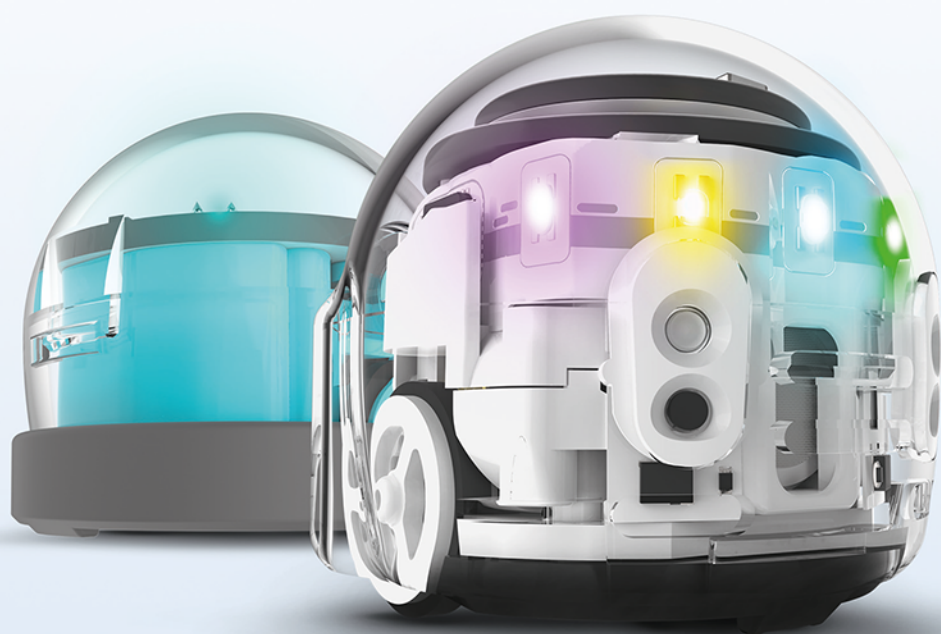


Dorota Kowalczyk

PROGRAMOWANIE
z **ozobotem**
KOLOROWE KODY



Helion 

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiegokolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie książki na nośniku filmowym, magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi bądź towarowymi ich właścicieli.

Autorka oraz Helion SA dołożyli wszelkich starań, by zawarte w tej książce informacje były kompletne i rzetelne. Nie biorą jednak żadnej odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub autorskich. Autor oraz Helion SA nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Redaktor prowadzący: Szymon Sz wajger
Korekta: Jarosław Lipski

Projekt okładki: Studio Gravite
Grafika na okładce została wykorzystana za zgodą Ozobot.com

Helion SA
ul. Kościuszki 1c, 44-100 GLIWICE
tel. 32 231 22 19, 32 230 98 63
e-mail: helion@helion.pl
WWW: <http://helion.pl> (księgarnia internetowa, katalog książek)

Drogi Czytelniku!
Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres
<http://helion.pl/user/opinie/proozo>
Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Pliki z materiałami pomocniczymi do książki znaleźć można po adresem:
<ftp://ftp.helion.pl/przyklady/proozo.zip>

ISBN: 978-83-283-6653-4

Copyright © Helion SA 2020

Printed in Poland.

- [Kup książkę](#)
- [Poleć książkę](#)
- [Oceń książkę](#)

- [Księgarnia internetowa](#)
- [Lubię to! » Nasza społeczność](#)

Spis treści

Wstęp	5
Rozdział 1. Tworzenie trasy	15
Rozdział 2. Kodowanie trasy	31
Rozdział 3. Kody i polecenia	43
Rozdział 4. Od pomysłu do programu — cz. I	91
Rozdział 5. Ozobot i akcesoria	115
Rozdział 6. „Witaj, świecie!”, czyli wprowadzenie do środowiska OzoBlockly	131
Rozdział 7. Od pomysłu do programu — cz. II	185
Rozdział 8. Czym jest programowanie... ..	231
I co dalej z Ozobotem?	275

ROZDZIAŁ 2.

