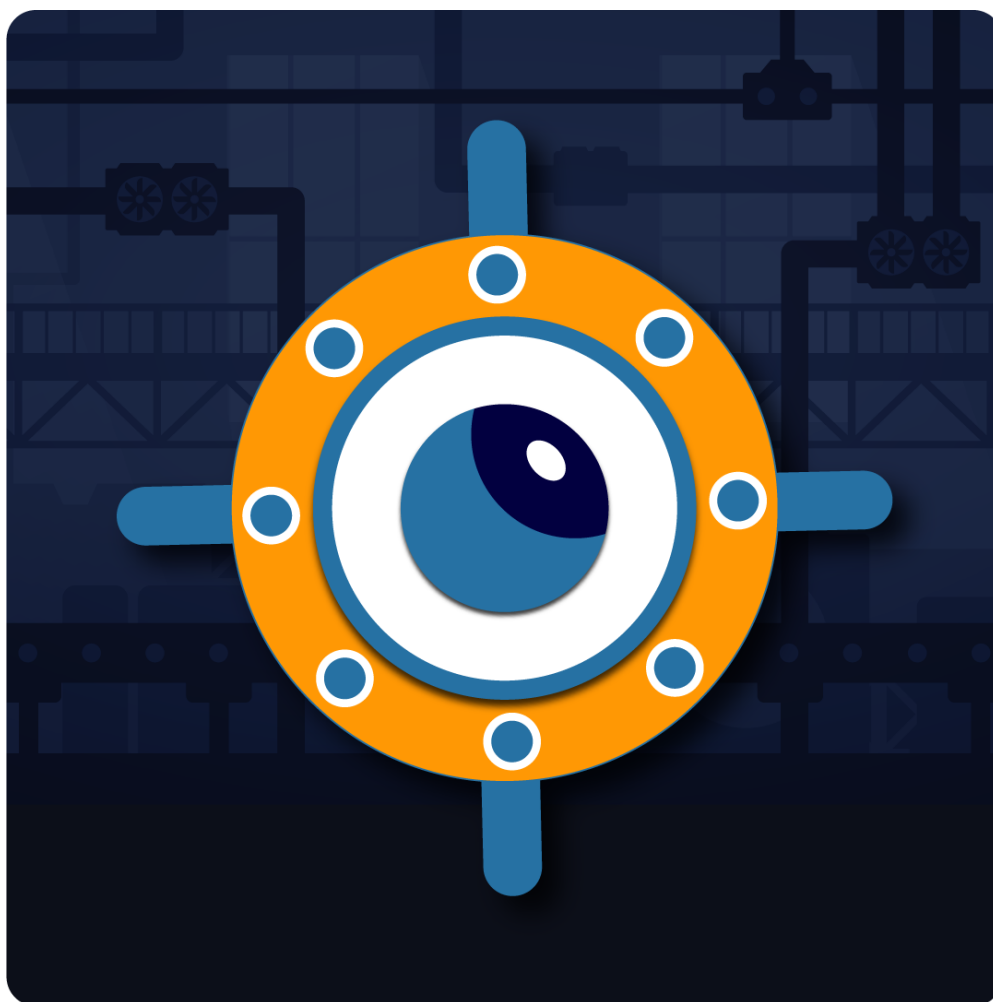


Jedź Lumi!

Kurs kodowania na podłodze interaktywnej SmartFloor



Opracowanie aplikacji i kursu: LavaVision
31-866 Kraków, Skarżyńskiego 5/07
biuro@lavavision.pl

Data wydania aplikacji: 2021-03-15

Aplikacja dostępna na www.motioncube.io

Dostępna dla podłogi interaktywnej SmartFloor™ www.smartfloor.edu.pl
oraz dla urządzeń: tablice interaktywne, ekrany dotykowe.

