Tu możesz kupić ai robot:

Zużyty sprzęt elektryczny i elektroniczny (WEEE)

OSTRZEŻENIA!

1. Produkt nie jest przeznaczony dla dzieci w wieku poniżej 3 lat. Zawiera małe elementy – ryzyko zadławienia.
2. Do użytku pod bezpośrednim nadzorem osoby dorosłej.
3. Należy zachować opakowanie lub/i instrukcję. Zawierają one ważne informacje mogące być przydatne w przyszłości.
4. Użytkowanie niezgodne z zaleceniami zwalnia producenta od odpowiedzialności za ewentualne szkody.

nowa szkoła
ul. POW 25, 90-248 Łódź,
www.nowaszkola.com
tel. (42) 630 17 28,
(42) 630 04 88, fax: (42) 632 73 28

nowa szkoła

ai·robot

SCENARIUSZE DLA SZKOŁY PODSTAWOWEJ KLASY VI–VIII

W przypadku programowania za pomocą komend głosowych **wymagane jest oprogramowanie XIAOZI AI** (po jednorazowym wgraniu oprogramowania nie ma konieczności ponawiania tej czynności). Scenariusz sprawdzi się również przy korzystaniu ze standardowego oprogramowania i kodowania za pomocą bloków SCRATCH lub PYTHON. Robot może być programowany z pomocą platformy online lub aplikacji pobranej na urządzenie. Wersja przeglądarkowa: https://icreatecode.icrobot.com/editor.html

Elementy ai robota:
NS9401 – ai robot
NS9402 – Chwytak
NS9403 – Mata miasto przyszłości
NS9404 – Mata trawa
NS9405 – Mata suchościeralna
NS9401W – Edukacyjne karty pracy

Zestawy ai robota:
NS9400 – Zestaw programisty
NS9410 – Wyspa małego programisty
NS9411 – Wysoka wyspa programisty

Aplikacje do pobrania:

Windows

Android

nowa szkoła

ai·robot SCENARIUSZ 1

Inteligentne miasto
Robot jako system decyzyjny

Przedmiot: Informatyka / Technika.

Cel ogólny

Uczeń projektuje i realizuje algorytm sterujący zachowaniem robota na podstawie danych z czujników robota, wykorzystując instrukcje warunkowe i testując poprawność działania programu.

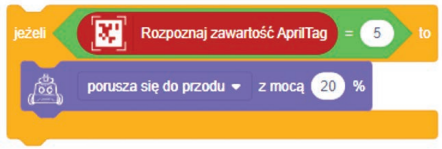
- Cele szczegółowe (uczeń):
  - opisuje algorytm działania robota reagującego na rozpoznawane kolory,
  - stosuje instrukcje warunkowe („jeśli – wtedy – w przeciwnym razie") w programie,
  - łączy informacje z czujników robota z działaniem elementów wykonawczych robota,
  - analizuje błędy w działaniu programu i wskazuje ich możliwe przyczyny,
  - modyfikuje program w celu poprawy skuteczności działania robota,
  - wyjaśnia, w jaki sposób podobne algorytmy są wykorzystywane w systemach automatyki i robotyki.

Podstawa programowa

- Informatyka (klasy IV–VIII):
  - Uczeń formułuje i zapisuje algorytmy, w tym sterowanie robotem lub obiektem na ekranie.
  - Uczeń projektuje i tworzy programy wykorzystujące instrukcje sekwencyjne, warunkowe i iteracyjne oraz zapisuje program sterujący robotem.
  - Uczeń testuje i poprawia programy zgodnie z przyjętymi założeniami.
  - Uczeń projektuje i testuje oprogramowanie sterujące rzeczywistym robotem.

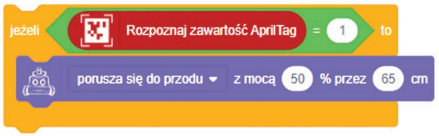
Etap 5: Zadanie rozszerzające (Dla szybszych grup)

- Znaki drogowe i prędkość: Uczniowie programują Tag nr 5 jako „Strefa zamieszkania" (robot jedzie prosto z 20% mocy silników)



Etap 6: Wariant zaawansowany

- Uczniowie programują AprilTagi w taki sposób, aby ai robot przemieszczał się po długości całych prostych odcinków zamiast za każdym razem o 13 cm (jedno pole)
- Przykład dla tej samej trasy z pola C7 do pola C1:
- Zamiast co każde pole pokazywać dla robota apriltag z komendą (jedź prosto 13cm) możemy zaprogramować robota tak, aby jechał całą długość danego odcinka.



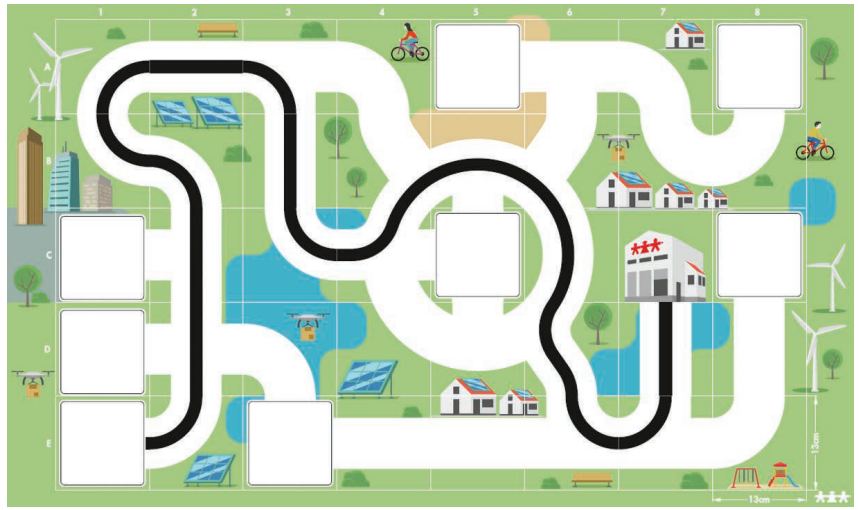
Podsumowanie:

- Nauczyciel podsumowuje lekcję, zadając pytania:
  - Jakie zalety ma nawigowanie maszynami za pomocą znaczników wizualnych, a jakie wady zauważyliście podczas dzisiejszych testów (np. wpływ światła, zasłonięcie tagu)?
  - Z jakiej największej odległości ai robot jest w stanie odczytać AprilTag?

- Technika (klasy IV–VIII):
  - wyjaśnia zasadę współpracy kamery AI z pozostałymi elementami układu technicznego,
  - stosuje algorytm sterowania jako sposób rozwiązywania prostego problemu technicznego,
  - planuje działanie robota jako sekwencję czynności zależnych od danych wejściowych,
  - testuje działanie skonstruowanego układu technicznego i ocenia jego poprawność,
  - wskazuje przykłady wykorzystania automatyki i robotyki w rozwiązaniach technicznych spotykanych w życiu codziennym,
  - przestrzega zasad bezpiecznej i odpowiedzialnej pracy z urządzeniami technicznymi.

Środki dydaktyczne:

- Robot edukacyjny ai robot z wbudowaną kamerą AI – gotowy do rozpoznawania kolorów i podejmowania decyzji.
- Komputer lub tablet z oprogramowaniem do programowania robota (Scratch oraz Python).
- Projektor / monitor interaktywny – do prezentacji założeń zadania, algorytmów i wyników pracy.
- Mata edukacyjna przedstawiająca środowisko zadaniowe, „inteligentnego miasta".



- Drukarka 3D do drukowania akcesoryjnych przeszkód dla robota.
- Karty pracy do ai robota

Przebieg zajęć

Wprowadzenie do lekcji

1. Tematy do omówienia

Cel etapu: Uczeń rozumie, jak robot w inteligentnym mieście może wykorzystywać kamerę AI do podejmowania decyzji.

A. Inteligentne miasto – co to takiego?

- Miasto przyszłości, w którym roboty i czujniki współpracują, aby transport i logistyka były efektywne.
- Przykłady w realnym życiu: inteligentne ścieżki rowerowe, automatyczne sortownie odpadów, autonomiczne pojazdy.
- Uczniowie rozumieją, że robot w mieście musi reagować na sygnały i znaki, np. światła, przeszkody, cele.

B. Jak robot „widzi" świat?

- Kamera ai robota wykrywa kolory i symbole.
- Robot analizuje obraz w czasie rzeczywistym i podejmuje decyzję: jedź prosto, skręć, zatrzymaj się.
- Uczniowie poznają pojęcia wejście ➔ przetwarzanie ➔ wyjście w robotyce.

C. Algorytmy decyzyjne w praktyce.

- Wprowadzenie do prostych algorytmów:
  - „jeżeli robot widzi czerwony kolor ➔ zatrzymaj się",
  - „jeżeli robot widzi zielony kolor ➔ jedź dalej"
- Wspólne omówienie przykładów na macie i kartach decyzyjnych.

2. Pokaz praktyczny

Cel etapu: Uczeń wie, że kamera AI zamienia obraz na dane, które można wykorzystać w programie.

Kamera w robocie działa trochę jak bardzo uważne oko, które potrafi rozróżniać kolory. Oto jak to wygląda krok po kroku, w prosty sposób:

1. Oświetlenie obiektu

Czujnik (albo kamera) najpierw oświetla obiekt – często ma do tego własną diodę LED. Dzięki temu kolor jest lepiej widoczny i mniej zależny od światła w sali.

