

Program - testowanie czujnika odległości

Poniżej program wykorzystujący czujnik odległości robota mBot do zmiany rozmiaru duszka na ekranie. Na przykład: odległość do przeszkody mała – duszek mały. Odległość rośnie – duszek rośnie.

```
kiedy zielona flaga kliknięta
ustaw rozmiar na 100%
zawsze
    ustaw [odległość] na (czujnik ultradźwiękowy port 3)
    jeśli <(odległość) > 50> wtedy
        ustaw [odległość] na 50
    jeśli <(odległość) < 5> wtedy
        ustaw [odległość] na 5
    ustaw [rozmiar duszka] na ((odległość - 5) * (180 / 45) + 20)
    ustaw rozmiar na (rozmiar duszka)%
```

Zmodyfikuj poprzedni program w taki sposób by przy odległości 25 duszek osiągał rozmiar 100, a przy odległości 50 duszek osiągał rozmiar 200.

```
kiedy zielona flaga kliknięta
ustaw rozmiar na 100%
zawsze
    ustaw [odległość] na (czujnik ultradźwiękowy port 3)
    jeśli <(odległość) > 50> wtedy
        ustaw [odległość] na 50
    jeśli <(odległość) < 25> wtedy
        ustaw [odległość] na 25
    ustaw [rozmiar duszka] na ((odległość - 25) * 4 + 100)
    ustaw rozmiar na (rozmiar duszka)%
```

Program, który pozwoli na sterowanie wysokością dźwięków odtwarzanych przez komputer przy pomocy czujnika odległości robota. Do generowania dźwięków użyj bloków z palety „muzyka”. Musisz ją zainstalować z centrum rozszerzeń.

```
kiedy zielona flaga kliknięta
dodaj rozszerzenie [Muzyka]
zawsze
    ustaw [odległość] na (czujnik ultradźwiękowy port 3)
    jeśli <(odległość) > 50> wtedy
        ustaw [odległość] na 50
    jeśli <(odległość) < 5> wtedy
        ustaw [odległość] na 5
    ustaw [wysokość nuty] na ((odległość - 5) * (40 / 45) + 40)
    zagraj nutę (wysokość nuty) przez (0.3) sekund
    czekaj (0.1) sekund
```

Program - dla autonomicznego robota

Poniżej program dla **mBota**, który pozwoli mu przejechać od jednej linii do drugiej i zatrzymać się tak, aby **kota napędowe znalazły się jak najbliżej drugiej linii** — czyli robot ma zatrzymać się tuż przed przeszkodą (np. linia, ściana, klocek), ale nie najechać na nią.

```
kiedy zielona flaga kliknięta
ustaw prędkość silników na 100
zawsze
    jeśli <(czujnik ultradźwiękowy port 3) > 6> wtedy
        ustaw silnik lewy na 100
        ustaw silnik prawy na 100
    w przeciwnym razie
        zatrzymaj silniki
        zakończ program
```

Rozbuduj program z poprzedniego ćwiczenia w taki sposób, by po osiągnięciu celu robot wrócił na miejsce startu jadąc tyłem. Następnie zawrócił i powtórzył poprzednie zadanie i wrócił przodem.

```
kiedy zielona flaga kliknięta

// 1. Jedź do przodu aż do przeszkody
zawsze
    jeśli <(czujnik ultradźwiękowy port 3) > 6> wtedy
        ustaw silnik lewy na 100
        ustaw silnik prawy na 100
    w przeciwnym razie
        zatrzymaj silniki
        zakończ pętlę
        przejdź do następnego kroku

// 2. Cofnij się przez określony czas
ustaw silnik lewy na -100
ustaw silnik prawy na -100
czekaj (2) sekundy
zatrzymaj silniki

// 3. Zawróć (obróć o 180°)
ustaw silnik lewy na 100
ustaw silnik prawy na -100
czekaj (1) sekunda
zatrzymaj silniki

// 4. Jedź do przodu aż do przeszkody (drugi raz)
zawsze
    jeśli <(czujnik ultradźwiękowy port 3) > 6> wtedy
        ustaw silnik lewy na 100
        ustaw silnik prawy na 100
    w przeciwnym razie
        zatrzymaj silniki
        zakończ pętlę
        przejdź do ostatniego kroku

// 5. Wracaj przodem na miejsce startu
ustaw silnik lewy na -100
ustaw silnik prawy na -100
czekaj (2) sekundy
zatrzymaj silniki
```

Na podłodze bądź macie edukacyjnej wyznacz tor w kształcie kwadratu o boku około 1m. Wytycz bazę (czerwony obszar startu i mety).