Dystrybucja w Polsce: Mentor Systemy Audiowizualne
Będargowo 29a, 72-005 Przecław
telefon: +48 (91) 488 51 51
e-mail: kontakt@smartfloor.edu.pl



Spis treści:

Opis kursu Jedź Lumi!	6
Cele kursu	6
Podstawowe informacje	6
Słownik	7
Założenia i ograniczenia	8
Zakres kursu	9
Sekwencje / Tworzenie sekwencji instrukcji	9
Warunki / Instrukcje warunkowe	11
Akcje / Podnieś i upuść	14
Pętle iteracyjne / Powtarzanie sekwencji w pętli	16
Funkcje / Definiowanie funkcji	18
Zasoby programu nauczania	19
Język programowania	19
Ogólna koncepcja programowania na podłodze interaktywnej	19
Składnia i semantyka języka programowania w aplikacji Jedź Lumi!	20
Składnia - zasady tworzenia sekwencji instrukcji	20
Semantyka - znaczenie poszczególnych instrukcji	21
Praca z aplikacją Jedź Lumi!	24
Struktura kursu	31
Scenariusze zajęć	33
Lekcja 1: Tworzenie sekwencji instrukcji	33
Cele operacyjne	33
Opis skrótowy	34
Cele ćwiczeń	34
Plan zajęć	34
Rozgrzewka	34
Dyskusja: Zasady zachowywania się w pracowni interaktywnej	34
Zabawa: Mucha wyszła	34
Nowe pojęcia	35
Ćwiczenia główne	35
Jedź Lumi! Poziom 1-10	35
Podsumowanie	37
Nauczanie rozszerzone	37
Przygotowanie do zajęć	37
Lekcja 2: Dodajemy warunek na trasie	38
Cele operacyjne	38
Opis skrótowy	38
Cele ćwiczeń	38

Plan zajęć	38
Rozgrzewka	39
Zabawa: Jestem robotem	39
Dyskusja	39
Nowe pojęcia	39
Ćwiczenia główne	39
Jedź Lumi! Poziom 11-20	39
Podsumowanie	41
Nauczanie rozszerzone	41
Przygotowanie do zajęć	41
Lekcja 3: Metoda podnieś i upuść	42
Cele operacyjne	42
Opis skrótowy	42
Cele ćwiczeń	42
Plan zajęć	43
Rozgrzewka	43
Zabawa: Układanie puzzli	43
Zabawa: Jestem robotem	43
Zabawa: Kodowanie graficzne	43
Dyskusja	44
Nowe pojęcia	44
Ćwiczenia główne	44
Jedź Lumi! Poziom 21-30	44
Podsumowanie	46
Nauczanie rozszerzone	46
Przygotowanie do zajęć	47
Lekcja 5: Wprowadzamy pętlę iteracyjną	47
Cele operacyjne	47
Opis skrótowy	47
Cele ćwiczeń	47
Plan zajęć	48
Rozgrzewka	48
Zabawa: Jestem robotem	48
Zabawa: Zapętlone Koderki	48
Nowe pojęcia	49
Ćwiczenia główne	49
Jedź Lumi! Poziom 41 - 50	49
Podsumowanie	51
Nauczanie rozszerzone	51
Przygotowanie do zajęć	51

Lekcja 8: Tworzymy funkcję	52
Cele operacyjne	52
Opis skrótowy	52
Cele ćwiczeń	52
Plan zajęć	52
Rozgrzewka	53
Zabawa: Śpiewamy piosenkę	53
Nowe pojęcia	53
Ćwiczenia główne	53
Jedź Lumi! Poziom 71 - 80	53
Podsumowanie	55
Nauczanie rozszerzone	55
Przygotowanie do zajęć	56

Opis kursu *Jedź Lumi!*

Cele kursu

- Realizacja treści z zakresu programowania komputerowego i rozwiązywania problemów algorytmicznych z wykorzystaniem podłogi interaktywnej.
- Tworzenie optymalnych algorytmów do sterowania wirtualnym bohaterem na planszy za pomocą sekwencji instrukcji.
- Rozwijanie rozumienia struktury i składni kodu programu.
- Wsparcie aktywności ucznia kształtującej umiejętności korzystania z rozwijających się umysłowych procesów poznawczych, takich jak myślenie logiczne, analityczne, algorytmiczne.
- Kształtowanie orientacji przestrzennej.
- Stymulowanie umiejętności planowania i przewidywania, myślenia przyczynowo-skutkowego.
- Tworzenie warunków nauczania pozwalających na naukę poprzez eksperymentowanie.
- Wspieranie sprawności motorycznych i sensorycznych uczniów w czasie zorganizowanych bezpiecznych zajęć.
- Tworzenie warunków nauczania sprzyjających współpracy i rozwijaniu umiejętności komunikacji społecznej uczniów.