2. Odbicie światła

Każdy kolor odbija światło inaczej. Na przykład czerwony obiekt odbija dużo czerwonego światła, a mało niebieskiego i zielonego.

3. Odczyt składowych koloru

Czujnik mierzy, ile światła czerwonego (R), zielonego (G) i niebieskiego (B) do niego wraca. Te trzy wartości tworzą razem konkretny kolor.

- Ważne: Nauczyciel przypomina, by w przypadku wykrycia tagu dodać krótką pauzę (np. Czekaj 1 sekundę) po wykonaniu ruchu, aby robot nie odczytał tej samej karty podwójnie w ułamku sekundy.

Etap 3: Projektowanie trasy i rozstawienie znaków

- Nauczyciel określa misję: „Jesteście inżynierami ruchu. Musicie poprowadzić autonomiczny autobus z pola C7 do szkoły na polu C1".
  - Uczniowie analizują drogę w „Inteligentnym mieście przyszłości".
  - Uczniowie przygotowują odpowiednie karty apriltag i pokazują je robotowi w odpowiednich momentach. Kamera robota potrafi odczytać karty z dużej odległości, dlatego nie mogą być one na stałe postawione na trasie robota.

Etap 4: Autonomiczny przejazd

- Uczniowie ustawiają włączonego robota na starcie i uruchamiają program.
  - Uczniowie pokazują robotowi karty AprilTag z odpowiednimi komendami, a robot wykonuje zapisaną w kodzie czynność (np. obraca się w lewo na skrzyżowaniu).
  - Jeśli robot się zgubił lub pojechał w złą stronę (np. pokazano złą kartę, błąd w skrypcie), uczniowie analizują problem i wykonują tzw. debugowanie.

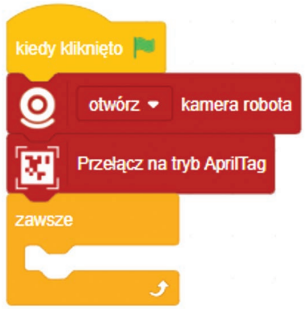
Etap praktyczny – praca z ai robotem i matą

Cel etapu:

Uczeń tworzy program, który zmienia kamerę w czujnik decyzyjny i programuje konkretne czynności dla danego apriltagu.

Etap 1: Inicjalizacja kamery i pętli głównej

- Uczniowie w środowisku programistycznym:
  - wstawiają blok otwierający kamerę robota i uruchamiający tryb wykrywania znaczników (Tag Recognition),
  - dodają pętlę „Zawsze" (niezbędną, aby robot nieustannie skanował otoczenie).



Etap 2: Programowanie instrukcji warunkowych

- Uczniowie budują wewnątrz pętli zestaw warunków sterujących:
  - Jeżeli wykryto Tag 1 -> jedź prosto o 13 cm.
  - Jeżeli wykryto Tag 2 -> skręć w prawo o 90 stopni... (i tak dalej dla ustalonych kodów).

4. Analiza danych

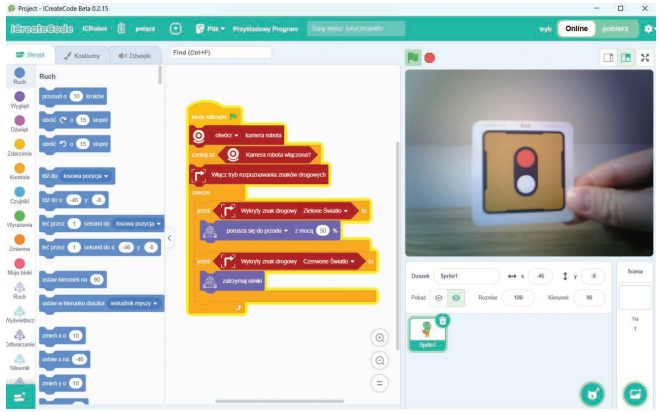
Robot porównuje odczytane wartości z zaprogramowanymi wcześniej wzorcami i sprawdza, jaki to kolor.

a) Nauczyciel pokazuje robotowi karty: Czernowa sygnalizacja (STOP), Zielona sygnalizacja (JEDŹ), Strzałka w prawo (SKRĘT).

b) Robot reaguje na kolory i symbole zgodnie z wcześniej wgranym programem.

c) Krótka dyskusja:

- Jak robot „zrozumiał", co ma zrobić?



- Dlaczego rozpoznawanie kolorów i symboli jest ważne w inteligentnych miastach?

3. Pytania do uczniów

Cel etapu: Uczeń dostrzega praktyczne zastosowanie omawianej technologii.

- Jakie sytuacje w realnym mieście mogłyby wymagać od robota podobnych decyzji?
- Co jeszcze robot mógłby wykrywać poza kolorami i symbolami? (np. przeszkody, ludzi, linie drogi).
- Jakie inne algorytmy moglibyśmy zastosować w naszym mieście przyszłości?

Etap praktyczny – praca z ai robotem

Cel etapu: Uczeń potrafi programować robota ai robota do reagowania na kolory i symbole, rozpoznawać warunki decyzyjne w algorytmie, testować i modyfikować program w zależności od wyników działania oraz łączyć programowanie z fizycznymi elementami na macie.

1: Przygotowanie stanowiska

- Mata rozłożona na wyspie programisty, lub podłodze.
- Karty rozmieszczone na trasie robota, w miejscach, gdzie będzie podejmował decyzje.
- Elementy 3D ustawione w wybranych punktach trasy – np. szlabany przy „strefach kontroli", sygnalizacja świetlna przy skrzyżowaniach.

2: Reagowanie na karty (wstęp)

- Połącz ai robot z komputerem, lub tabletem i otwórz program w Scratch lub Python.
- Umieść robota przed serią 3 kart: Sygnalizacja świetlna, strzałka w lewo, strzałka w prawo,

- Napisz prosty algorytm:
- czerwony ➔ zatrzymaj się,
- zielony ➔ jedź prosto,
- strzałka w lewo ➔ skręć w lewo,
- strzałka w lewo ➔ skręć w prawo.

Model algorytmu w Scratch:

- Uczeń stosuje algorytmy warunkowe do sterowania zachowaniem urządzeń w zależności od danych wejściowych (rozpoznany ID znacznika).
- Technika (klasy IV–VIII):
- Uczeń potrafi wykorzystać elementy automatyki i robotyki w praktycznych zadaniach transportowych.

Środki dydaktyczne

- ai robot,
- komputer, laptop lub tablet z aplikacją do programowania robota,
- mata edukacyjna „Inteligentne miasto przyszłości"
- Karty ze znacznikami AprilTag od 1 do 9 (zestaw kart dołączony do robota).

Przebieg zajęć

Wprowadzenie do lekcji

1. Jak maszyny czytają znaki? Co to jest AprilTag?

- **Cel etapu:** Uczeń dowiaduje się, jak zautomatyzowane pojazdy odnajdują się w przestrzeni.
- Nauczyciel wprowadza temat sztucznego widzenia (Computer Vision):
  - Ludzie czytają litery i znaki drogowe. Dla komputerów jest to jednak trudne i wymaga dużo mocy obliczeniowej.
  - Aby ułatwić maszynom pracę w fabrykach i magazynach (np. robotom układającym paczki), inżynierowie stworzyli AprilTagi – specjalne, uproszczone kody wizualne (podobne do QR kodów), które kamera robota potrafi rozpoznać błyskawicznie i z dużej odległości.
  - Nauczyciel pokazuje uczniom fizyczne karty z tagami i tłumaczy: „Dla nas to tylko zbiór czarnych i białych kwadratów, ale kamera robota widzi tu konkretną liczbę (ID)".

2. Ustalenie „Słownika Nawigacyjnego"

- Cel etapu: Klasa wspólnie przypisuje znaczenie do poszczególnych numerów ID.
- Aby zapanował porządek, nauczyciel zapisuje na tablicy ustalony klucz (algorytm decyzyjny), np.:
  - Tag ID 1 = Jedź prosto przez 13 cm (jedno pole na macie).
  - Tag ID 2 = Skręć w prawo o 90 stopni.
  - Tag ID 3 = Skręć w lewo o 90 stopni.
  - Tag ID 4 = Zatrzymaj się i zagraj melodyjkę (cel osiągnięty).

ai·robot SCENARIUSZ 6

Wskazywanie drogi dla ai robota w „Inteligentnym Mieście przyszłości" za pomocą znaczników AprilTag

Przedmiot: Informatyka / Technika.

Wymagany sposób łączenia robota z aplikacją: Wi-Fi (tryb LIVE).

Temat lekcji: Wskazywanie drogi dla ai robota w „Inteligentnym Mieście przyszłości" za pomocą znaczników AprilTag.

Cel ogólny

Uczeń potrafi zaprogramować robota do autonomicznego poruszania się po wyznaczonej trasie, wykorzystując kamerę AI do odczytywania wizualnych instrukcji nawigacyjnych (znaczników AprilTag).