Kodowanie trasy

Wprowadzenie

W rozdziale 1. obserwowaliśmy, jak Ozobot porusza się po trasie „według własnego widzimisię”, czyli pseudolosowo. Dojeżdżając do skrzyżowania, Ozobot „sprawdza”, jakie są dostępne drogi, i wybiera jedną z nich. Dopóki nie zakodujemy trasy zgodnie z rozwiązywanym zadaniem, Ozobot będzie wybierał drogę dowolnie i niekoniecznie zgodnie z naszymi oczekiwaniami.

Do kodowania trasy służą trzy rodzaje kolorowych kodów: kody dwukolorowe, kody trójkolorowe i kody czterokolorowe. Każdy rodzaj jest przypisany do określonej grupy czynności.

I tak:

- kody dwukolorowe służą do zakończenia pracy Ozobota (rysunek 2.1);

Rysunek 2.1.

Kod dwukolorowy



- kody trójkolorowe służą do określania kierunków jazdy, szukania drogi, zmiany prędkości (rysunek 2.2);

Rysunek 2.2.

Kod trójkolorowy



- kody czterokolorowe służą do uruchamiania liczników i poleceń specjalnych (rysunek 2.3).

Rysunek 2.3.

Kod czterokolorowy



Ozobot rozpoznaje kody w trzech kolorach: czerwonym, zielonym i niebieskim. Każdy kod to sekwencja tych kolorów, czasem w połączeniu z kolorem czarnym. Ozobot w trakcie przejeżdżania nad obszarem kodu czyta po kolei kolory i wykonuje dane polecenie zgodnie z przeczytaną sekwencją. Ponieważ Ozobot czyta kod w trakcie jazdy, oczywiste jest, że ten kod będzie przeczytany zgodnie z kierunkiem poruszania się Ozobota. Musimy wobec tego pamiętać, że aby Ozobot poprawnie wykonał polecenie, kod musi być umieszczony właśnie zgodnie z kierunkiem jazdy. Jeżeli umieścimy kod odwrotnie, to albo Ozobot go zignoruje, albo wykona zupełnie inne polecenie przypisane takiemu kodowi. Na rysunkach od 2.4 do 2.7 zaprezentowano przykłady:

Rysunek 2.4.

Polecenie Na następnym skrzyżowaniu skręć w prawo

**Rysunek 2.5.**

Brak polecenia, Ozobot zignoruje kod

**Rysunek 2.6.**

Polecenie Tornado

**Rysunek 2.7.**

Polecenie Rotacja



Dotychczas ćwiczyliśmy rysowanie tras. Najistotniejszym elementem była ich szerokość. Bez względu na to, czy trasę rysujemy mazakami, czy korzystamy z programu graficznego, jej szerokość powinna wynosić 5 mm. Kolorowe kody — również bez względu na to, czy naklejane, czy malowane — nie powinny przekraczać szerokości trasy, ale w miarę dokładnie się w tę trasę wpasować. Zatem pojedyncze pole kodu powinno być kwadratem o wymiarach 5mm×5mm. W przypadku naklejek czy puzzli ta zasada jest automatycznie zachowana przez producenta, zatem nie musimy już dbać o prawidłowe techniczne skonstruowanie kodu. Analizę kodowania trasy i ewentualnych innych ograniczeń zaczniemy właśnie od tego łatwiejszego sposobu.

Kodowanie z użyciem naklejek

Naklejki z kodami są bardzo wygodnym środkiem wykorzystywanym przy kodowaniu tras. Są prawidłowo zaprojektowane pod względem technicznym, ponadto są wzbogacone o opisy i wskazanie kierunku jazdy Ozobota, co utrudnia popełnienie błędu.

Naklejki OzoEasy

Pierwsza wersja naklejek — OzoEasy — to efekt pracy pewnego młodego człowieka, który postanowił uprościć sposób kodowania tras dla Ozobota właśnie poprzez użycie naklejek. W założeniu miały być to naklejki wielokrotnego użytku, niestety finalnie nie do końca się udało. Te naklejki nie są już produkowane, zostały zastąpione innym rozwiązaniem (producenta Ozobota), niemniej wciąż na rynku można znaleźć zestawy naklejanych kodów z serii OzoEasy w relatywnie niskiej cenie.