Podstawowe informacje

- Aplikacja “Jedź Lumi!” przeznaczona jest na podłogę interaktywną z obsługą pisaków świetlnych
- Kurs dedykowany jest uczniom w wieku szkolnym (7-14 lat).
- Aplikacja zawiera 110 poziomów - interaktywnych plansz do rozwiązania za pomocą sekwencji instrukcji do sterowania bohaterem.
- Instrukcje dla bohatera mają postać wirtualnych bloczków obrazkowych.
- Kurs podzielony jest na 11 lekcji z dedykowanymi 10 kolejnymi poziomami do rozwiązania.
- Stopień trudności ćwiczenia wzrasta z każdym kolejnym poziomem.
- Brak konieczności liczenia pojedynczych kroków bohatera. Bohater - wirtualny robot jedzie po ścieżce zgodnie z zadanym kierunkiem, aż natknie się na przeszkodę.

- Ilość miejsca na kod jest ograniczona i może być inna dla różnych poziomów. Należy zmieścić się z rozmiarem algorytmu w danej długości.
- Bohater może zbierać przedmioty na trasie w dowolnej kolejności i ilości. Może je odkładać również w dowolnej kolejności. Należy upewnić się, że odkładając przedmiot do kontenera, robot posiada przedmiot z danej kategorii.

Słownik

Słownictwo informatyczne zostało wyjaśnione szerzej w rozdziale *Zakres programu nauczania*. Poniższa tabela prezentuje pojęcia i skróty stosowane w dokumencie.

Skrót / pojęcie	Wyjaśnienie/ Definicja
akcja	Zdefiniowane zdarzenie / czynność bohatera na planszy, taka jak <i>Podnieś</i> lub <i>Upuść</i> .
algorytm	Lista kroków (sekwencja) potrzebna do wykonania zadania przez bohatera
aplikacja	Aplikacja interaktywna <i>Jedź Lumi!</i> , inaczej gra.
bohater	Wirtualny robot o imieniu Lumi, bohater aplikacji <i>Jedź Lumi!</i>
debugowanie	Proces poszukiwania i usuwania błędów w algorytmie.
funkcja	Sekwencja instrukcji, którą można zdefiniować jeden raz i wywoływać wiele razy w programie.
gra	Aplikacja interaktywna <i>Jedź Lumi!</i> , inaczej aplikacja.
kod	Kod źródłowy / kod źródłowy programu / sekwencja instrukcji w algorytmie.
pętla	Pętla iteracyjna, w której wykonujemy sekwencję instrukcji kilka razy z rzędu.
pisak	Pisak świetlny / pisak interaktywny, za pomocą którego obsługujemy aplikację na podłodze interaktywnej.
podłoga interaktywna	Urządzenie projekcyjne wyświetlające na podłodze obszar interaktywny, reagujący na ruch lub pisaki świetlne.
poziom	Zadanie / plansza / poziom trudności w aplikacji <i>Jedź Lumi!</i>
program	Algorytm, który został zakodowany w czymś, co może być uruchomione przez maszynę cyfrową.
programowanie	Proces tworzenia programu.
robot	Wirtualny bohater w aplikacji <i>Jedź Lumi!</i>

sekwencja	Lista instrukcji.
warunek	Instrukcja, która wykonuje się tylko w określonych warunkach.

Założenia i ograniczenia

W Tabeli: *Założenia i ograniczenia* wymienione zostały kryteria przyjęte dla zaprojektowanego kursu kodowania.

Tabela: Założenia i ograniczenia

Parametr / Założenie / Ograniczenie	Wartość
Wymagana umiejętność czytania u ucznia	NIE
Wymagana umiejętność liczenia do 10	TAK
Wiek uczniów	7-14 lat
Zalecana obecność nauczyciela podczas zajęć	TAK
Wymagane specjalistyczne przygotowanie nauczyciela	NIE
Praca indywidualna z uczniem	TAK
Praca nauczyciela z grupą uczniów	TAK
Praca uczniów w parach lub grupach	TAK
Możliwość samodzielnego korzystania z aplikacji interaktywnej przez ucznia	TAK
Możliwość dostosowania gotowych scenariuszy zajęć przez nauczyciela do potrzeb uczniów	TAK
Jedno lub więcej rozwiązań dla zadania.	TAK
Praca przy stanowisku komputerowym	NIE
Przygotowane środowisko do interaktywnego nauczania: podłoga interaktywna SmartFloor™ z zainstalowanym oprogramowaniem MotionCube™ - aplikacją <i>Jedź Lumi!</i> Pisaki interaktywne - 2 sztuki (np. 1 długi, 1 krótki)	TAK