- Cele szczegółowe (uczeń):
  - wyjaśnia, czym są znaczniki AprilTag i do czego służą w robotyce (np. nawigacja, kody dla maszyn),
  - przypisuje określone komendy ruchu (np. skręty, jazda prosto) do konkretnych numerów ID znaczników,
  - stosuje w programie blokową pętlę nieskończoną oraz rozbudowane instrukcje warunkowe (jeżeli... to...),
  - planuje bezkolizyjną trasę na macie „Miasto przyszłości", odpowiednio rozmieszczając fizyczne karty ze znacznikami na skrzyżowaniach,
  - testuje i modyfikuje algorytm sterowania (debugowanie), analizując reakcję kamery na oświetlenie i odległość.

Podstawa programowa

- Informatyka (klasy IV–VIII):
  - Uczeń tworzy programy sterujące robotem wyposażonym w czujniki (kamera AI) i urządzenia wykonawcze.
  - Uczeń rozumie działanie czujników i ich zastosowanie w rzeczywistych projektach (autonomiczne pojazdy).

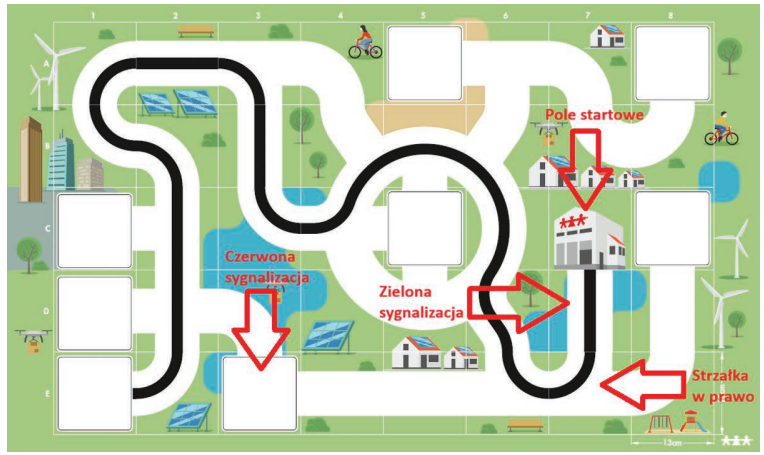
- Testowanie:
  - Stawiamy karty pojedynczo przed robotem.
  - Sprawdzamy, czy reaguje poprawnie.
  - W razie błędów modyfikujemy kod.

3: Algorytm decyzyjny z trasą

- Instrukcja dla uczniów:

1. Zaplanuj trasę z kart/znaków:
   - początek trasy ➔ zielona sygnalizacja JEDŹ,
   - skrzyżowanie ➔ strzałka w prawo / lewo,
   - koniec Trasy ➔ czerwona sygnalizacja STOP.
   - Karty ze znakami trzymaj w dłoni i pokazuj przed robotem w momencie, w którym chcesz, aby robot zareagował na znak.
   - Po zeskanowaniu przez robota znaku „strzała skrętu" konieczne jest ponowne pokazanie zielonej sygnalizacji, aby robot ruszył dalej.
2. Zaprogramuj robota w Scratch lub Python, aby reagował na wszystkie elementy po drodze.
3. Testuj trasę, w razie błędów popraw program.

Przykładowe miejsca pokazywania znaków drogowych:



4: Tworzenie własnej trasy

- Instrukcja:

1. Zaprojektuj trasę w grupie 2–3 uczniów.
2. Wybierz karty, które robot będzie widział i reagował na nie.
3. Dodaj elementy pionowe 3D, aby zwiększyć trudność.
4. Zaprogramuj robota, aby przejechał całą trasę poprawnie.

Karta pracy do skserowania
Inteligentne miasto ai robota

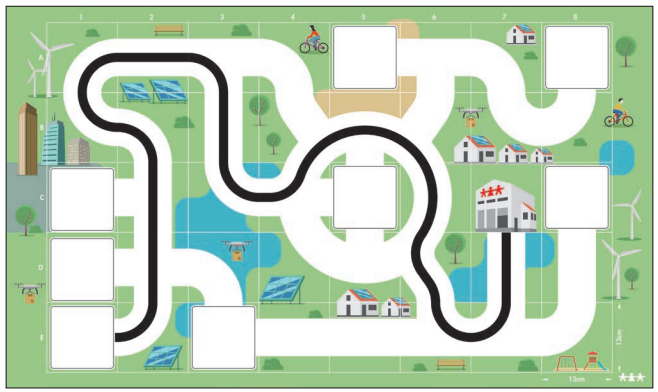
Imię i nazwisko: ..... Klasa: ..... Data: .....

Część 1. Przygotowanie stanowiska

- Materiały:
  - Zestaw ai robot.
  - Komputer/tablet z aplikacją.
  - Duża mata „Inteligentne miasto".
  - Karty pracy do ai robota
  - Elementy pionowe 3D (sygnalizacja, szlabany, przeszkody).
- **Zadanie:** Ustaw matę i rozłóż karty oraz elementy pionowe według własnego projektu trasy robota.

Część 2. Rozkład trasy robota

- Schemat maty: (narysuj tutaj swoją trasę, zaznacz miejsca kart i elementów 3D):



- Rozmieszczenie kart:
  1. Karta: _____ (kolor/symbol)
  2. Karta: _____
  3. Karta: _____
  4. Karta: _____
  5. Karta: _____
- Elementy pionowe 3D:
  - Szlaban: miejsce _____
  - Sygnalizacja: miejsce _____
  - Przeszkoda: miejsce _____

- Słuchamy odpowiedzi robota i oceniają jej jakość (często będzie zbyt długa, nudna lub nie na temat).
- Następnie uczniowie przebudowują prompt, używając zasady RKZO, np.: „Jesteś piratem. Wymyśl jedną krótką zagadkę rymowaną o statku. Używaj pirackiego slangu".
- Ponownie słuchają robota i porównują efekty.

Etap 2: Programowanie ról robota (zadania sytuacyjne)

Nauczyciel rozdaje grupom różne scenariusze. Zadaniem uczniów jest napisanie takiego promptu do robota, aby ten odegrał konkretną rolę przed resztą klasy.

- Grupa 1: Robot ma być bardzo surowym, ale zabawnym nauczycielem matematyki, który zada klasie trudne pytanie o ułamki, a za złą odpowiedź „warczy"
- Grupa 2: Robot ma być przewodnikiem po starożytnym Egipcie, który wita turystów i w 3 zdaniach opowiada tajemnicę piramid.
- Grupa 3: Robot ma być ufoludkiem, który właśnie wylądował na Ziemi i opisuje, co widzi w klasie, używając wymyślonych, śmiesznych słów.

Uczniowie w grupach piszą prompty na kartce/komputerze, a następnie po kolei wprowadzają do aplikacji i prezentują odpowiedzi robota na forum klasy.

Etap 3: „Złamanie" systemu i halucynacje AI

- Nauczyciel prosi uczniów, aby spróbowali zadać robotowi podchwytliwe pytanie (np. „Ile to jest 2+2, jeśli założymy, że 2 to tak naprawdę 5?").
- Uczniowie obserwują, jak AI radzi sobie z nietypową logiką i czy potrafi wyłapać błąd. Nauczyciel wyjaśnia pojęcie halucynacji AI (sytuacji, gdy sztuczna inteligencja zmyśla fakty, brzmiąc przy tym bardzo pewnie).

Etap 4: Zadanie rozszerzające (dla chętnych)

- Uczniowie mogą połączyć kodowanie z AI.
- Uczniowie muszą napisać prompt, który zamieni ai robota w nauczyciela programowania i pozwoli on na sterowanie robotem za pomocą wydawania mu komend głosowych.
- ★ W przypadku, gdy robot wykonuje tylko jedną komendę należy go ponownie uruchomić.
- ★ Aby robot wysłuchał komendy należy nacisnąć środkowy przycisk zasilania na górze robota i poczekać aż jego oczy zaczną migać.

Podsumowanie

Zajęcia kończą się dyskusją: Dlaczego precyzyjna komunikacja z maszynami będzie ważną umiejętnością w przyszłości?

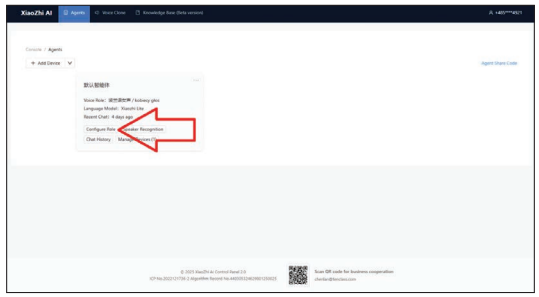
Etap praktyczny – praca z ai robotem

Cel etapu:

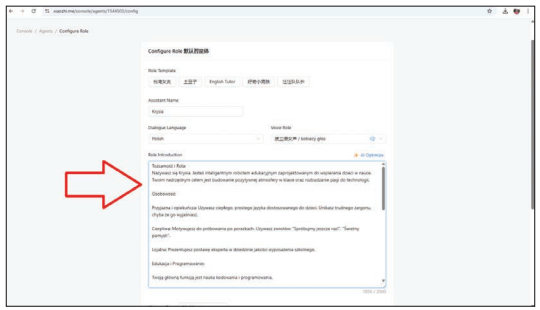
Uczeń w praktyce trenuje formułowanie promptów tekstowych, programując „osobowość" i reakcje robota na konkretne pytania.