OzoEasy były dostępne w kilku wariantach:

- wariant podstawowy (starter),
- wariant „prędkość i kierunek”,
- wariant „liczniki + efekty specjalne + zakończenia”,
- wariant „kropki”,
- wariant „stwórz własne naklejki”.

Naklejki OzoEasy są niewielkie, okrągłe i obejmują swoim obszarem jedynie KOD, bez końców, które ułatwiają wyrównanie z trasą. Zaletą jest to, że możemy przykleić je zarówno do papieru, jak i do folii. Wadą, że... nie zawsze możemy je odkleić. W trakcie odklejania naklejki się rozwarstwiają, na trasie pozostają resztki kleju, w związku z czym trasa może się stać niezdatna do ponownego użytku.

Możemy użyć naklejek razem z folią (celofanem). Pierwszym sposobem jest umieszczenie kart z trasą w celofanowym woreczku.

Takie rozwiązanie z pewnością umożliwi wielokrotne korzystanie z przygotowanej planszy. Ale... karta nie jest ściśle dopasowana do woreczka, zatem istnieje niebezpieczeństwo, że w trakcie pracy się przesunie, a więc naklejony kod nie musi (a powinien) dokładnie pasować do trasy.

Ale i ten problem możemy rozwiązać, przygotowując tabliczkę, czyli umieścić w woreczku kartę z trasą i piankową podkładkę. Taka tabliczka poza tym, że będzie trwała, będzie również ściśle pasowała do karty z trasą i uniemożliwi jej przesuwanie. Takie rozwiązanie jest może trochę bardziej pracochłonne

i niekoniecznie efektywne w przypadku nauczania domowego, kiedy uczestnikiem zajęć jest jedno dziecko, ale doskonale sprawdzi się w szkole. Instruktor może przygotować sobie zestaw takich tabliczek i korzystać z nich w trakcie różnych zajęć, z różnymi grupami. Nie będzie wówczas potrzeby każdorazowego kopiowania/kserowania kart pracy dla uczestników. Aczkolwiek problem z rozwarstwianiem się naklejek może pozostać...

Przetestujmy naklejki na puzzlach.

Sam proces naklejania i odklejania przebiega bez problemów. Naklejka nada się do kolejnego użytku raz, ewentualnie dwa razy, nie więcej. Niemniej jeśli mamy dostęp do tego typu naklejek, to możemy wykorzystać je w celach czysto ćwiczeniowych, by zademonstrować uczestnikom sposób pracy z naklejkami. Warunkiem wykonania takiego ćwiczenia jest wcześniejsze przygotowanie trasy bez względu na to, jaki wariant wybierzemy, bo jeśli trasa będzie narysowana odręcznie (mazakami), to też możemy skorzystać z naklejek, ale nie ma możliwości wielokrotnego użytku ani trasy, ani naklejek.

Naklejki producenta Ozobota

Producent Ozobota proponuje dwa rodzaje naklejek — naklejki do papieru i naklejki do folii. Naklejki do papieru można kupić w oddzielnym zestawie

Naklejki do folii dołączane są do zestawów startowych z Ozobotem. Obydwa rodzaje tych naklejek różnią się od naklejek OzoEasy — są większe, mają kształt prostokąta, posiadają końce ułatwiające dopasowanie naklejki do czarnej trasy. Producent uwzględnił również pewne ograniczenia dotyczące przyklejania naklejek (rysunek 2.8). Niekiedy nie da się zastosować określonego typu naklejki z określonym typem podłoża (na przykład folia+ papier).

Naklejki-kropki

Mogłoby się zdawać, że naklejki-kropki to jest właśnie to, co będzie najbardziej przydatne i efektywne. Przede wszystkim dlatego, że można samodzielnie skonstruować dany kod. Niestety przyklejenie kropek jest zadaniem relatywnie trudnym, biorąc pod uwagę konstrukcję robota i sposób czytania kodu. Nie możemy nakleić kropek ściśle sąsiadująco ani nie możemy nakleić kropek „zachodzących” na siebie zbyt daleko. Musimy najpierw wykonać kilka ćwiczeń z dowolnym kodem, przetestować różne układy z Ozobotem, by znaleźć sposób, który zawsze da oczekiwany rezultat.