Zakres kursu

Sekwencje / Tworzenie sekwencji instrukcji

Cele operacyjne, uczeń:

- określa kierunki i ustala położenie bohatera w stosunku do własnej osoby, a także w stosunku do innych przedmiotów, rozróżnia stronę lewą i prawą, górę i dół,
- rozumie pojęcia pion i poziom,
- układa sekwencję instrukcji dla wirtualnego robota za pomocą bloczków ze strzałkami kierunkowymi,
- odczytuje sekwencję instrukcji zapisaną za pomocą bloczków ze strzałkami kierunkowymi,
- omija przeszkody w postaci ścian labiryntu.

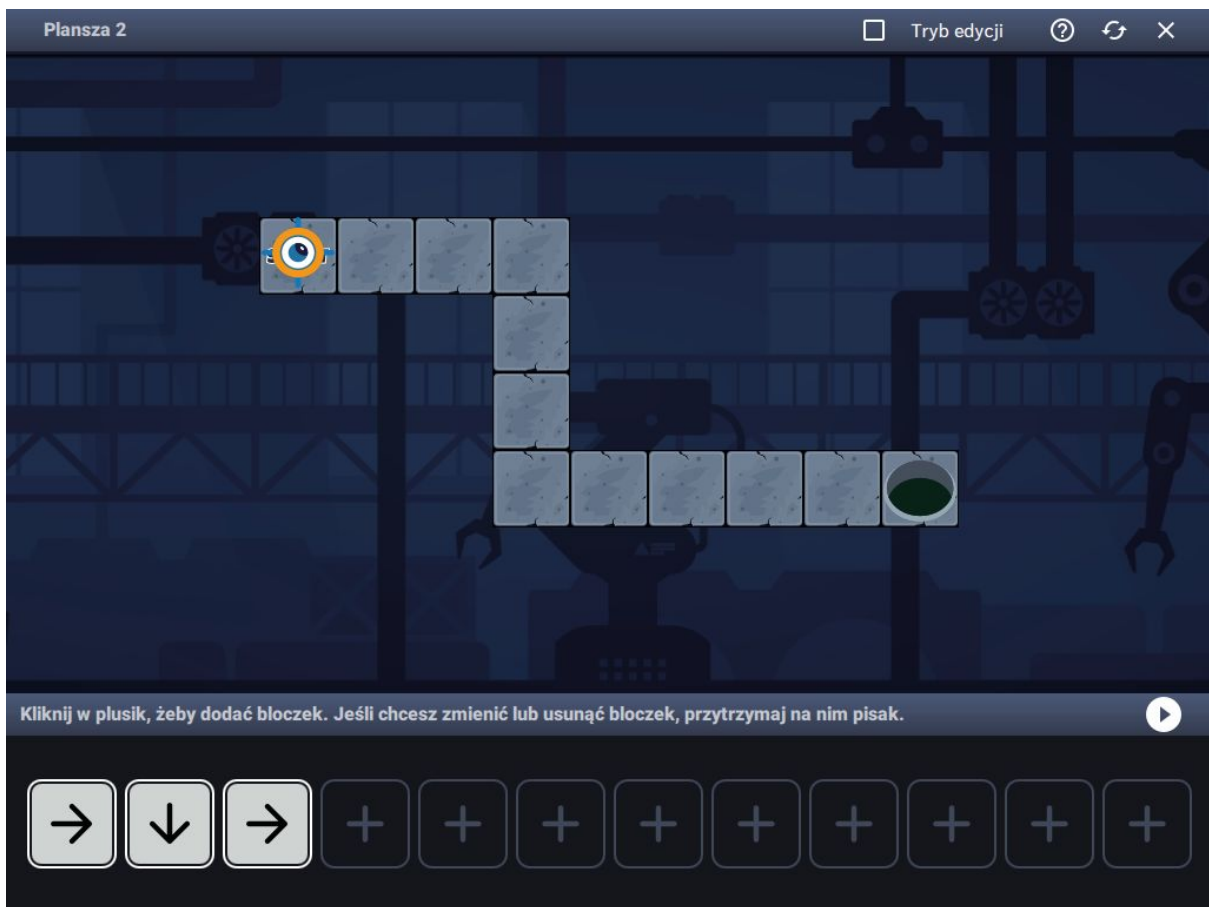
Tworzenie kodu sekwencyjnego należy do podstawowych technik projektowania algorytmów. Kiedy rozwiązujemy problem, w którym jesteśmy w stanie przewidzieć wszystkie sytuacje, które mogą się wydarzyć oraz rozwiązać zadanie w postaci kolejno następujących po sobie instrukcji programu, wówczas najbardziej podstawową techniką jest zapisanie wszystkich czynności, które muszą po sobie nastąpić. Takie postępowanie nazywa się tworzeniem sekwencji instrukcji kodu, w skrócie tworzeniem sekwencji. Podejście to jest wystarczające dla wielu prostych zadań algorytmicznych, składających się z kilku lub kilkunastu instrukcji do wykonania. Podejście to jest również łatwo przyswajalne dla dzieci w wieku przedszkolnym, ponieważ można je odnieść do sytuacji znanych dziecku z życia codziennego, np. wyjście na spacer, gotowanie zupy itp.