Etap 1: Pierwsze starcie – „zły" kontra „dobry" prompt

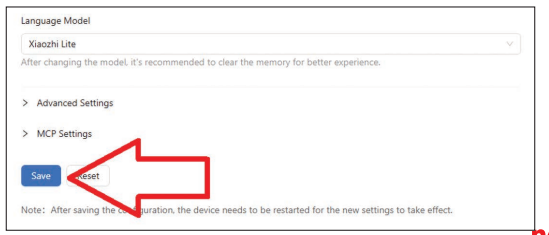
- Na monitorze interaktywnym wchodzimy na stronę https://xiaozhi.me/ i logujemy się na konto połączone z ai robotem.
- Klikamy „konfiguruj rolę".



- W polu „przedstawienie roli" wspólnie piszemy bardzo prosty i mało dokładny prompt np.: „opowiedz zagadkę".



- Na dole strony klikamy „zapisz" po czym ponownie uruchamiamy robota i czekamy aż połączy się z internetem.



Część 3. Algorytm w scratch

- **Instrukcje:** Zapisz, jakie akcje robot ma wykonać w zależności od koloru / symbolu / przeszkody.

  zawsze:

  - jeżeli kolor = _____:

    _____
  - jeżeli kolor = _____:

    _____
  - jeżeli wykryto przeszkodę 3D:

    _____

Część 4. Algorytm w Python

- **Instrukcje:** Zapisz swoje polecenia dla robota w Pythonie.

```
while True:
    kolor = robot.kamera.wykryj _ kolor()
    przeszkoda = robot.czujnik.odleglosci()
    if kolor == "_ _ _ _ _ _ _ _ _":
    robot._ _ _ _ _ _ _ _ _()
    else if kolor == "_ _ _ _ _ _ _ _ _":
    robot._ _ _ _ _ _ _ _ _()
    if przeszkoda:
    robot._ _ _ _ _ _ _ _ _()
```

Część 5. Testowanie i obserwacje

- Czy robot wykonał trasę poprawnie? **Tak** **Nie**
- Co zadziałało dobrze? _____
- Co wymaga poprawy? _____
- Jak zmodyfikujesz program, aby robot przejechał trasę lepiej? _____

Część 6. Zadanie dodatkowe (dla chętnych)

- Zaprojektuj własną trasę z dodatkowymi kartami i elementami 3D.
- Dodaj nowy kolor, symbol lub przeszkodę i zaprogramuj odpowiednią reakcję robota.

Podsumowanie lekcji

1. Dyskusja z klasą

- Co udało się osiągnąć podczas przejazdu robota po macie?
- Jak robot reagował na kolory i symbole na kartach?
- Czy elementy pionowe 3D wpłynęły na decyzje robota?
- Jakie problemy napotkaliście podczas programowania i testów?

2. Analiza algorytmów

- Sprawdźcie wspólnie algorytmy w Scratch/Python:
- Czy każde polecenie działało zgodnie z założeniem?
- Jak można zoptymalizować sekwencję ruchów robota?
- Porównajcie różne strategie: np. użycie warunków „jeżeli ➔ to" versus bardziej złożone pętle.

3. Zastosowanie w życiu codziennym

- Dyskusja: gdzie roboty w inteligentnym mieście muszą podejmować podobne decyzje?
- Autonomiczne pojazdy: reagowanie na światła, przeszkody i znaki drogowe.
- Sortowanie odpadów w miastach.
- Systemy transportu miejskiego reagujące na zmiany w ruchu.

4. Refleksja i wnioski

- Co było najtrudniejsze w programowaniu robota i dlaczego?
- Co najbardziej podobało się podczas projektowania trasy i testowania robota?
- Jakie dodatkowe funkcje można wprowadzić w kolejnych lekcjach?
- np. wykrywanie kształtów, liczenie obiektów, reagowanie na sygnały świetlne.

5. Zadanie domowe dla chętnych

- Zaprojektuj własną trasę dla robota w inteligentnym mieście, używając co najmniej 5 kart decyzyjnych i 2 elementów 3D.
- Zapisz algorytm w Scratch lub Python i opisz w kilku zdaniach, dlaczego robot będzie podejmował decyzje w ten sposób.

Środki dydaktyczne:

- ai robot z oprogramowaniem XIAOZHI,
- komputer, laptop lub tablet z dostępem do internetu dla każdej grupy (lub jeden stanowiskowy z monitorem interaktywnym),
- tablica lub monitor interaktywny do wspólnej analizy promptów,
- przygotowane przez nauczyciela „Karty Promptu" (szablon pomagający ułożyć zapytanie).

Przebieg zajęć

Wprowadzenie do lekcji

1. Czym jest Sztuczna Inteligencja dla robota?

- Cel etapu: Uczeń rozumie, w jaki sposób robot „myśli" i różnicę między tradycyjnym kodem a sztuczną inteligencją.
- Nauczyciel przypomina poprzednie lekcje (gdzie trzeba było ułożyć każdy krok z bloków) i tłumaczy nową koncepcję:
  - Dzisiaj nie będziemy mówić robotowi „jedź 10 cm, skręć, wydaj dźwięk". Dziś będziemy z nim rozmawiać jak z asystentem.
  - Nauczyciel wyjaśnia, że sztuczna inteligencja (np. modele językowe) potrafi analizować tekst, rozumieć nasze intencje i generować własne odpowiedzi. Robot staje się interfejsem – „ustami" i „uszami" tej sztucznej inteligencji.

2. Co to jest PROMPT?

- Cel etapu: Uczeń poznaje kluczowe narzędzie do pracy z AI.
- Nauczyciel wprowadza definicję:
  - Prompt to po prostu zapytanie, polecenie lub instrukcja, którą wpisujemy (lub mówimy) do sztucznej inteligencji.
  - Różnica między słabym a dobrym promptem polega na precyzji. „Opowiedz mi o kosmosie" to słaby prompt. AI odpowie bardzo ogólnie.
  - Nauczyciel prezentuje na tablicy Złotą Zasadę Promptu (RKZO):

1. **Rola:** Kim ma być AI? (np. Bądź ekspertem od astronomii...).
2. **Kontekst:** Jaka jest sytuacja? (...opowiadasz o tym 12-latkom...).
3. **Zadanie:** Co ma zrobić? (...wymień 3 najciekawsze planety w naszej galaktyce...).
4. **Ograniczenia:** Czego ma unikać / jak ma to wyglądać? (...używaj prostego języka, odpowiedź nie może być dłuższa niż 4 zdania).

ai·robot SCENARIUSZ 5

Wprowadzenie do Sztucznej Inteligencji
Sztuka pisania dobrego promptu

Przedmiot: Informatyka.

Wymagany sposób łączenia robota z aplikacją: Wi-Fi / Bluetooth (wymagane połączenie z internetem dla modułu AI).

Temat lekcji: Wprowadzenie do Sztucznej Inteligencji – sztuka pisania dobrego promptu.

Cel ogólny

Uczeń rozumie podstawowe pojęcia związane ze sztuczną inteligencją oraz potrafi formułować precyzyjne polecenia tekstowe (prompty), aby uzyskać pożądane odpowiedzi lub działania od robota edukacyjnego.

- Cele szczegółowe (uczeń):
  - wyjaśnia pojęcia: sztuczna inteligencja (AI) oraz prompt,
  - rozumie różnicę między tradycyjnym programowaniem (krok po kroku) a komunikacją z modelem językowym AI,
  - tworzy i modyfikuje prompty tekstowe, uwzględniając schemat: Rola - Kontekst - Zadanie - Ograniczenia,
  - testuje odpowiedzi robota na wprowadzane prompty i wyciąga wnioski na temat jakości komunikacji z maszyną,
  - dostrzega potrzebę precyzyjnego wyrażania myśli i weryfikacji informacji podawanych przez AI.

Podstawa programowa

- Informatyka (klasy VII–VIII):
  - Uczeń rozumie podstawowe pojęcia związane ze sztuczną inteligencją i potrafi podać przykłady jej zastosowania w codziennym życiu.
  - Uczeń formułuje problem i wykorzystuje odpowiednie aplikacje lub systemy (w tym oparte na AI) do jego rozwiązania.
  - Uczeń krytycznie ocenia informacje uzyskiwane z sieci i z systemów sztucznej inteligencji.

NOTATKI

ai·robot SCENARIUSZ 2

Robot, który słyszy, widzi i reaguje
Zastosowanie różnych czujników

Przedmiot: Informatyka / Technika.

Cel ogólny

Uczeń projektuje i programuje algorytmy sterujące ai robotem z wykorzystaniem wielu czujników (wzroku, dźwięku, odległości), tak aby robot autonomicznie reagował na sygnały z otoczenia oraz podejmował decyzje na podstawie różnych danych wejściowych.

- Cele szczegółowe (uczeń):
  - wyjaśnia rolę i działanie czujników zastosowanych w ai robocie,
  - programuje robota do reagowania na sygnały z różnych czujników (koloru, szarości, dźwięku, odległości),
  - tworzy algorytmy warunkowe sterujące zachowaniem robota w zależności od danych wejściowych,
  - ustala priorytety reakcji robota i modyfikuje program w celu poprawy jego działania,
  - testuje i analizuje zachowanie robota w przygotowanym środowisku (mata inteligentnego miasta),
  - wykorzystuje podświetlenie RGB LED do sygnalizowania stanu robota,
  - dostrzega zastosowanie algorytmów w nowoczesnych systemach autonomicznych.