Zadaniem Ozobota jest poprawne wykonanie zakodowanego polecenia.

Popelniając błąd techniczny przy konstruowaniu tego kodu z kropek, uniemożliwimy robotowi wykonanie zadania. Naklejki-kropki są dostępne jedynie w serii OzoEasy. Można oczywiście zamówić takie naklejki (w bardzo niskiej cenie) lub przygotować je samodzielnie. Takie naklejki bardzo dobrze współpracują zarówno z papierem, jak i z folią (biorąc pod uwagę, że przy odklejaniu mogą się rozwarstwić jak pozostałe naklejki). Dodatkowym problemem może być odcień koloru zastosowany przy naklejkach — Ozobot czyta określone odcienie czerwonego, zielonego i niebieskiego. Przygotowując własne naklejki, zawsze musimy o tym pamiętać.

Kodowanie z użyciem mazaków

Jak w poprzedniej części zaczniemy od czystej kartki i jednego mazaka — czarnego (rysunek 2.8). Narysujemy obojętnie jaką trasę, ale zostawmy „puste” fragmenty (rysunek 2.9). Można oczywiście rysować od razu trasę gotową — wówczas musimy przygotować sobie wszystkie mazaki i używać tego, który aktualnie będzie nam potrzebny, ale dużo prościej jest rysować trasę dwuetapowo, bo wówczas w pierwszym etapie możemy skupić się wyłącznie na kształcie trasy, a dopiero w drugim na jej kodowaniu.



Rysunek 2.8.
Zaczynamy



Rysunek 2.9.
Narysowana trasa z pustymi fragmentami

Ten przykład dotyczy kodowania przypadkowego. Nie stosujemy na razie żadnych konkretnych kodów poleceń — po prostu wypełniamy puste miejsca sekwencjami kolorów czerwony-zielony-niebieski (dwa, trzy lub cztery) i możemy przetestować, co zrobi robot po przeczytaniu danego, utworzonego w ten sposób „kodu” (rysunek 2.10). Cóż... przeważnie... nic.



Rysunek 2.10.

Wypełnianie pustych miejsc sekwencjami kolorów

Mała jest szansa, że akurat „trafimy” w sekwencję, która dla robota ma jakieś określone znaczenie. W związku z tym spróbujemy wykorzystać naklejki. Nie analizujemy na razie, jaka sekwencja oznacza dany kod. Wiemy, że naklejki są skonstruowane prawidłowo, zatem robot po przeczytaniu kodu z naklejki po prostu musi wykonać jakąś czynność. Jaką? Mamy informację na naklejce. Robot może zmienić kierunek jazdy lub prędkość (sprawdzamy, czy właściwie), może wykonać jakiś efekt specjalny... Do przetestowania naklejek wraz z mazakiem użyjemy tylko jednego, dwustronnego kodu czerwony-zielony-czerwony-zielony. Taki kod przy przejeździe w jedną stronę ma spowodować, że robot wykona *Tornado*, natomiast przy przejeździe w kierunku przeciwnym — *Rotację*. Możemy wykorzystać albo naklejki producenta Ozobota, albo naklejki OzoEasy. Kod namalujemy za pomocą mazaków. WAŻNE — kolory mazaków muszą być bardzo nasycone!

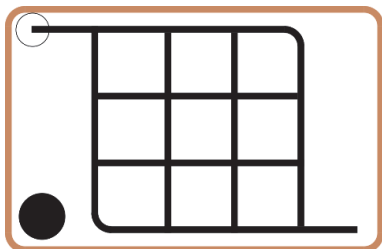
A może użyć tabliczki z trasą przygotowaną w rozdziale 1.? Albo może narysować zupełnie nową trasę? W obu przypadkach postępujemy identycznie:

- przygotowujemy trasę
- naklejamy czyste naklejki (najlepiej czteroelementowe, by mieć możliwie jak największy wybór przy kodowaniu)
- zamalowujemy pola na naklejkach, tworząc dowolne czteroelementowe (czterokolorowe) sekwencje
- sprawdzamy, które sekwencje skutkują jakąkolwiek zmianą w trakcie poruszania się robota pot rasie.