W aplikacji *Jedź Lumi!* sekwencje instrukcji są konstruowane poprzez układanie w jednej poziomej linii bloczków z instrukcjami nawigującymi robotem - *jedź w prawo, jedź w lewo, jedź do góry, jedź do dołu*. Uruchomienie poprawnie przygotowanej sekwencji ruchów robota pozwoli na doprowadzenie robota od pola startowego, czyli początkowego położenia bohatera na planszy) do pola mety, którym w aplikacji jest otwór, czasami zakryty włazem.

W ramach kursu *Jedź Lumi!* jest przygotowanych dziesięć ćwiczeń wprowadzających (poziomy 1-10), a łącznie jest sto dziesięć zadań, w których tworzymy sekwencje instrukcji dla robota, stosując różne techniki programowania, których uczymy się podczas tego kursu. Poziom trudności ćwiczeń wzrasta z każdym kolejnym zadaniem.

Przykładowe zadanie o niskim stopniu skomplikowania pokazano na Rysunku: Sekwencje - poziom 2.

Rysunek: Sekwencje - Poziom 2



W powyższym zadaniu sekwencja składa się z trzech instrukcji, które kolejno wskazują robotowi kierunek jazdy. Zadanie ma jedno poprawne rozwiązanie.

Przykładowe zadanie o wyższym stopniu skomplikowania pokazano na Rysunku: Sekwencje - poziom 10.

Rysunek: Sekwencje - Poziom 10



W powyższym rozwiązaniu sekwencja kodu składa się z jedenastu instrukcji. Zadanie ma więcej niż jedno poprawne rozwiązanie.

Warunki / Instrukcje warunkowe

Cele operacyjne, uczeń:

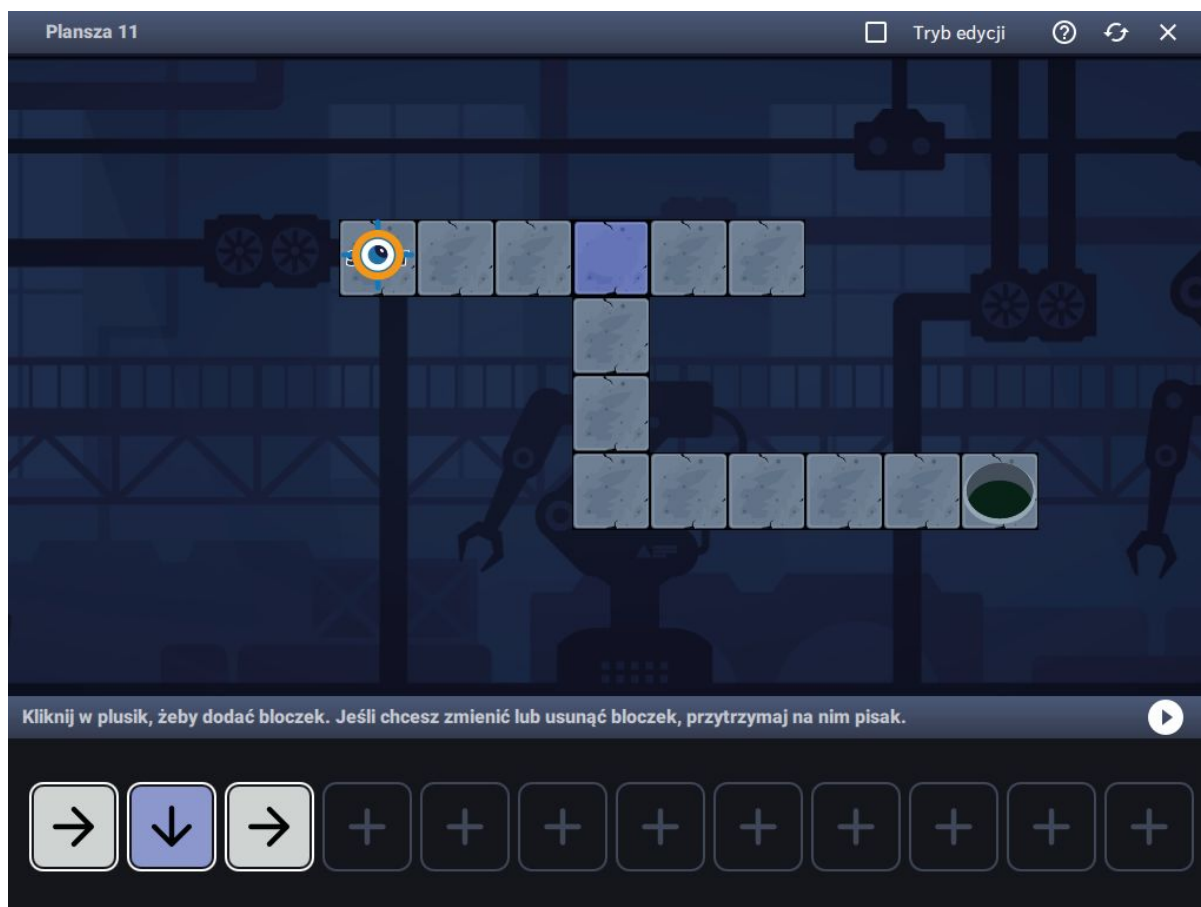
- stosuje w sekwencji instrukcję warunkową *“jeśli jesteś na polu oznaczonym kolorem niebieskim, to zmień kierunek jazdy”*.

Instrukcje warunkowe (w skrócie Warunki) pozwalają na uruchomienie określonych fragmentów kodu (sekwencji instrukcji) tylko, jeśli wystąpi określona sytuacja, tj. warunek zostanie spełniony. Warunki w językach programowania mają postać dodatkowych instrukcji kodu typu *jeżeli, to* (ang. *If, then*), które należy umieścić w projektowanym algorytmie. Dla instrukcji warunkowej należy określić warunek, który ta instrukcja będzie sprawdzać. Jeśli warunek będzie spełniony, to zostanie wykonana pewna sekwencja kodu, którą należy zdefiniować w ramach zakresu warunkowego, ograniczonego początkiem i końcem zakresu.

Ze względu na specyfikę ruchu robota w aplikacji *Jedź Lumi!*, który po otrzymaniu instrukcji jazdy w określonym kierunku, kieruje się zawsze do końca korytarza, zmiana kierunku w innych miejscach (np. w odgałęzieniach w środku korytarza) jest możliwa dzięki użyciu instrukcji warunkowych. W tym celu wprowadzono na planszach pola decyzyjne, tzw. pola warunkowe, na których może być podjęta decyzja o zmianie kierunku jazdy. Instrukcje te mają postać wyrażenia “*zmień kierunek jazdy, jeśli jesteś na polu oznaczonym kolorem niebieskim*”. Aby wymóc na użytkowniku konieczność użycia bloczków warunkowych (decyzyjnych), plansze zostały tak zaprojektowane, by nie dało się ich poprawnie rozwiązać bez zastosowania tych bloczków. Jeśli bloczek z warunkowym skretem nie zostanie użyty, Lumi będzie kontynuował swój kierunek jazdy, aż natknie się na przeszkodę, np. w postaci ściany.

