

Program LineFollower – opis działania

Program realizuje implementację wybranych algorytmów sterowania dla robotów śledzących linię tzw: „Line Followers”. Program przygotowano w języku Python, wykorzystując bibliotekę „pygame”.

Przyjęte modele są uproszczone, podczas sterowania nie uwzględniono min. masy modeli.

Również symulacja odczytu koloru przez detektor jest uproszczona, wielkość punktu mierzonego przez prawdziwy detektor jest stosunkowo niewielka, podczas gdy wielkość punktu pomiaru w programie to jeden piksel (przy 40 pikselach szerokości modelu, i 60 pikselach długości).

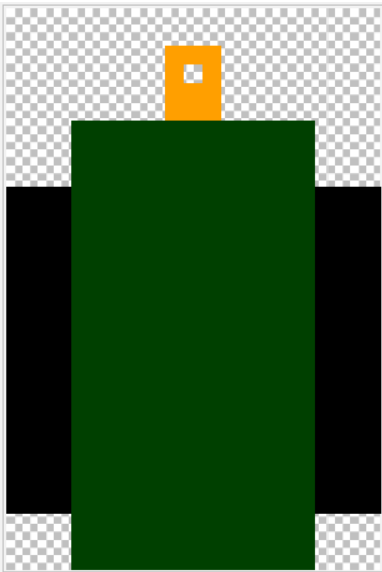
Realność modelu podnosi sposób implementacji ruchu robotów, prawdziwy robot pokonując trasę z punktu do punktu ma możliwość bieżącej korekty w przypadku napotkania linii. W programie roboty poruszają się „skacząc” do nowej pozycji, co może prowadzić do przeoczenia linii której szerokość będzie węższa niż jednorazowo przemierzona odległość. Jest to więc dobre odzwierciedlenie działania czujnika który potrzebuje czasu na przetworzenie odczytanych informacji i w tym czasie nie ma możliwości wykonywania nowych pomiarów. W połączeniu opisane cechy uczyniły symulowane roboty mniej dokładne niż ich realne odpowiedniki i wymogły stosowanie szerszej linii niż ta wykorzystywana podczas zawodów.

Przyjęto jednakowy styl ruchu dla każdego modelu, robot może poruszać się w przód i w tył, a także obracać się w miejscu.

Wszystkie modele zwizualizowano wykorzystując program GIMP.

Do uruchomienia programu wymagany jest język „Python” oraz biblioteki „pygames” oraz „numpy”

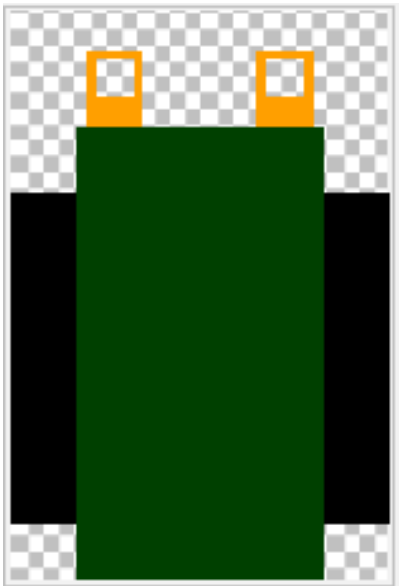
Zaimplementowane typy robotów:



A. Robot z jednym detektorem

Jeden detektor utrudnił skuteczną implementację algorytmu śledzenia linii, dlatego w tym typie robota zaimplementowano algorytm śledzenia krawędzi. Detektor umiejscowiony jest na żółtym wystającym segmencie, dokładny punkt pomiaru to kwadratowe wycięcie.

Rysunek 1. Robot śledzący z jednym detektorem



B. Robot z dwoma detektorami

Kolejny typ robota posiada 2 detektory. Detektory skierowane są pionowo w dół, algorytm zakłada więc że robot jest na właściwej trasie kiedy żaden z detektorów nie wykrywa linii.