Poniżej skrypt, po uruchomieniu którego robot okrąży przeszkodę umieszczoną w środku kwadratu i powróci do bazy.

kiedy zielona flaga kliknięta

```
// Start z bazy
powiedz "Startuję!" przez (2) sekundy

// 1. Pierwszy bok kwadratu
ustaw silnik lewy na 100
ustaw silnik prawy na 100
czekaj (2) sekundy
zatrzymaj silniki

// Skręt w lewo (90°)
ustaw silnik lewy na -100
ustaw silnik prawy na 100
czekaj (0.6) sekundy
zatrzymaj silniki

// 2. Drugi bok
ustaw silnik lewy na 100
ustaw silnik prawy na 100
czekaj (2) sekundy
zatrzymaj silniki

// Skręt w lewo
ustaw silnik lewy na -100
ustaw silnik prawy na 100
czekaj (0.6) sekundy
zatrzymaj silniki

// 3. Trzeci bok
ustaw silnik lewy na 100
ustaw silnik prawy na 100
czekaj (2) sekundy
zatrzymaj silniki

// Skręt w lewo
ustaw silnik lewy na -100
ustaw silnik prawy na 100
czekaj (0.6) sekundy
zatrzymaj silniki

// 4. Czwarty bok
ustaw silnik lewy na 100
ustaw silnik prawy na 100
czekaj (2) sekundy
zatrzymaj silniki

// Powrót do bazy
powiedz "Wróciłem do bazy!" przez (2) sekundy
```

Program - zawody sumo robotów.

Potrzebne będą:

- Co najmniej 2 zespoły - odpowiednio do ilości posiadanych robotów (optymalnie 2 zawodników do 1 robota).
- Ring do zapasów sumo w kształcie koła o średnicy około 80 cm z obwódką w kontrastującym kolorze o szerokości co najmniej 25mm rozłożony na podłodze.

Zasady pojedynku robotów sumo:

- walka polega na wzajemnym spychaniu się z ringu dwóch robotów mBot
- przed rozpoczęciem walki zawodnicy umieszczają na planszy roboty bokiem do siebie w jednakowej odległości od krawędzi ringu (sędzia może zarządzić inne rozmieszczenie robotów)
- na znak sędziego zawodnicy uruchamiają roboty i odchodzą od planszy na odległość około 1m
- robot, który jako pierwszy opuści ring przegrywa walkę (przez opuszczenie ringu rozumiemy sytuację, w której obydwa napędzane koła robota będą znajdować się poza krawędzią ringu)
- sędzia może w dowolnym momencie przerwać walkę i zarządzić jej powtórzenie.

Poniżej program dzięki któremu robot będzie mógł poruszać się po ringu nie wyjeżdżając poza jego krawędź.

Uwaga: robot po uruchomieniu programu przez 5 sekund musi pozostać w bezruchu.

Czujnik linii mBota znajduje się z przodu robota i zwraca wartości:

- **0** – oba czujniki widzą ciemną powierzchnię (np. czarny ring)
- **1** – lewy czujnik widzi jasną powierzchnię
- **2** – prawy czujnik widzi jasną powierzchnię
- **3** – oba czujniki widzą jasną powierzchnię (robot na krawędzi)

kiedy zielona flaga kliknięta

```
// 1. Odliczanie przed startem
powiedz "Start za 5 sekund!" przez (2) sekundy
czekaj (5) sekund

// 2. Walka sumo – poruszanie się po ringu
zawsze
  ustaw [linia] na (czujnik linii port 2)

  jeśli <(linia) = 0> wtedy
    // bezpiecznie na ringu – jedź do przodu
    ustaw silnik lewy na 100
    ustaw silnik prawy na 100

  jeśli <(linia) = 1> wtedy
    // lewy czujnik widzi krawędź – skręć w prawo
    ustaw silnik lewy na 50
    ustaw silnik prawy na -50
    czekaj (0.3) sekundy

  jeśli <(linia) = 2> wtedy
    // prawy czujnik widzi krawędź – skręć w lewo
    ustaw silnik lewy na -50
```

```

ustaw silnik prawy na 50
czekaj (0.3) sekundy

jeśli <(linia) = 3> wtedy
    // oba czujniki widzą krawędź – cofnij się
    ustaw silnik lewy na -100
    ustaw silnik prawy na -100
    czekaj (0.5) sekundy
    // zawróć
    ustaw silnik lewy na 100
    ustaw silnik prawy na -100
    czekaj (0.4) sekundy

```

❓ Dodaj **czujnik odległości**, by robot wykrywał przeciwnika i atakował.

❓ Dodaj **dźwięk startowy** lub **miganie LED** w czasie odliczania.