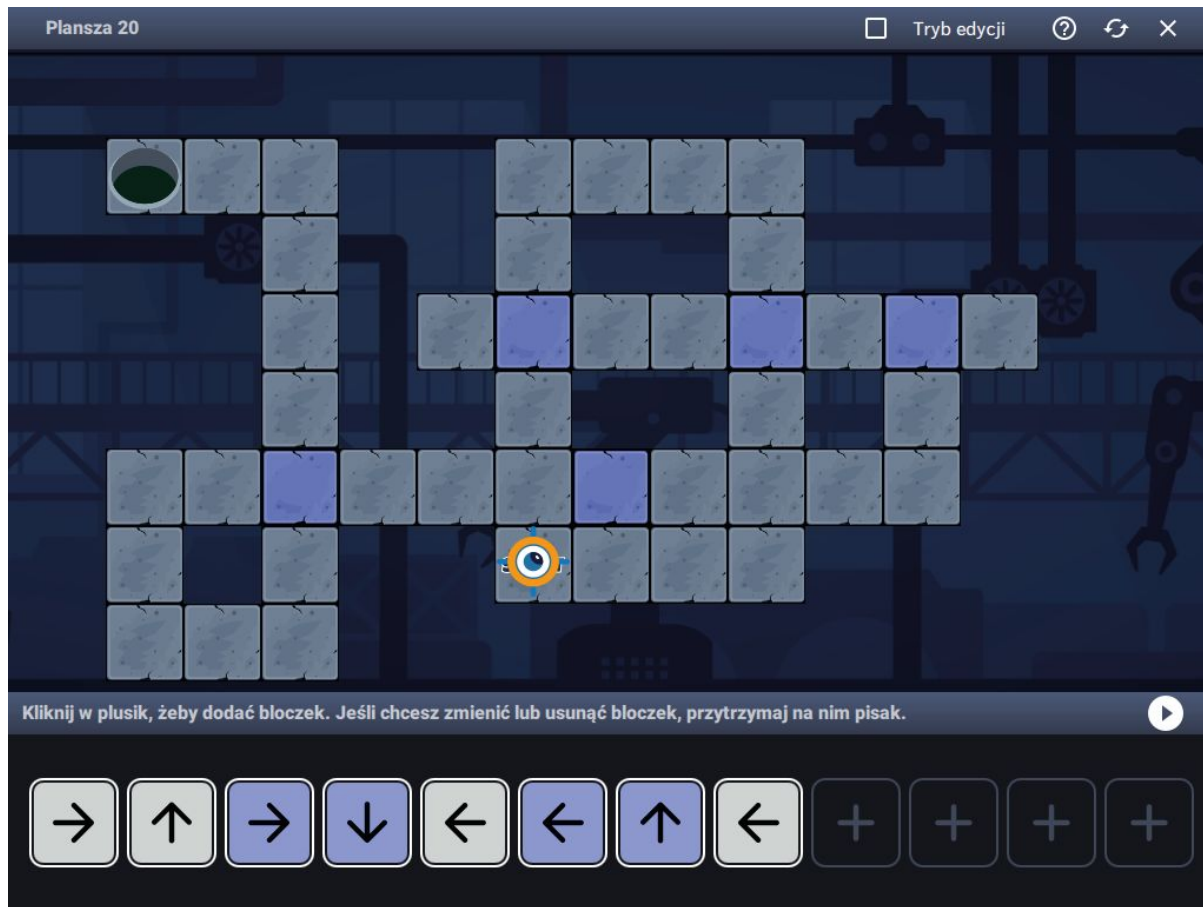
W ramach kursu *Jedź Lumi!* jest przygotowanych dziesięć zadań wprowadzających w to zagadnienie (poziomy 11-20), a łącznie jest siedemdziesiąt ćwiczeń z wykorzystaniem instrukcji warunkowych i innych technik programowania.

Przykładowe zadanie (Plansza 11.) wymagające użycia instrukcji warunkowej dla zmiany kierunku jazdy zaprezentowano poniżej.