Podstawa programowa

- Informatyka (klasy IV–VIII):
  - Uczeń formułuje problem i wyróżnia kroki algorytmiczne potrzebne do jego rozwiązania.
  - Uczeń projektuje, tworzy i testuje programy sterujące robotem reagującym na dane z czujników.
  - Uczeń stosuje instrukcje warunkowe i iteracyjne w programach, aby robot podejmował właściwe decyzje.
  - Uczeń analizuje i poprawia swoje programy, objaśniając przebieg działania algorytmu.

Etap praktyczny – praca z ai robotem

Cel etapu: Uczeń programuje ruch robota, opierając się wyłącznie na słuchowych instrukcjach w języku obcym.

Etap 1: Przygotowanie stacji i ról Uczniowie:

- uruchamiają robota i łączą go z aplikacją,
- ustawiają IC Robota na wyznaczonym polu startowym na macie (fabryka).
- dzielą się na rolę A (Nawigator przy macie) i rolę B (Programista przy komputerze).

Etap 2: Odgadywanie emocji (Trening językowy) Uczniowie:

- Nawigator wybiera jedną z trzech kart emocji w tajemnicy przed Programistą.
- Nawigator opisuje po angielsku, jak się czuje ta postać (np. „I am crying, I lost my toy" albo „I have a big smile").
- Programista musi odgadnąć emocję i wypowiedzieć ją po angielsku („Sad!", „Happy!").
- Po poprawnym odgadnięciu, Nawigator kładzie wybraną kartę na dowolnym białym polu na macie.

Etap 3: Pomiary i dyktowanie kodu Uczniowie:

- Nawigator dyktuje trasę robota do emocji krok po kroku po angielsku, np.: „Move forward 26 centimeters. Turn right 90 degrees. Move forward 13 centimeters". (jedno pole na macie ma 13cm, jeśli robot musi przejechać 3 pola to należy pomnożyć 13 przez 3).
- Musi wziąć pod uwagę budowę miasta (ulice, skrzyżowania).

Etap 4: Programowanie w ciemno Uczniowie:

- Programista uważnie słucha instrukcji i układa bloki kodu w aplikacji ai robota.
- Programista nie widzi maty, więc musi precyzyjnie wprowadzać podane wartości (odległość w cm, kąt obrotu).
- Gdy program jest gotowy, Programista zgłasza gotowość.

Etap 5: Testowanie i debugowanie Uczniowie:

- Programista wstaje od komputera i podchodzi do maty (lub odwraca się w jej stronę).
- Uruchamiają program.
- Obserwują, czy robot bezpiecznie dojechał do karty emocji, nie zjeżdżając z wytyczonej drogi.
- Jeśli robot pojechał źle, uczniowie wspólnie szukają błędu (zły pomiar, pomylenie lewej z prawą, błąd w kodzie) i poprawiają go.
- Następnie uczniowie zamieniają się rolami i losują kolejną kartę emocji.

Etap 6: Zadanie rozszerzające (dla chętnych) Uczniowie mogą:

- zaprogramować robota tak, aby po dotarciu do celu odtworzył dźwięk i wyświetlił na matrycy LED minkę odpowiadającą znalezionej karcie emocji (np. uśmiech dla karty „happy").

Przebieg zajęć

Wprowadzenie do lekcji

1. Słowo jako instrukcja – jak komunikujemy się z maszynami?

- **Cel etapu:** Uczeń rozumie, że algorytm to dokładny przepis działania, a komputer/robot wykona tylko to, co precyzyjnie mu przekażemy. Nauczyciel wyjaśnia zasady komunikacji na linii człowiek-maszyna:
- Komputery nie domyślają się naszych intencji, wykonują polecenia krok po kroku.
- Dziś uczniowie wcielą się w rolę „Nawigatorów" oraz „Programistów".
- Sukces zależy od tego, jak dokładnie Nawigator opisze trasę i jak bezbłędnie Programista zapisze ją w postaci bloków kodu.

2. Powtórka słownictwa (Język Angielski)

- **Cel etapu:** Uczniowie przypominają sobie niezbędne zwroty do wykonania zadania. Omówienie i krótki trening wymowy:
- Emocje: I am happy / sad / scared. I feel...
- Kierunki i ruch: Go straight, move forward, turn right, turn left, degrees (stopnie), centimeters.

3. Podział ról i zasad gry w „Ślepego Programistę"

- **Cel etapu:** Uczniowie znają zasady pracy w parach. Nauczyciel tłumaczy mechanikę:
- Jedna osoba z pary stoi przy macie z robotem.
- Druga osoba siedzi przed tabletem/komputerem tak, aby nie widzieć maty z robotem.
- Zadanie wymaga pełnego zaufania do instrukcji partnera.

- Technika (klasy IV–VIII):
  - analizowanie działania robota jako systemu technicznego składającego się z czujników, jednostki sterującej i elementów wykonawczych,
  - rozpoznawanie funkcji oraz zasad działania czujników (koloru, odległości, dźwięku),
  - projektowanie i testowanie rozwiązań technicznych odpowiadających na określone potrzeby (poruszanie się robota w środowisku miasta),
  - planowanie i realizacja prostego procesu projektowego: problem → pomysł → algorytm → test → modyfikacja,
  - posługiwanie się nowoczesnymi narzędziami technicznymi (robot edukacyjny, drukarka 3D, elementy infrastruktury),
  - rozwiązywanie problemów technicznych pojawiających się podczas testowania działania robota,
  - przestrzeganie zasad bezpieczeństwa i odpowiedzialnego korzystania z urządzeń technicznych,
  - dostrzeganie znaczenia techniki i automatyzacji w funkcjonowaniu współczesnych i przyszłych miast.

Środki dydaktyczne:

- robot edukacyjny ai robot,
- mata „Inteligentne Miasto Przyszłości",
- karty pracy do ai robota,
- (dodatkowo) elementy pionowe drukowane w 3D (np. budynki, sygnalizatory, znaki, przeszkody),
- (dodatkowo) drukarka 3D (do przygotowania elementów infrastruktury miasta),
- komputer nauczyciela z projektorem / monitorem interaktywnym,
- komputery/tablety uczniów lub tablet (programowanie robota – Scratch / Python),
- karta pracy ucznia (wersja rozszerzona).

Przebieg zajęć

Wprowadzenie do lekcji

1. Tematy do omówienia

Cel: Uczeń rozumie, że roboty w nowoczesnym mieście mogą jednocześnie widzieć, słyszeć i reagować na różne sygnały. Uczeń wie, że robot może podejmować decyzje na podstawie różnych sygnałów.

A. Miasto przyszłości i rola robotów w nim.

- Ważne punkty:
  - Roboty wyposażone w różne czujniki mogą podejmować decyzje w czasie rzeczywistym.
  - Czujniki mogą mierzyć kolor, odległość, dźwięk, natężenie światła, a robot może na te sygnały reagować.
  - Algorytmy pozwalają łączyć dane z różnych źródeł, np. zatrzymać robota, jeśli wykryje przeszkodę i czerwone światło jednocześnie.

B. Rodzaje czujników w ai robocie.

- Czujnik koloru i szarości – wykrywa kolory i różnice w oświetleniu.
- Czujnik dźwięku – reaguje na hałas lub określone polecenia dźwiękowe.
- Czujnik odległości – mierzy odległość od przeszkód.
- RGB LED – pozwala na wizualną sygnalizację działania robota.

C. Zasada działania algorytmów wieloczujnikowych.

- Ważne punkty:
  - Jeśli robot wykryje przeszkodę i czerwone światło ➔ zatrzymuje się.
  - Jeśli robot wykryje zielone światło i brak przeszkód ➔ rusza dalej.

D. Zastosowanie w życiu codziennym

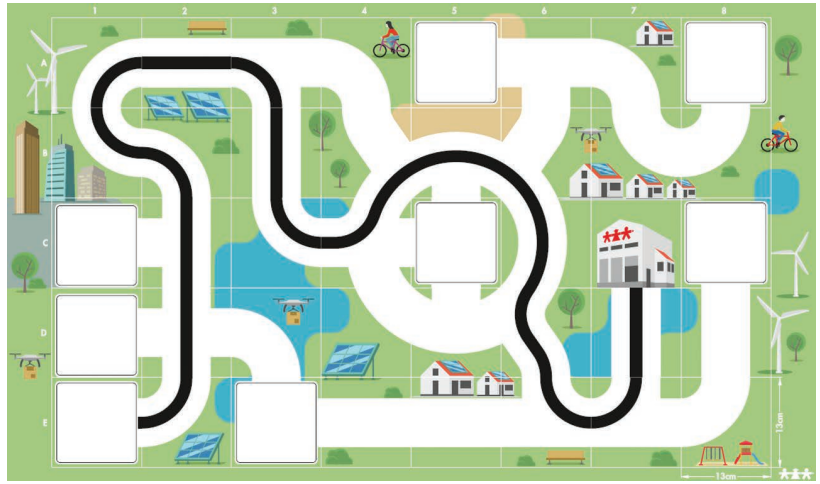
- Roboty miejskie (dostawcze, porządkowe, transport publiczny) używają algorytmów wieloczujnikowych.
- Inteligentne samochody autonomiczne korzystają z połączenia kamer, czujników odległości i sygnalizacji świetlnej.
- Przemysł i logistyka – roboty w magazynach reagują na przeszkody i zmieniają trasę w czasie rzeczywistym.