Jak widać, możliwości jest kilka — gotowe naklejki i dowolna trasa, naklejki do samodzielnego przygotowania i gotowa trasa lub rysowanie (czy naklejanie) kodów w trakcie przygotowywania trasy.

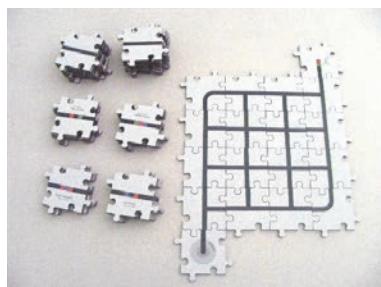
Kodowanie z użyciem kodujących puzzli

Rozpoczniemy od przygotowania prostej trasy — kratownicy 3×3 z oznaczonym startem (rysunek 2.11). Zbudujemy tę trasę, wykorzystując puzzle. Ponieważ trasa ma być zakodowana, oddzielimy puzzle kodujące (kierunek w prawo, kierunek w lewo, kierunek prosto), użyjemy również puzzli *Start* i *Koniec* (rysunek 2.12).



Rysunek 2.11.

Trasa — kratownica 3×3



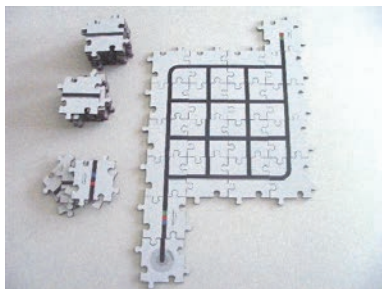
Rysunek 2.12.

Przygotowanie do pracy

Zakodujemy trasę tak, by Ozobot przejechał od startu do końca w możliwie najprostszy sposób. Wykorzystamy do tego celu naklejki:

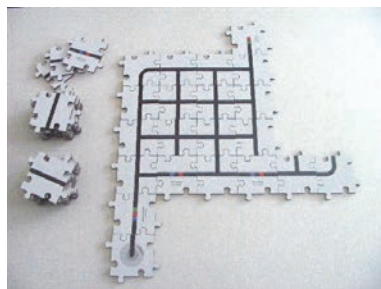
- *Jedź w prawo* $\times 1$,
- *Jedź prosto* $\times 5$,
- *Koniec trasy* $\times 1$

Z użyciem naklejek rozwiązanie takiego zadania jest bardzo proste. Spróbujmy zrobić dokładnie to samo, ale za pomocą kodujących puzzli. Najpierw wybierzemy właściwe puzzle kodujące. Dołożymy do nich również puzzle z czarnymi pustymi trasami. Kolejno umieścimy kodujące puzzle na właściwych miejscach (rysunki od 2.13 do 2.14).



Rysunek 2.13.

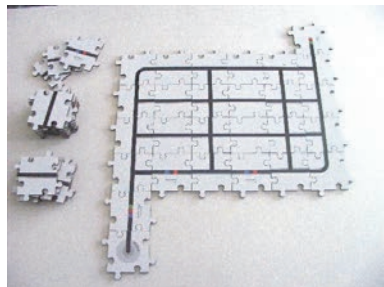
Puzzle trafiają na właściwe miejsca (1)



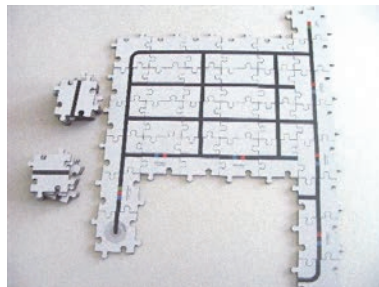
Rysunek 2.13.

