
Scenariusz lekcji

Ozobot w klasie: Spacer losowy po układzie współrzędnych

Opracowanie scenariusza: Richard Born

Adaptacja scenariusza na język polski: mgr Piotr Szlagor

Tematyka: Informatyka, Matematyka, Losowość, Algorytm, Funkcje

Grupa wiekowa: Gimnazjum

Czas trwania: 45 minut

Punkty podstawy programowej:

INFORMATYKA - III etap edukacyjny

Treści nauczania - wymagania szczegółowe

5. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego. Uczeń:

- 1. wyjaśnia pojęcie algorytmu, podaje odpowiednie przykłady algorytmów rozwiązywania różnych problemów;*
- 2. formułuje ścisły opis prostej sytuacji problemowej, analizuje ją i przedstawia rozwiązanie w postaci algorytmicznej;*
- 3. stosuje arkusz kalkulacyjny do rozwiązywania prostych problemów algorytmicznych;*
- 4. opisuje sposób znajdowania wybranego elementu w zbiorze nieuporządkowanym i uporządkowanym, opisuje algorytm porządkowania zbioru elementów;*
- 5. wykonuje wybrane algorytmy za pomocą komputera.*

6. Wykorzystywanie komputera oraz programów i gier edukacyjnych do poszerzania wiedzy i umiejętności z różnych dziedzin. Uczeń:

- 1. wykorzystuje programy komputerowe, w tym edukacyjne, wspomagające i wzbogacające naukę różnych przedmiotów;*
- 2. wykorzystuje programy komputerowe, np. arkusz, kalkulacyjny, do analizy wyników eksperymentów, programy specjalnego przeznaczenia, programy edukacyjne;*
- 3. posługuje się programami komputerowymi, służącymi do tworzenia modeli zjawisk i ich symulacji, takich jak zjawiska: fizyczne, chemiczne, biologiczne, korzysta z internetowych map;*

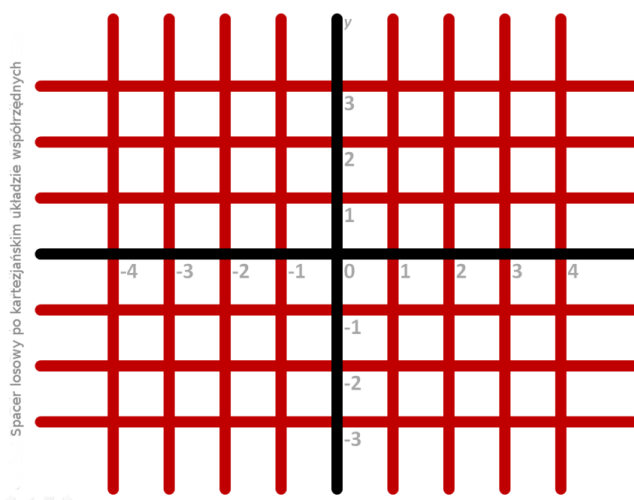
Przed przystąpieniem do pracy:

- Naładuj w pełni swojego Ozobota,
- Ustaw maksymalną jasność ekranu, by robot nie miał problemów z pobraniem programu ze strony *ozoblockly.com*,
- Skalibruj Ozobota do swojego ekranu.

Wstęp

Spacer losowy to ścieżka, która powstaje poprzez wykonywanie losowych decyzji co do kierunku, gdy tylko napotykamy na drodze jakąś możliwość wyboru (np. natrafiamy na skrzyżowanie). Przykładowo, gdybyś w czasie swojego spaceru rzucał monetą i w zależności od wyniku podążał na skrzyżowaniu w lewo lub w prawo, to wykonywałbyś bardzo prosty spacer losowy. Spacer losowy można wykonywać na jednowymiarowych osiach, dwuwymiarowych płaszczyznach (np. na kartezjańskim układzie współrzędnych), czy nawet w trzech wymiarach. Zagadnienie to jest badane przez naukowców związanych z wieloma dziedzinami nauki. W ekonomii używamy spacerów losowych do modelowania cen akcji, w fizyce do lepszego przybliżania ruchów cząsteczek w gazach i cieczech, a w informatyce choćby do szacowania rozmiaru sieci WWW.

Ozobota można bardzo łatwo zaprogramować do wykonywania spaceru losowego. A oglądanie jak go przemierza jest bardzo ciekawe. Może zdarzyło ci się pracować z programem do takich spacerów, gdy przeglądałeś przykłady zamieszczone na stronie *ozoblockly.com*. Robot w czasie jego wykonywania poruszał się, dowolnie losując sobie kierunek, w którym następnie podążał. Teraz, twoim zadaniem będzie takie zaprogramowanie Ozobota, żeby w losowy sposób przemieszczał się po kartezjańskim układzie współrzędnych.