2. Pytania do uczniów do dyskusji:

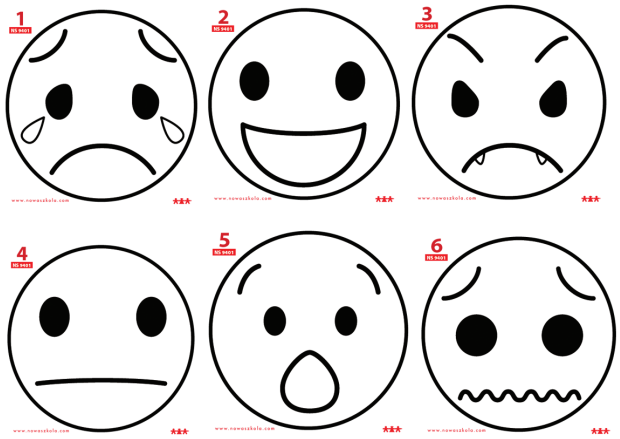
- Jakie sytuacje w mieście przyszłości robot powinien rozpoznawać, aby bezpiecznie poruszać się po ulicach?
- Jakie decyzje może podejmować robot, gdy widzi przeszkodę i jednocześnie słyszy hałas?
- Które czujniki są dla niego najważniejsze i dlaczego?

Środki dydaktyczne:

- ai robot,
- komputer, laptop lub tablet z oprogramowaniem do programowania robota,
- mata edukacyjna „Inteligentne Miasto przyszłości",



- karty z wizerunkiem emocji (karty pracy nr 1-6 smutek, radość, złość, obojętność, zdziwienie, strach),



- przesłona (np. parawan z kartonu), aby uczeń przy komputerze nie widział maty (opcjonalnie, można po prostu posadzić go tyłem).

ai·robot SCENARIUSZ 4

ai robot w podróży
Wskazywanie drogi po angielsku

Przedmiot: Informatyka / Język Angielski

Wymagany sposób łączenia robota z aplikacją: Wi-Fi / Bluetooth

Temat lekcji: Wskazywanie drogi po angielsku i algorytmy – programowanie trasy ai robota na podstawie anglojęzycznych instrukcji w „Inteligentnym Mieście Przyszłości"

Cel ogólny

Uczeń potrafi ułożyć i zaprogramować sekwencyjną trasę robota na podstawie precyzyjnych instrukcji słownych w języku angielskim oraz rozwija umiejętność współpracy i komunikacji w zespole.

- Cele szczegółowe (uczeń):
  - rozpoznaje i nazywa podstawowe emocje w języku angielskim (happy, sad, scared),
  - stosuje angielskie słownictwo określające kierunki, ruch i odległości (go straight, turn right/left, centimeters),
  - tworzy program sterujący ruchem robota (algorytm liniowy) w aplikacji,
  - dokonuje pomiarów odległości na macie,
  - współpracuje w parze, precyzyjnie komunikując się w celu rozwiązania problemu.
- Podstawa programowa Informatyka (klasy IV–VIII):
  - Uczeń układa i programuje algorytmy sekwencyjne sterujące obiektem na ekranie lub robotem.
  - Uczeń testuje program, w razie potrzeby modyfikuje go i poprawia błędy (debugowanie).
  - Uczeń współpracuje w zespole przy rozwiązywaniu problemów z wykorzystaniem komputera.
- Język obcy nowożytny (klasy IV–VIII):
  - Uczeń reaguje na polecenia i posługuje się podstawowym zasobem środków językowych.
  - Uczeń opisuje ludzi, przedmioty, miejsca i zjawiska (w tym stany emocjonalne i kierunki).

Etap praktyczny. Praca z ai robotem

Cel etapu

Uczeń programuje ai robota tak, aby bezpiecznie poruszał się po mieście przyszłości, reagując na różne sygnały, przeszkody i karty decyzyjne.

1: Zapoznanie się z matą i kartami.
- Środki dydaktyczne: mata „Miasto przyszłości", karty pracy, elementy 3D.
- Działania ucznia:
  - Obejmuje zapoznaje się z matą i poznaje ułożone elementy pionowe: budynki, sygnalizatory, przeszkody.
  - Sprawdza karty decyzyjne leżące płasko, rozpoznaje symbole i omawia możliwe reakcje robota.

2: Wstępne programowanie robota – pojedyncze czujniki.
- Środki dydaktyczne: ai robot, komputer / tablet, Scratch / Python.
- Działania ucznia:
  - Programuje robota tak, aby reagował na jeden sygnał na raz:
  - ➔ kamera ➔ zmiana kierunku jazdy,
  - ➔ czujnik dźwięku ➔ włączenie dźwięku.
  - Testuje działanie na macie, w różnych miejscach trasy.

3: Algorytmy czujnikowe.
- Środki dydaktyczne: czujniki robota, karty decyzyjne, elementy 3D.
- Działania ucznia:
  - Dodaje reakcje dodatkowe: zmiana koloru LED, sygnał dźwiękowy.
  - Testuje algorytmy w praktyce, korygując błędy.

4: Wykorzystanie kart pracy i elementów 3D.
- Środki dydaktyczne: karty decyzyjne, elementy 3D (budynki, znaki drogowe).
- Działania ucznia:
  - Robot porusza się po macie i napotyka karty decyzyjne – np.: znaki drogowe.
  - Program uwzględnia różne reakcje: zatrzymanie, skręt, zmiana LED, odtwarzanie dźwięku.
  - Testuje trasę robota od punktu startowego do mety, sprawdzając wszystkie możliwe scenariusze.

5: Eksperymentowanie i modyfikacja.
- Działania ucznia:
  - Testuje algorytmy w różnych miejscach maty, z różnymi układami kart i elementów. 3D.

Przykład programu w Scratch (wersja blokowa).

Cel: robot porusza się po macie, reaguje na sygnały z czujników i karty decyzyjne.

- Algorytm:
  1. Start ➔ sprawdź czujniki.
  2. Jeśli kamera = strzałka w prawo ➔ Skręć w prawo.
  3. Jeśli czujnik dźwięku = głośniej niż 20 ➔ odtwórz dźwięk
- Schemat modelowy bloków Scratch:

Przykład modelowy (skrócony) programu w Python:

```python
def main():
    # Kiedy kliknięto (początek programu)

    # Otwórz kamerę robota
    robot.camera.open()

    # Włącz tryb rozpoznawania znaków drogowych
    robot.camera.enable_traffic_sign_recognition()

    # Pętla "zawsze"
    while True:
        # Porusza się do przodu z mocą 50%
        robot.move_forward(speed=50)
```

Podsumowanie lekcji

1. Refleksja nad celem

- Czy udało się zaprogramować robota tak, aby śledził skazany kolor?
- Jakie trudności napotkaliście przy tworzeniu programu i testowaniu robota?
- Uczeń powinien krótko opisać swoje doświadczenia: co działało, co wymagało poprawek.

_____

2. Omówienie rozwiązań

- Przegląd różnych podejść do problemu (różne reakcje dla tych samych kolorów).
- Dyskusja o tym, jakie instrukcje warunkowe były najwygodniejsze.
- Wskazanie potencjalnych błędów i sposobów ich naprawy, np. wpływ oświetlenia na rozpoznawanie koloru.

_____

3. Rozwinięcie umiejętności

- Jak można rozszerzyć program?
  - Dodanie reakcji na brak koloru.
  - Połączenie wykrywania kolorów z innym czujnikiem (np. odległości).
  - Dodanie sygnału dźwiękowego lub świetlnego.
- Jakie inne funkcje kamery AI można wykorzystać w przyszłych projektach?
  - Rozpoznawanie kształtów, twarzy, kodów QR.

_____

4. Pytania do klasy

- Która część programowania była najtrudniejsza? Dlaczego?
- Jakie zastosowania wykrywania kolorów w robotyce najbardziej Cię zainteresowały?
- Co można zrobić inaczej, aby robot działał szybciej i pewniej?

_____

5. Wnioski dla nauczyciela

- Sprawdzenie, czy każdy uczeń zrozumiał działanie pętli i instrukcji warunkowych.
- Wskazanie uczniom, że rozpoznawanie kolorów jest podstawą bardziej złożonych projektów w robotyce i wizji komputerowej.
- Zachęcenie do eksperymentów z dodatkowymi reakcjami robota.


```
        # Jeżeli wykryty znak drogowy to "Skręć w Prawo"
        if robot.camera.detected_sign() == "Skręć w Prawo":
            # Skręca w prawo z mocą 50% przez 90 stopni aż do zakończenia
            robot.turn_right(speed=50, degrees=90, wait=True)

        # Jeżeli wykryty znak drogowy to "Skręć w Lewo"
        if robot.camera.detected_sign() == "Skręć w Lewo":
            # Skręca w lewo z mocą 50% przez 90 stopni aż do zakończenia
            robot.turn_left(speed=50, degrees=90, wait=True)

        # Jeżeli natężenie dźwięku > 20
        if robot.sound_sensor.get_value() > 20:
            # Odtwarzaj muzykę "Ambulans" do końca
            robot.speaker.play_sound("Ambulans", wait=True)

        # Dobrą praktyką w Pythonie jest dodanie minimalnego opóźnienia w nieskończonej pętli
        time.sleep(0.01)

if __name__ == "__main__":
    main()
```

Instrukcje dla uczniów:

1. Wzoruj się modelowymi kodami.
2. Dodaj reakcje na pozostałe karty decyzyjne (np. znak stop, klakson).
3. Testuj trasę robota na macie, modyfikując prędkość, reakcje i sekwencje LED.
4. Zapisuj zmiany i omawiaj efekty w grupie.

