning of ID 1 arrow from the result
      if x >= 150 and x <= 170 then
        Drive 90 90 0.25
      else if x < 150 then -
        Drive 40 90 0.25
      else if x > 170 then -
        Drive 90 40 0.25
      else
        Stop
  
```

f

```

define Drive SpeedA SpeedB Time
  digital pin 8 output LOW
  digital pin 9 output LOW
  if SpeedA > 0 then
    digital pin 12 output HIGH
  else
    digital pin 12 output LOW
  if SpeedB > 0 then
    digital pin 13 output HIGH
  else
    digital pin 13 output LOW
  pwm pin 3 output abs of SpeedA
  pwm pin 11 output abs of SpeedB
  wait Time seconds
  
```

<http://erasmus-artie.eu>



## Obie opcje – przetestuj swój algorytm

Przygotuj białą powierzchnię (papier lub MDF) i za pomocą taśmy izolacyjnej stwórz linię.



Prześlij program na swojego robota, umieść go gdzieś na linii i zobacz, jak się porusza. Czy śledzi linię? Czy gubi linię? Czy jest jakiś sposób na radzenie sobie z tym?

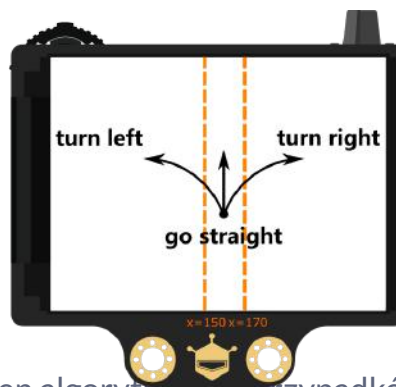
Oto kilka wskazówek, które mogą Ci pomóc:

- Spróbuj zmienić kąt kamery.
- Spróbuj uzyskać **początek X** strzałki dla ID 1 zamiast końca X.

Dodaj kod zabezpieczający, korzystając z następującej wskazówki:

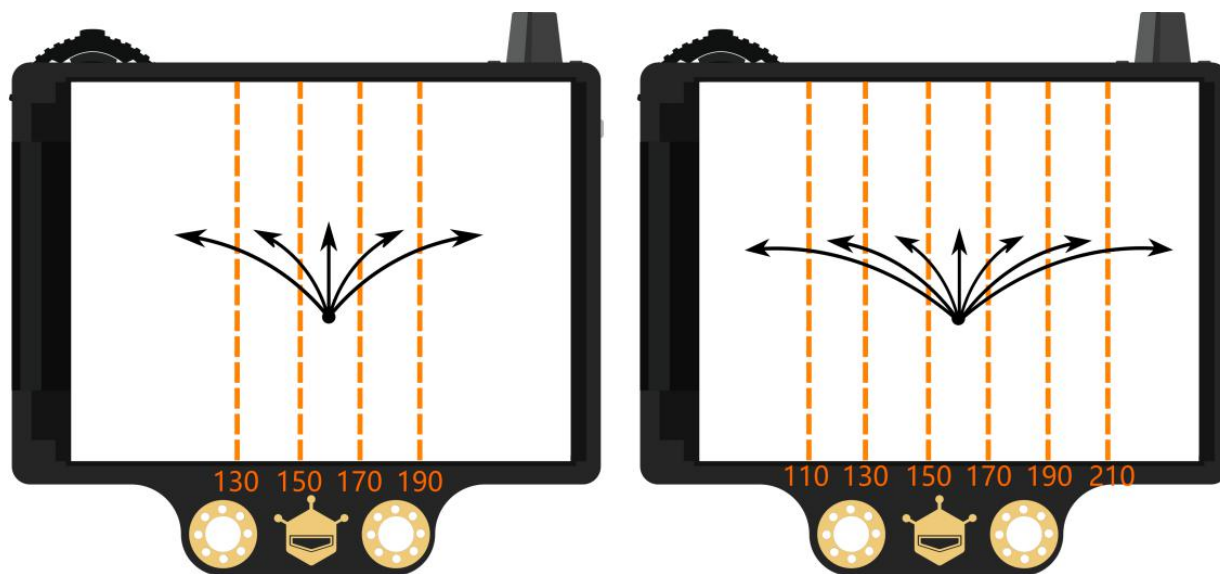
- Jeśli linia jest obecna i wykryto strzałkę, przejdź do kodu obsługi strzałki i zapisz zawartość zmiennej `x` do nowej zmiennej o nazwie **last x**.
- Jeśli linia zostanie utracona i nie zostanie wykryta strzałka, użyj **last x** zamiast `x`, aby wrócić do linii.

I na koniec najważniejsze - wprowadź korekty, aby ruch był płynny, poprzez dostosowanie prędkości silników i czasu jazdy. Algorytm ten analizuje pozycję `x`; posiada 3 możliwości jak na zdjęciu poniżej.



Czy możesz zoptymalizować ten algorytm dla 5 przypadków lub nawet 7, aby ruch był płynniejszy?

Spójrz na obrazki poniżej, aby dowiedzieć się, jak to zrobić.



Podsumowanie zdobytej wiedzy:

1. Zrozumienie głównych zasad śledzenia linii
2. Umiejętność korzystania z funkcji śledzenia linii w HuskyLens
3. Zastosowanie i optymalizacja algorytmu śledzenia linii

Jak sama nazwa wskazuje, robot śledzący linię to zautomatyzowany pojazd, który podąża za widoczną linią umieszczoną na powierzchni. Ta linia stanowi ścieżkę, po której porusza się robot śledzący linię. Zazwyczaj wykorzystuje się czarną linię na białej powierzchni, choć można ją również dostosować jako białą linię na czarnej powierzchni.

Zazwyczaj początkujący i studenci mają pierwsze doświadczenia z robotami właśnie tego typu. W przemyśle wykorzystuje się olbrzymie roboty śledzące linie do wspomaganie zautomatyzowanego procesu produkcji. Są one również wykorzystywane w celach wojskowych, pomocy ludziom, usług dostarczania, itp. Teraz rozumiemy główne zasady śledzenia linii i umiemy stosować funkcję śledzenia linii w HuskyLens. Wiemy także, jak stosować i optymalizować algorytm śledzenia linii.

Omów z uczniami różnice i podobieństwa między śledzeniem i podążaniem za obiektami a śledzeniem i podążaniem za linią.

## PODSUMOWANIE

Poznaliśmy główne zasady śledzenia linii za pomocą funkcji HuskyLens i zoptymalizowaliśmy algorytm śledzenia linii.