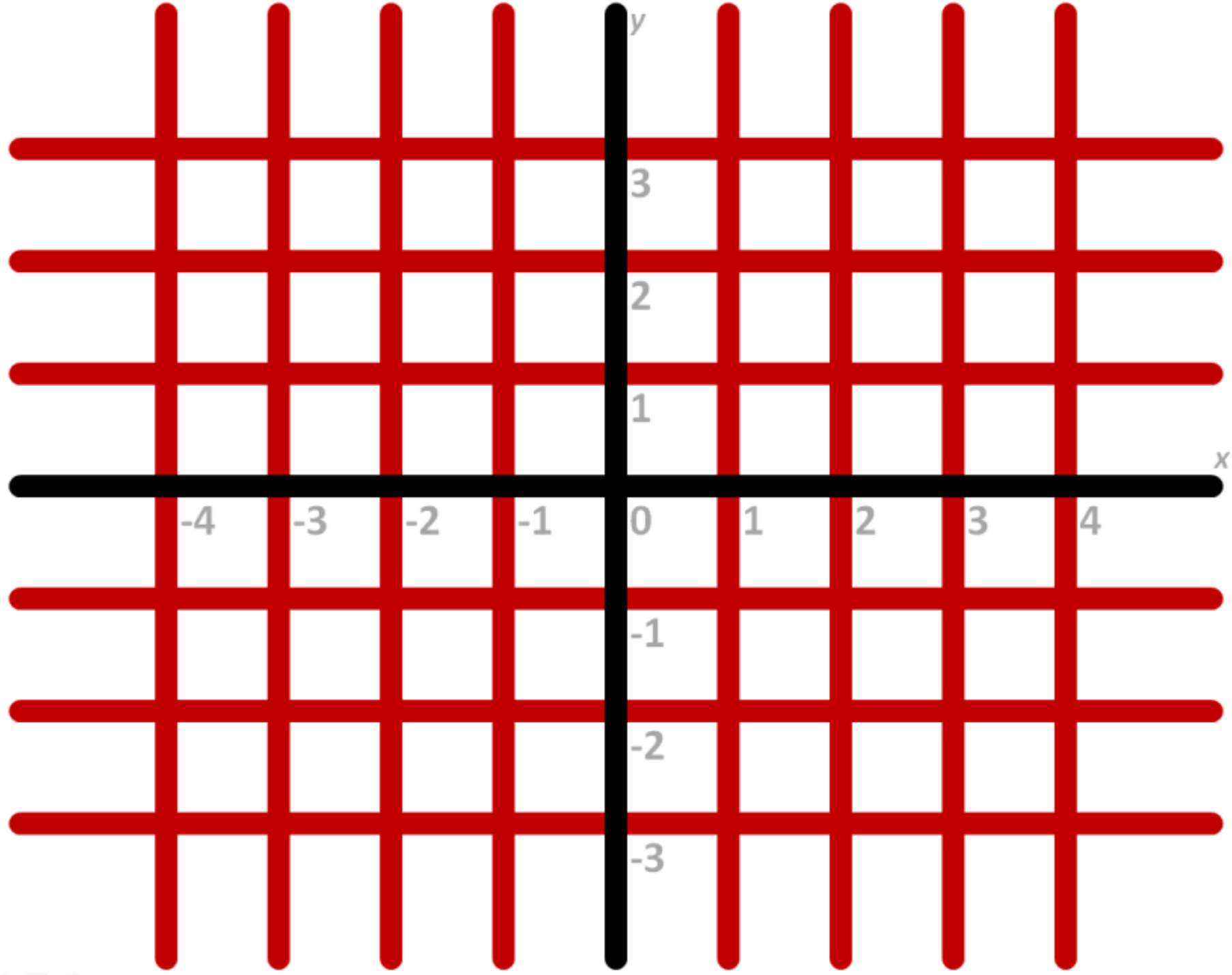


Stwórz z bloczków na stronie ozoblockly.com program, który wykona 40 losowych kroków na kartezjańskim układzie współrzędnych. Plansza jest dostępna do wydrukowania na następnej stronie. Jego mniejsza wersja, przydatna w czasie studiowania wymagań programu znajduje się powyżej.