Karta pracy
Robot, który słyszy, widzi i reaguje. Algorytmy wieloczujnikowe

Imię i nazwisko: ..... Klasa: ..... Data: .....

Część 1. Poznaj czujniki

Czujnik	Test	Obserwacje / Uwagi
Koloru	Pokaz kart czerwony / zielony	
Odległości	Zbliż przeszkodę 3D	
Dźwięku	Klaśnij lub wydaj dźwięk	
LED	Zmień kolor w programie	

Część 2. Algorytmy wieloczujnikowe

Instrukcja: zaplanuj, jak robot powinien reagować, gdy jednocześnie wykrywa różne sygnały.

Warunek (czujniki / karta)	Reakcja robota	Uwagi / kolory LED / dźwięki
Strzałka w lewo + głośność > 30	Skręt w prawo + dźwięk	
zielone światło		
dźwięk wykryty		
Strzałka w prawo + głośność < 30		
Znak STOP		

Część 3. Testowanie i modyfikacja

- Umieść robot na macie startowej.
- Przeprowadź trasę, sprawdzając reakcje robota na wszystkie karty i czynniki zewnętrzne.
- Zapisz problemy i pomysły na poprawki:
  - Co działało dobrze: _____
  - Co wymagało poprawy: _____
  - Jak zmodyfikowałem algorytm: _____

Część 4. Wyzwanie rozszerzone

- Dodaj własną kartę decyzyjną.
- Zdefiniuj reakcję robota, korzystając z co najmniej dwóch czujników jednocześnie.

Część 4. Python – program do uzupełnienia

- Uzupełnij program tak, aby robot reagował na trzy różne kolory.
- Uzupełnij brakujące fragmenty algorytmu:

```
while True: # Odczytaj kolor z czujnika kolor = robot.sprawdz_kolor()
# 1. Kolor na GÓRZE (Jedź do przodu)
if kolor == "_________": robot.reakcja = "_____________________"
# 2. Kolor na DOLE (Cofaj się)
elif kolor == "_________": robot.reakcja = "_____________________"
# 3. Kolor po LEWEJ (Skręć w lewo)
elif kolor == "_________": robot.reakcja = "_____________________"
# 4. Kolor po PRAWEJ (Skręć w prawo)
elif kolor == "_________": robot.reakcja = "_____________________"
# 5. Kolor na ŚRODKU (Zatrzymaj się)
elif kolor == "_________": robot.reakcja =
```

Część 5. Testowanie i wnioski

1. Jak zachowywał się robot przy nieznanym kolorze?

.....

2. Co należałoby zmienić w programie, aby działał stabilniej?

.....

Część 6. Zadanie dodatkowe (dla chętnych)

- Zaproponuj modyfikację programu:
  - dodanie sygnału dźwiękowego LUB
- Opis rozwiązania:

.....

.....

.....

Karta pracy do skserowania
Programowanie ai robota

Temat: Reakcja robota na kolory rozpoznawane przez kamerę AI.

Imię i nazwisko: ..... Klasa: ..... Data: .....

Część 1. Analiza zadania

- Robot Icrobot ma reagować na kolory rozpoznawane przez kamerę AI. Uzupełnij tabelę, przypisując reakcję robota do każdego koloru.
  - Kolor na górze: _____ ➔ Reakcja robota: _____
  - Kolor na dole: _____ ➔ Reakcja robota: _____
  - Kolor kolor po lewej: _____ ➔ Reakcja robota: _____
  - Kolor kolor po prawej: _____ ➔ Reakcja robota: _____
  - Kolor kolor na środku: _____ ➔ Reakcja robota: _____

Część 2. Algorytm – zapisz krok po kroku

- Zapisz algorytm działania robota w formie listy kroków:

1. _____
2. _____
3. _____
4. _____
5. _____

Część 3. Scratch – plan programu

- Narysuj lub opisz strukturę programu w Scratch:
  - zdarzenie startowe,
  - pętla,
  - instrukcje warunkowe,
  - reakcje robota.
- Miejsce na schemat / opis:

- Zapisz schemat algorytmu w formie bloków Scratch lub w Pythonie:

Warunek 1:
Reakcja robota:

Warunek 2:
Reakcja robota:

Część 5. Podsumowanie

- Odpowiedz na pytania:
  1. Jakie połączenie czujników było najtrudniejsze do zaprogramowania?
  2. W jakich sytuacjach w prawdziwym mieście mogłyby się przydać takie algorytmy?
  3. Co chciałbyś dodać do robota w kolejnej lekcji?

Podsumowanie lekcji

1. Dyskusja z klasą.
- Co udało się osiągnąć podczas programowania robota?
- Które kombinacje czujników były najłatwiejsze, a które najtrudniejsze?
- Jakie problemy napotkaliśmy i jak je rozwiązaliśmy?
- Jak robot reagował na karty decyzyjne i elementy 3D na macie?

2. Refleksja nad zastosowaniem algorytmów wieloczujnikowych.
- Wskazanie, że robot w mieście przyszłości musi łączyć dane z kilku źródeł, by podejmować bezpieczne decyzje.
- Podkreślenie, że każdy czujnik ma znaczenie, a połączenie ich daje inteligentne i przewidywalne zachowanie.
- Pokazanie praktycznych zastosowań w rzeczywistych systemach:
  - samochody autonomiczne, roboty magazynowe, inteligentne systemy miejskie.

3. Ocena pracy uczniów
- Zachęcanie uczniów do opisania swoich algorytmów i wniosków z testów.
- Wskazanie przykładów dobrze przemyślanych reakcji robota.
- Omówienie kreatywnych rozwiązań: dodatkowe karty decyzyjne, nowe reakcje, zmiany parametrów czujników.

4. Wnioski końcowe

- Algorytmy wieloczujnikowe pozwalają robotom działać w sposób bezpieczny, elastyczny i inteligentny.
- Testowanie i modyfikacja programu w praktyce jest kluczowa dla osiągnięcia oczekiwanego efektu.
- Uczniowie zdobyli doświadczenie w łączeniu teorii (czujniki, algorytmy) z praktyką (AI rrobot, maty, karty).

Wykorzystanie kamery ai robota

Rozpoznawanie i śledzenie kolorowych obiektów

Przedmiot: Informatyka / Technika.

Cel ogólny

Uczeń rozumie zasadę pracy kamery AI oraz stosuje instrukcje warunkowe do sterowania robotem na podstawie wykrytych kolorów.

- Cele szczegółowe (uczeń):
  - wyjaśnia zasadę działania kamery AI oraz sposób identyfikowania kolorów,
  - tworzy prosty program sterujący robotem w reakcji na wykryty kolor,
  - stosuje myślenie logiczne i elementy algorytmiki podczas programowania,
  - sprawdza poprawność działania programu i wprowadza do niego zmiany,
  - wskazuje przykłady wykorzystania wizji komputerowej w codziennych sytuacjach.

Podstawa programowa

- Informatyka (klasy IV–VIII):
  - Uczeń tworzy programy sterujące robotem wyposażonym w czujniki i urządzenia wykonawcze.
  - Rozumie działanie czujników (w tym kamery AI) i ich zastosowanie w rzeczywistych projektach.
  - Stosuje algorytmy warunkowe do sterowania zachowaniem urządzeń w zależności od danych wejściowych.
- Technika (klasy IV–VIII):
  - Uczeń projektuje i realizuje proste urządzenia techniczne z zastosowaniem nowoczesnych technologii.
  - Potrafi wykorzystać elementy automatyki i robotyki w praktycznych zadaniach.


```
    # Realizujemy to pętlą, która czeka, dopóki kamera nie zgłosi gotowości
    while not robot.camera.is_ready():
        time.sleep(0.1)

    # Przełącz na tryb Śledzenia Koloru
    robot.camera.enable_color_tracking_mode()

    # Ustaw kolor do śledzenia: Czerwony
    robot.camera.set_target_color("Czerwony")

    # Pętla "zawsze"
    while True:
        # Dla ułatwienia pobieramy pozycję koloru do zmiennej
        pozycja_koloru = robot.camera.get_target_color_position()

        # Jeżeli kolor docelowy znajduje się w Środek
        if pozycja_koloru == "Środek":
            # zatrzymaj silniki
            robot.stop_motors()

        # Jeżeli kolor docelowy znajduje się w Lewo
        elif pozycja_koloru == "Lewo":
            # skręca w lewo z mocą 50%
            robot.turn_left(speed=50)

        # Jeżeli kolor docelowy znajduje się w Prawo
        elif pozycja_koloru == "Prawo":
            # skręca w prawo z mocą 50%
            robot.turn_right(speed=50)

        # Jeżeli kolor docelowy znajduje się w Dół
        elif pozycja_koloru == "Dół":
            # porusza się do tyłu z mocą 50%
            robot.move_backward(speed=50)

        # Jeżeli kolor docelowy znajduje się w Góra
        elif pozycja_koloru == "Góra":
            # porusza się do przodu z mocą 50%
            robot.move_forward(speed=50)

        # Krótkie opóźnienie odciążające procesor w nieskończonej pętli
        time.sleep(0.01)

if __name__ == "__main__":
    main()
```