Rysunek 2. Robot śledzący z dwoma czujnikami



C. Robot z 7 detektorami

Ostatni robot posiada aż 7 detektorów, jego budowa nawiązuje do wyglądu profesjonalnych robotów wykorzystywanych w zawodach. Do sterowania wykorzystano regulator PD

Rysunek 3. Robot śledzący z 7 czujnikami

Sterowanie w programie

Skrócony opis sterowania:

- Strzałki – sterowanie robotem
- Klawisze 1,2,3 – zmiana typu robota
- Spacja – uruchomienie algorytmu

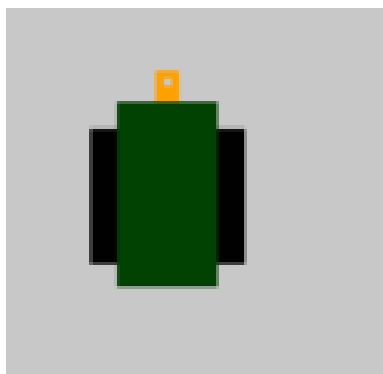
Rozszerzony opis sterowania:

- Każdym robotem można dowolnie sterować przy pomocy klawiatury, z wykorzystaniem strzałek

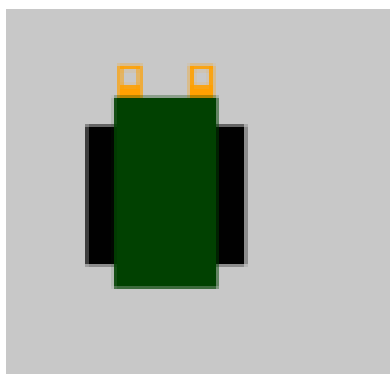


Rysunek 4. Robot typu A, skierowany na środek pola przy użyciu strzałek

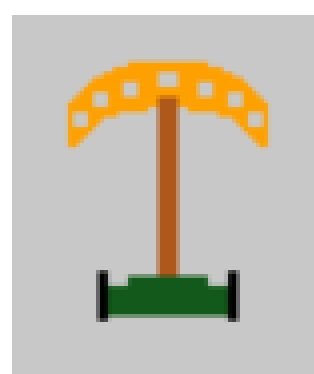
- Klawiszami 1,2 i 3 można przełączać się pomiędzy typami robotów,



Rysunek 7. Typ „A”, klawisz "1"



Rysunek 6. Typ „B”, klawisz "2"



Rysunek 5. Typ „C”, klawisz 3

-ustawi też domyślną dla nich szerokość linii

- Wciśnięcie lewego klawisza myszy spowoduje przeniesienie kursora na początek trasy i rozpoczęcie rysowania



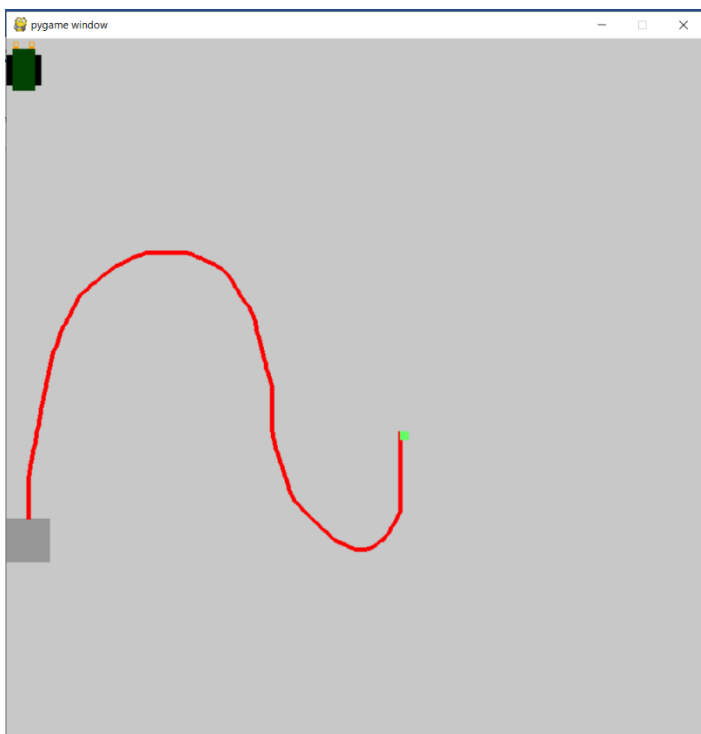
Rysunek 8. Rysowanie trasy dla robota

- Zwolnienie lewego przycisku myszy przeniesie robota na obszar startowy i rozpocznie algorytm