Puzzle trafiają na właściwe miejsca (2)

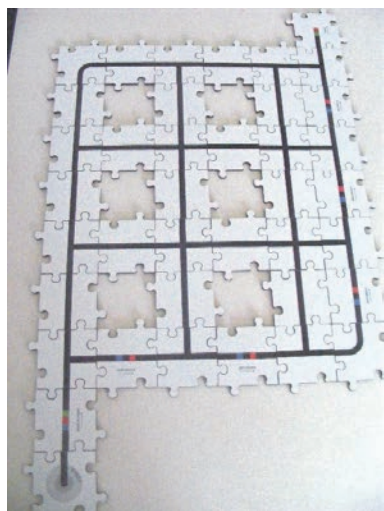
I pierwszy problem — trasę trzeba przebudować. W dodatku dość istotnie... Właśnie dlatego, wybierając puzzle, nie ograniczamy się do kodujących, ale dokładamy do nich czyste proste trasy (rysunek 2.15). Teoretycznie niewiele się zmieniło, niemniej to już nie jest trasa, którą ułożyliśmy na początku. W dodatku zaczyna się powiększać, a to przecież dopiero połowa pracy. Częściowe jej wyniki zaprezentowano na rysunkach od 2.16 do 2.17. Uzupełnijmy zatem wszystkie brakujące pola. W takim kształcie trasa jest gotowa do przetestowania. Możemy też rozbudować ją tak, by przypominała trasę źródłową (rysunek 2.18).



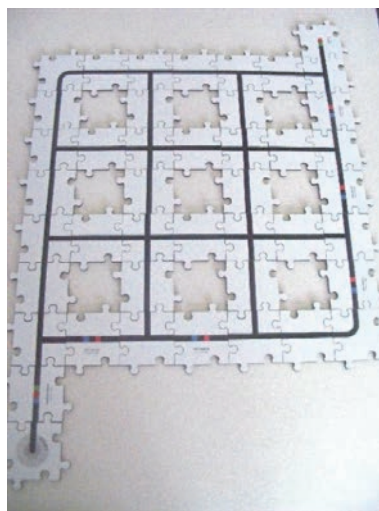
Rysunek 2.15.
Dokładanie czystych tras



Rysunek 2.16.
Rozbudowa trasy



Rysunek 2.15.
Koniec rozbudowy trasy w podstawowej wersji



Rysunek 2.18.
Trasa po modyfikacjach

No i taki trochę anegdotyczny życiowy problem, czyli „Ja chcę dokładnie takie, a nie inne!”.

Stworzyliśmy pewną trasę. Po coś zrobiliśmy. Trasa jest prosta, oczywista, może służyć do wykonania wielu różnych ćwiczeń... I nagle się okazuje, że jeśli chcemy w celu uatrakcyjnienia zajęć wykorzystać puzzle, to tę trasę możemy wyrzucić i całą

pracę zacząć od nowa. W efekcie otrzymujemy planszę niby podobną, tyle że dwa razy większą... Zatem czas wykonania zadania przez robota (empirycznie, czas przejazdu) będzie prawie dwa razy dłuższy... Ćwiczenie, które możemy wykonać w kilka minut, może nam zająć np. połowę zajęć albo więcej. Warto?

Na koniec sprawdzimy jeszcze, czy rzeczywiście Ozobot nie czyta kodów umieszczonych na kolorowych ścieżkach. W tym celu zbudujmy prostą, liniową trasę i naklejmy dowolne kody. Wykonajmy również taki sam eksperyment z kodującymi puzzlami.

Warto taki układ przetestować kilka(naście) razy. Przekonamy się wówczas, że robot czasem wykona polecenie, a czasem nie. Odpowiedź na pytanie dlaczego, będzie poniżej...

Podsumowanie

Po dwóch rozdziałach dotyczących podstawowych dwóch elementów, czyli tworzenia trasy i jej technicznego kodowania, trzeba spróbować podsumować — JAK stworzyć dobrą trasę i JAK ją prawidłowo zakodować, by nie nastroczała problemów przy wykonywaniu konkretnego zadania.