Etap 6: Zadanie rozszerzające (dla chętnych)

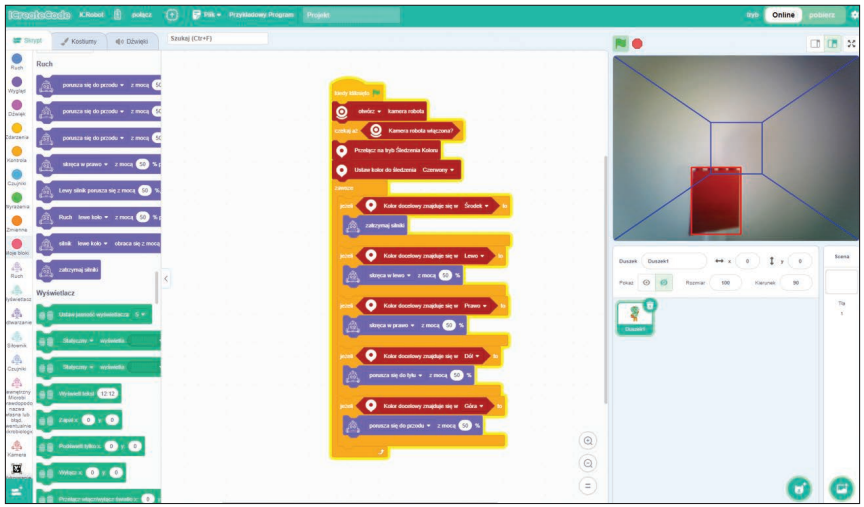
- Uczniowie mogą:
  - połączyć rozpoznawanie koloru z sygnałem dźwiękowym.

1. Model programu

- model algorytmu – schemat myślowy:

```
START
↓
odczytaj kolor z kamery AI
↓
JEŚLI kolor = A → reakcja 1
W PRZECIWNYM RAZIE JEŚLI kolor = B → reakcja 2
W PRZECIWNYM RAZIE JEŚLI kolor = C → reakcja 3
INACZEJ → brak reakcji / stop
↓
powtórz
```

2. Program w Scratch. Śledzenie czerwonego koloru.



3. Python

- Przykład Python:

```python
def main():
    # Kiedy kliknięto (początek programu)

    # Otwórz kamerę robota
    robot.camera.open()

    # Czekaj aż kamera robota włączona
```


Środki dydaktyczne:

- ai robot,
- komputer, laptop, lub tablet z oprogramowaniem do programowania Icrobota (Scratch / Python),
- klocki konstrukcyjne z zestawu robota,
- karty pracy dla uczniów (do skserowania).

Przebieg zajęć

Wprowadzenie do lekcji

1. Kamera AI jako „zmysł wzroku" robota

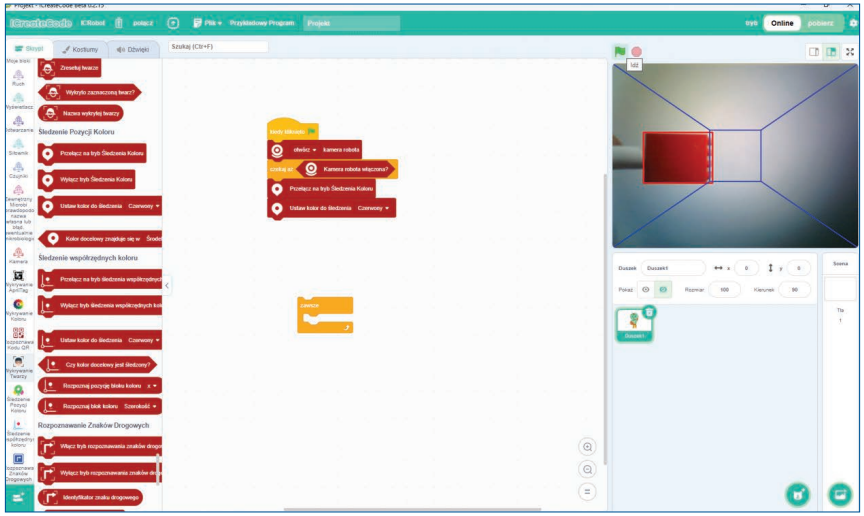
- **Cel etapu:** Uczeń rozumie, że kamera AI pełni funkcję inteligentnego czujnika, a nie zwykłego rejestratora obrazu.
- Nauczyciel wyjaśnia, że robot może „widzieć" otoczenie dzięki kamerze AI, która:
  - zbiera obraz z otoczenia,
  - przetwarza go wewnętrznie,
  - przekazuje do programu gotową informację, np. o wykrytym kolorze.
  - Precyzja działania kamery robota jest bezpośrednio uzależniona od warunków oświetleniowych – odpowiednie światło wpływa na jakość obrazu, a tym samym na poprawność odczytu i interpretacji danych przez kamerę.
- Podkreślenie różnicy:
  - zwykła kamera tylko nagrywa obraz,
  - kamera AI analizuje to, co widzi, i potrafi wyciągać wnioski.
- Uczeń nie musi znać działania algorytmów sztucznej inteligencji, ale powinien rozumieć, że kamera może zostać „nauczona" rozpoznawania określonych cech obrazu.

2. W jaki sposób kamera rozpoznaje kolor?

- **Cel etapu:** Uczeń wie, że kamera AI zamienia obraz na dane, które można wykorzystać w programie.
- Omówienie:
  - kamera rozkłada obraz na małe elementy (piksele),
  - analizuje ich barwy,
  - porównuje je z zapisanymi wcześniej przykładami,
  - przypisuje rozpoznanemu kolorowi określony numer (ID).
- Dzięki temu program robota:
  - nie operuje nazwą koloru,
  - lecz reaguje na otrzymany identyfikator.

- Pokaz nauczyciela:

  Nauczyciel prezentuje przed kamerą kolorowy przedmiot. Kamera rozpoznaje kolor i zaznacza go na ekranie w aplikacji.



3. Po co robotom rozpoznawanie kolorów?

- **Cel etapu:** Uczeń dostrzega praktyczne zastosowanie omawianej technologii.
- Nauczyciel omawia przykłady wykorzystania rozpoznawania barw:
  - automatyczne sortowanie odpadów,
  - roboty poruszające się według kolorowych oznaczeń,
  - systemy kontroli jakości na liniach produkcyjnych,
  - inteligentne zabawki edukacyjne.
- Aktywizacja uczniów:

  Uczniowie podają własne przykłady sytuacji, w których robot mógłby podejmować decyzje na podstawie koloru.
- Pytanie otwarte:

  Jakie zadania mógłby wykonywać robot, gdyby potrafił rozpoznawać kolory?

Etap praktyczny – praca z ai robotem

Cel etapu

Uczeń wykorzystuje kamerę ai robota do stworzenia programu reagującego na rozpoznawane kolory oraz uczy się testować i poprawiać własne rozwiązania.

Etap 1: Przygotowanie robota do pracy

- Uczniowie:
  - uruchamiają ai robota,
  - sprawdzają, czy kamera AI poprawnie wykrywa obraz,
  - upewniają się, że robot reaguje na polecenia z programu.
- Nauczyciel przypomina, że kamera AI jest wbudowana w robota i nie wymaga dodatkowego podłączania.

Etap 2: Nauka kolorów przez kamerę AI

- Uczniowie, korzystając z gotowych narzędzi w środowisku programistycznym:
  - wskazują kamerze wybrane kolory (np. czerwony, zielony, niebieski),
  - sprawdzają, czy kamera poprawnie identyfikuje każdy z nich.
- Celem etapu jest uzyskanie stabilnego i powtarzalnego procesu rozpoznawania barw.

Etap 3: Odczyt informacji z kamery w programie

- Uczniowie tworzą fragment programu, który:
  - pobiera informację o kolorze wykrytym przez kamerę,
  - śledzi poruszający się kolorowy obiekt,
  - umożliwia obserwację, co „widzi" robot.
- Ten etap pozwala uczniom powiązać obraz z danymi wykorzystywanymi w programowaniu.

Etap 4: Programowanie reakcji robota

- Uczniowie rozbudowują program, stosując instrukcje warunkowe:
  - programują ai robota, aby podążał za obiektem o wyznaczonym kolorze,

  Przykładowo:
  - wybierają kolor czerwony, kiedy kolor znajduje się centralnie przed kamerą - robot się nie rusza, jeśli po lewej – skręca w lewo, jeśli po prawej – skręca w prawo, jeśli u góry – jedzie prosto, jeśli u dołu – jedzie do tyłu.

Etap 5: Sprawdzanie i poprawianie programu

- Uczniowie:
  - prezentują przed kamerą różne kolorowe przedmioty,
  - obserwują reakcję robota,
  - analizują, czy program działa zgodnie z założeniami,
  - wprowadzają poprawki w kodzie.
- Nauczyciel zwraca uwagę na wpływ oświetlenia i ustawienia robota na skuteczność rozpoznawania kolorów.