Wymagania programu

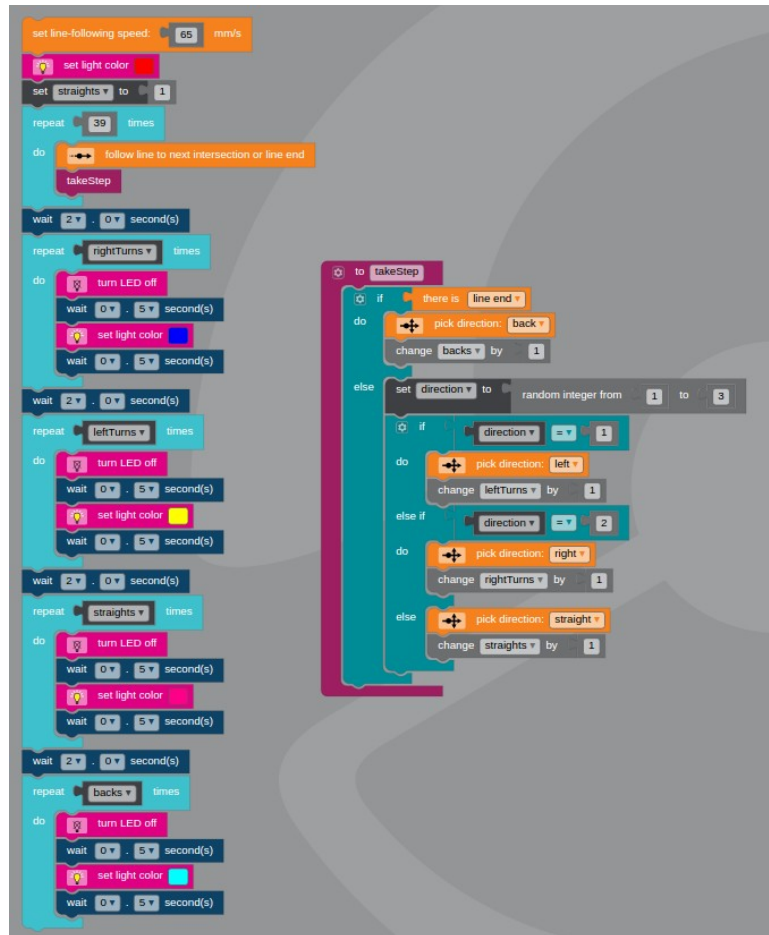
1. Użyj *Mode 4 (Advanced)*, możliwego do wyboru na stronie ozoblockly.com, aby wykonać to zadanie.
2. Ozobot powinien poruszać się z prędkością opisaną jako *fast* (65 mm/s), ale zaleca się, żeby w czasie tworzenia programu ustawić dla niego mniejszą prędkość.
3. Upewnij się, że Ozobot jest poprawnie skalibrowany na kartce, zanim uruchomisz program.
4. Miejscem startowym, na którym będziesz ustawiał robota, powinno za każdym razem być początkiem układu współrzędnych, czyli punktem (0,0). Przód Ozobota powinien być ustawiać w stronę dodatnich liczb na osi y (do góry).
5. Wszystkie kroki powinny być o długości jednej jednostki.
6. Pierwszy ruch Ozobota powinien być wykonany z punktu (0,0) do punktu (0,1). Kolejne 39 kroków powinno się losować, gdy robot będzie napotykał na skrzyżowania. Powinien wówczas wybrać, czy pójdzie w prawo, w lewo lub prosto, przy czym każdemu z tych wyborów należy nadać jednakowe prawdopodobieństwo zajścia.
7. Jedynym momentem, kiedy Ozobot będzie się cofał, będzie wówczas gdy dotrze do którejś z krawędzi naszego labiryntu.
8. Gdy robot zakończy wykonywanie spaceru losowego, powinien natychmiast się zatrzymać, a następnie „wymigać” liczbę skrętów w prawo, w lewo, przejazdów prosto i cofnięć, które wykonał w czasie swojej trasy.
 1. Liczba skrętów w prawo powinna zostać pokazana poprzez mrugnięcie NIEBIESKIM światłem taką liczbę razy, jaka wystąpiła w czasie spaceru.
 2. Liczba skrętów w lewo powinna zostać pokazana poprzez mrugnięcie ŻÓŁTYM światłem taką liczbę razy, jaka wystąpiła w czasie spaceru.
 3. Liczba przejazdów prosto przez skrzyżowanie powinna zostać pokazana przez mrugnięcia w kolorze FIOLETOWYM
 4. Liczba cofnięć powinna być pokazana w kolorze CZERWONYM.
9. Pamiętaj, żeby sprawdzić, czy liczba mrugnięć Ozobota jest równa liczbie 40.
10. Po wykonaniu wszystkich mrugnięć, Ozobot powinien się wyłączyć.

Spacer losowy po kartezjańskim układzie współrzędnych



Krótko o kodzie

Kod całego programu, widoczny na Rysunku 1. Składa się z dwóch sekcji. Pierwsza, widoczna po lewej, to główna część skryptu. Tam na początku ustawiamy prędkość na 65 mm/s, kolor na diody czerwony i wtedy następuje wywołanie funkcji o nazwie *takeStep*, powtarzanej w pętli 39 razy. Program przenosi wtedy swoje działanie do grupy bloków widocznej po prawej. Funkcja *takeStep* ma dwa podstawowe działania. Po pierwsze sprawdza, czy robot nie próbuje wyjechać poza planszę oraz losuje kierunek przejazdu przez skrzyżowanie. Po drugie: zlicza wszystkie kierunki, w jakie udał się Ozobot i zapisuje je do zmiennych *backs*, *leftTurns*, *rightTurns* i *straights*.



Rysunek 1

Po wykonaniu pętli *repeat*, robot przejdzie do wyświetlania danych dotyczących kierunków przejazdu, za pomocą „mrugania” diodą LED. Każda operacja taka jest opisana pętlą z taką liczbą powtórzeń, jaka została zapisana w jeden ze zmiennych *backs*, *leftTurns*, *rightTurns* i *straights*.

Cały kod programu można podejrzeć i załadować do Ozobota, włączając w przeglądarce internetowej stronę <http://ozoblockly.com/editor?mode=4#26wm2e> lub skanując kod widoczny po prawej..