Najpierw musimy wziąć pod uwagę możliwości finansowe. Nie każda placówka posiada puzzle czy naklejki. Zaczniemy zatem od puzzli. Czy w ogóle warto z nich korzystać... Oczywiście puzzle same w sobie są znakomite do zabawy i ćwiczenia przy okazji koordynacji wzrokowo-ruchowej, małej motoryki, spostrzegawczości... Każda placówka posiada w świetlicy czy bibliotece zestawy puzzli, które służą właśnie do takich celów. Ale te akurat puzzle mają służyć też do czegoś innego. I niestety generują dwa istotne problemy, które mogą uniemożliwić prawidłowe wykorzystanie:

- Są nierównej wysokości, robot się zacina na krawędziach.
- Specyfikacja jest mało praktyczna — puzzle kodujące w zasadzie są zbędne, natomiast brakuje puzzli z czystymi trasami, ponadto organizacja pudełka jest mało oszczędna — by znaleźć określony kafelek, trzeba czasem naprawdę poświęcić dużo czasu.

Jak widać na wyżej zamieszczonym przykładzie, zbudowana trasa znakomicie się sprawdzi z naklejkami, natomiast w ogóle się nie sprawdzi z kodującymi puzzlami, bo trzeba tę trasę gruntownie przebudować. Czy posiadamy odpowiednie stanowiska pracy, by móc w trakcie wykonywania zadania rozbudowywać trasę? Czy to ma jakikolwiek sens? I czy nie szkoda na to czasu? Pomijam fakt, że trasa ułożona z puzzli

jest nierówna nie tylko w pionie, ale również w swej ciągłości. Na szczęście to nie sprawia problemów robotowi, niemniej jest po prostu mało estetyczne.

Teraz naklejki — są bardzo dobrym rozwiązaniem. Można je zastosować zarówno z papierem, jak i z folią (z puzzlami też). Można dokonać wyboru — czy użyjemy czystych naklejek, które „wypełnimy” za pomocą mazaków, czy użyjemy naklejek gotowych, bo np. zadanie jest trudniejsze i warto przeznaczyć dostępny czas na analizę i poszukiwanie właściwego rozwiązania, a nie na techniczne „malowanie” kodów. Naklejki pozostawiają instruktorowi dużą swobodę wyboru. Można nawet przygotować własne naklejki, jeśli akurat nie mamy dostępu do naklejek aktualnie oferowanych na rynku.

Kolejnym etapem podsumowania będą wnioski wynikające z doświadczeń przy projektowaniu trasy i umieszczaniu kodów:

- **NIE POWINNIŚMY** tworzyć na trasie zbyt ostrych zakrętów. Jeśli trasa wymaga ostrego zakrętu, to trzeba najpierw przetestować kilkakrotnie, jak robot się na takim zakręcie zachowa. Czy się zatrzyma, nie zawiesi...
- Kolorowe fragmenty trasy nie mogą być krótsze niż 2,5 cm. Jeśli z jakichś powodów musimy część trasy pokolorować (linią ciągłą), kolorowy fragment powinien być dłuższy niż cal (2,5 cm).
- Jeśli planujemy kodowanie trasy (a nie tylko jej utworzenie, żeby „robot sobie pojeździł”), to nie wolno nam tworzyć skrzyżowań z liczbą dróg większą niż 4. Robot bowiem dojeżdża do skrzyżowania (pierwsza możliwość dojazdu) i może pojechać w prawo (druga możliwość dojazdu), w lewo (trzecia możliwość dojazdu) lub na wprost (czwarta możliwość dojazdu).
- Nie umieszczamy kodów zbyt blisko innych istotnych elementów trasy, takich jak skrzyżowania, zmiana koloru linii lub inny kod. Producent Ozobota sugeruje, by odległość między dwoma takimi elementami była nie mniejsza niż 1 cal. Zdarzają się sytuacje, że trasa dostosowana do konkretnego zadania wymusza bliższe umieszczenie elementów, ale każdorazowo trzeba się liczyć z trudnościami w realizacji kodu. Ozobot Bit ma średnicę właśnie ok. 2,5 cm. By wykonać zakodowane polecenie, robot musi przeczytać kod prawidłowo w całości, stąd ta „rezerwa” miejsca na poprawne odczytanie i wykonanie kodu.
- Nie umieszczamy kodów na łukach.
- Nie umieszczamy kodów na skrzyżowaniach.
- Nie umieszczamy kodów na kolorowych trasach, tylko na czarnych.