❓ Dodaj **tryb agresywny**: jeśli przeciwnik jest blisko, robot przyspiesza.

Zaproponuj żeby Twój mBot miał **strategię ataku i obrony**, rozbudujemy program o **czujnik odległości** (ultradźwiękowy), który pozwoli robotowi:

- **Wykrywać przeciwnika** z przodu.
- **Atakować**, gdy przeciwnik jest blisko (np. < 20 cm).
- **Bronić się**, gdy zbliża się do krawędzi ringu (wykrywane przez czujnik linii).

kiedy zielona flaga kliknięta

```

// 1. Odliczanie przed walką
powiedz "Start za 5 sekund!" przez (2) sekundy
czekaj (5) sekund

```

```

// 2. Walka sumo z atakiem i obroną
zawsze
    ustaw [linia] na (czujnik linii port 2)
    ustaw [odległość] na (czujnik ultradźwiękowy port 3)

```

```

// 🛡️ OBRONA – wykryto krawędź ringu
jeśli <(linia) = 1> wtedy
    // lewy czujnik widzi krawędź
    ustaw silnik lewy na 50
    ustaw silnik prawy na -50
    czekaj (0.3) sekundy

```

```

jeśli <(linia) = 2> wtedy
    // prawy czujnik widzi krawędź
    ustaw silnik lewy na -50
    ustaw silnik prawy na 50
    czekaj (0.3) sekundy

```

```

jeśli <(linia) = 3> wtedy
    // oba czujniki widzą krawędź
    ustaw silnik lewy na -100
    ustaw silnik prawy na -100
    czekaj (0.5) sekundy
    // zawróć
    ustaw silnik lewy na 100
    ustaw silnik prawy na -100
    czekaj (0.4) sekundy

```

```
// 🚀 ATAK – przeciwnik blisko
jeśli <<(linia) = 0> i <(odległość) < 20>> wtedy
    ustaw silnik lewy na 100
    ustaw silnik prawy na 100

// 🔍 SZUKAJ – przeciwnik daleko
jeśli <<(linia) = 0> i <(odległość) > 20>> wtedy
    ustaw silnik lewy na 100
    ustaw silnik prawy na -100
```

- 🔍 Dodaj **tryb agresywny**: jeśli przeciwnik jest bardzo blisko (< 10 cm), robot przyspiesza.
- 🔍 Dodaj **efekty LED**: czerwony = atak, niebieski = obrona, zielony – szukanie przeciwnika
- 🔍 Dodaj **licznik czasu walki** lub **punkty**.

kiedy zielona flaga kliknięta

```
// 1. Odliczanie przed walką
powiedz "Start za 5 sekund!" przez (2) sekundy
czekaj (5) sekund

// 2. Ustaw czas walki
ustaw [czas_walki] na 30

// 3. Walka sumo z efektami LED i licznikiem
zawsze
    ustaw [linia] na (czujnik linii port 2)
    ustaw [odległość] na (czujnik ultradźwiękowy port 3)

    jeśli <(czas_walki) = 0> wtedy
        zatrzymaj silniki
        powiedz "Koniec walki!" przez (2) sekundy
        powtórz (5)
            ustaw kolor LED na [biały]
            czekaj (0.2) sekundy
            wyłącz LED
            czekaj (0.2) sekundy
        zakończ program

    zmniejsz [czas_walki] o 1
    czekaj (1) sekunda

// 🛡️ OBRONA – wykryto krawędź ringu
jeśli <(linia) = 1> wtedy
    ustaw kolor LED na [niebieski]
    ustaw silnik lewy na 50
    ustaw silnik prawy na -50
    czekaj (0.3) sekundy

jeśli <(linia) = 2> wtedy
    ustaw kolor LED na [niebieski]
    ustaw silnik lewy na -50
    ustaw silnik prawy na 50
    czekaj (0.3) sekundy

jeśli <(linia) = 3> wtedy
    ustaw kolor LED na [niebieski]
    ustaw silnik lewy na -100
```

```
ustaw silnik prawy na -100
czekaj (0.5) sekundy
ustaw silnik lewy na 100
ustaw silnik prawy na -100
czekaj (0.4) sekundy
```

```
// 🚀 ATAK – przeciwnik blisko
jeśli <<(linia) = 0> i <(odległość) < 20>> wtedy
    ustaw kolor LED na [czerwony]
    ustaw silnik lewy na 100
    ustaw silnik prawy na 100
```

```
// 🔍 SZUKAJ – przeciwnik daleko
jeśli <<(linia) = 0> i <(odległość) > 20>> wtedy
    ustaw kolor LED na [zielony]
    ustaw silnik lewy na 100
    ustaw silnik prawy na -100
```