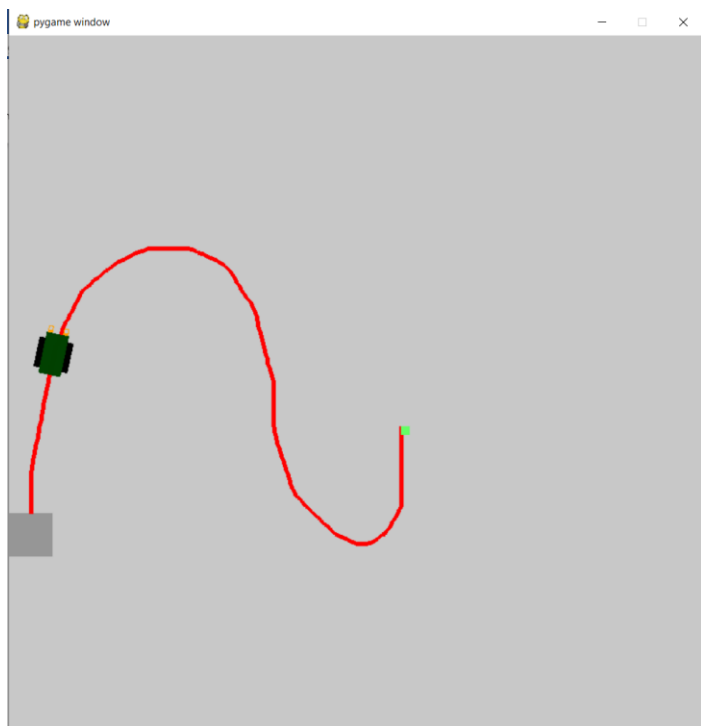
Rysunek 9. Robot typu "C" podczas pokonywania trasy

- Zmiana typu robota po narysowaniu trasy pozwoli na przetestowanie różnych robotów na tej samej trasie



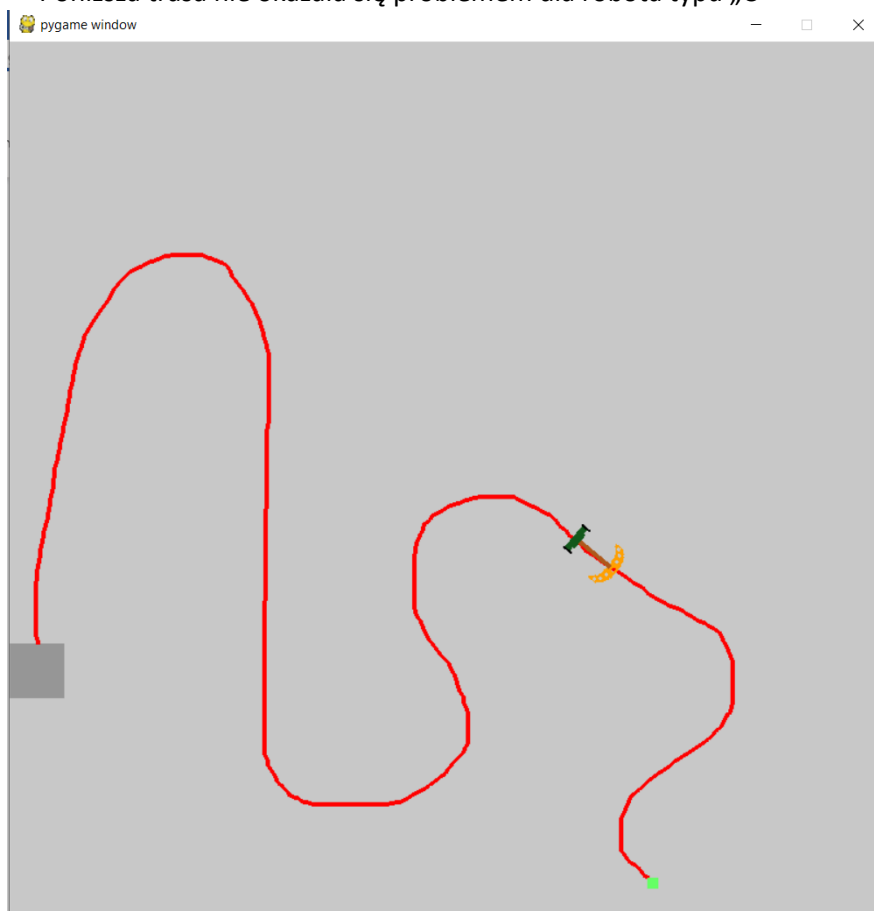
Rysunek 10. Robot typu "B" z trasą robota typu "C" z rysunku 9

- Aby uruchomić robota dla tej samej trasy należy wcisnąć klawisz spacji



Rysunek 11. Robot typu "B" uruchomiony na trasie z rysunku 9

Poniższa trasa nie okazała się problemem dla robota typu „C”



Sprawiła natomiast problem robotowi typu „B”