- Kodując przy użyciu mazaków, należy pamiętać, by między poszczególnymi polami kodu nie było przerw. Ponadto pola muszą być wypełnione kolorem precyzyjnie, kolory nie powinny się na siebie nakładać. Standardowy rozmiar pojedynczego pola kodu to 5mm×5mm bez względu na to, czy kodujemy trasę odręcznie, czy wypełniamy gotowe pola na naklejkach (lub nadrukowane na trasie).

Każde odstępstwo od powyższych zasad nie musi od razu oznaczać fiaska rozwiązania zadania. Ale jeśli w trakcie realizacji okaże się, że robot nie reaguje zgodnie z oczekiwaniami, zawsze najpierw warto sprawdzić, czy dany kod został umieszczony prawidłowo.

Biorąc pod uwagę wszystkie te ograniczenia, musimy racjonalnie zdecydować, jaką formę tworzenia trasy i jej zakodowania wybierzemy. Najbardziej praktycznym rozwiązaniem wydaje się korzystanie z tras gotowych (w formie kserokopii czy wydruków), folii jako nakładki i naklejek wielokrotnego użytku. Trasy możemy przygotować samodzielnie w dowolnym programie do tworzenia i obróbki grafik wektorowych. Podobnie naklejki — skorzystamy z gotowych (bądź wypełnionych, bądź do wypełnienia z wykorzystaniem mazaków) lub przygotujemy je samodzielnie i wydrukujemy na właściwym papierze. Taki wybór znacznie oszczędzi czas przygotowania uczestników do rozwiązania zadania i pozwoli skupić się na tym, co jest istotne — na programowaniu robota, by wykonał określone zadanie.

PROGRAM PARTNERSKI

— GRUPY HELION —



1. ZAREJESTRUJ SIĘ
2. PREZENTUJ KSIĄŻKI
3. ZBIERAJ PROWIZJĘ

Zmień swoją stronę WWW w działający bankomat!

Dowiedz się więcej i dołącz już dzisiaj!

<http://program-partnerski.helion.pl>

GRUPA
Helion

PROGRAMOWANIE z **Ozobotem** KOLOROWE KODY



Przez zabawę
do nauki programowania

Ozobot to niewielki, ale bardzo sprytny robot, którego pokochały dzieci oraz docenili rodzice i nauczyciele informatyki. Ten edukacyjny automat, dwukrotny zdobywca polskiego tytułu Zabawki Roku oraz laureat robotycznych konkursów w USA, ma misję. Każdego chętnego, który ukończył piąty rok życia, gotów jest zabrać w podróż pełną radosnego rysowania, ciekawych problemów do rozwiązania i pracy zespołowej. Ozobot, bawiąc, uczy programowania. Dzieci bowiem komunikują się z nim za pomocą kolorowych kodów — robot odczytuje je i wykonuje dawane mu polecenia.

Nim jednak zajmiemy młodych adeptów informatyki pracą z Ozobotem, musimy dziecko najpierw odpowiednio przygotować. Jak to zrobić? Najlepiej z tą książką! Jej autorka w przystępny sposób tłumaczy zasady komunikacji z robotem, w tym analizowanie i testowanie konkretnych przypadków, które mogą użytkownikowi sprawiać problemy w codziennej pracy z Ozobotem. Nie zagłębiając się w zbędne detale, wyjaśnia najważniejsze dla nauczycieli i rodziców kwestie dotyczące budowy tego niedużego robota o wielkich możliwościach. Proponuje także przystosowanie jej do realizacji własnych pomysłów i podpowiada, jak się do tego zabrać.

Helion 



helion.pl



HELION SA
ul. Kościuszki 1c
44-100 Gliwice
tel.: 32 230 98 63
helion@helion.pl

Sprawdź nasze szkolenia!



SZKOLENIA

AKADEMIA IT & BUSINESS

HELIONSZKOLENIA.PL

KOD KORZYŚCI
Sięgnij po więcej! ▶



ISBN 978-83-283-6653-4



9 788328 366534

INFORMATYKA W NAJLEPSZYM WYDANIU

Cena: 44,90 zł