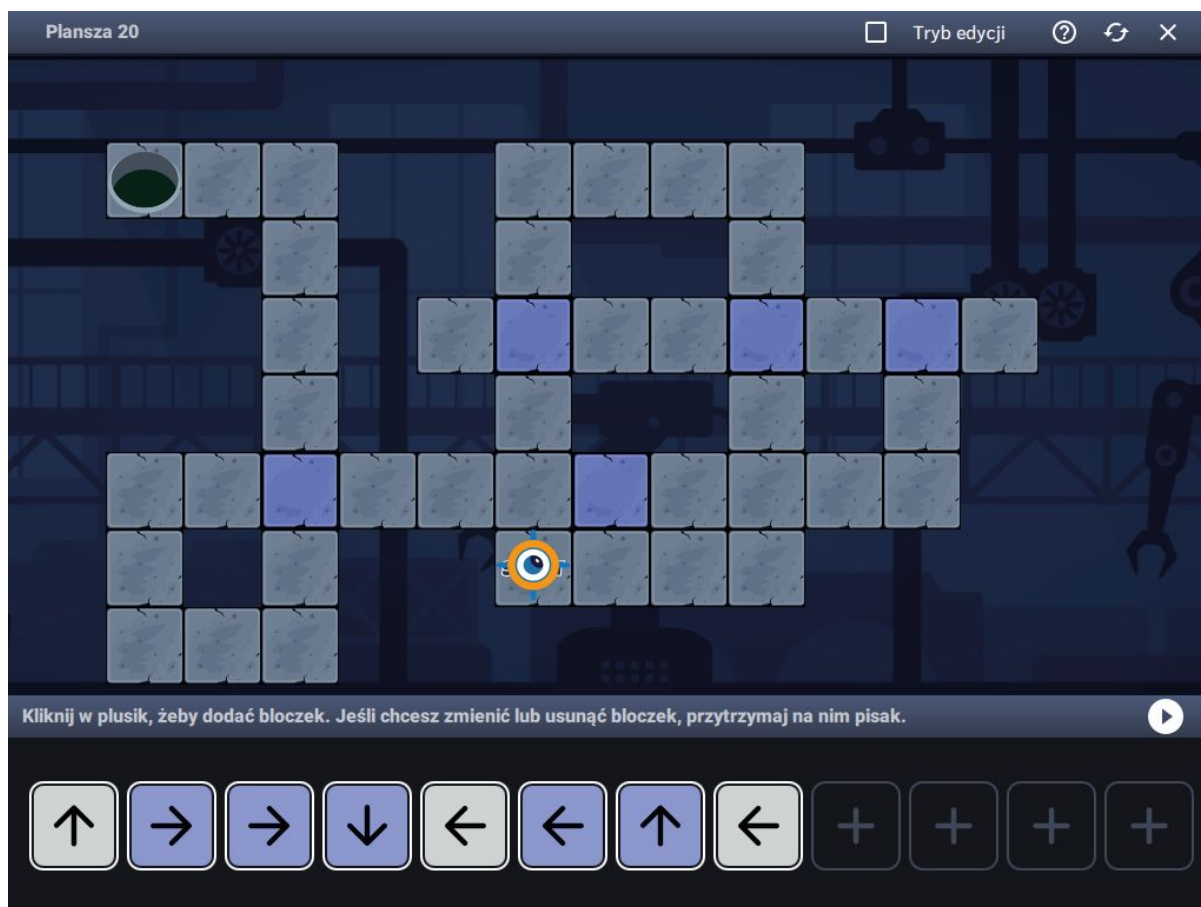
W powyższym zadaniu zmiana kierunku jazdy z *jedź w prawo* na *jedź do dołu* jest możliwa dzięki użyciu bloczka warunkowego oznaczonego kolorem niebieskim. Zadanie ma więcej niż jedno poprawne rozwiązanie.

Przykład złożonego zadania pokazano na poniższym rysunku. Zadanie to ma przynajmniej dwa poprawne rozwiązania.



Powyższe rozwiązanie wymaga użycia kilku instrukcji warunkowych. Nie ma konieczności skorzystania ze wszystkich niebieskich pól decyzyjnych w sekwencji.

Poniżej pokazano inne rozwiązanie dla tego samego zadania.



W powyższym rozwiązaniu skorzystano ze wszystkich pól decyzyjnych.

Akcje / Podnieś i upuść

Cele operacyjne, uczeń:

- określa wzajemne położenie przedmiotów na płaszczyźnie,
- stosuje w sekwencji instrukcji akcję “Podnieś”,
- stosuje w sekwencji instrukcji akcję “Upuść”.

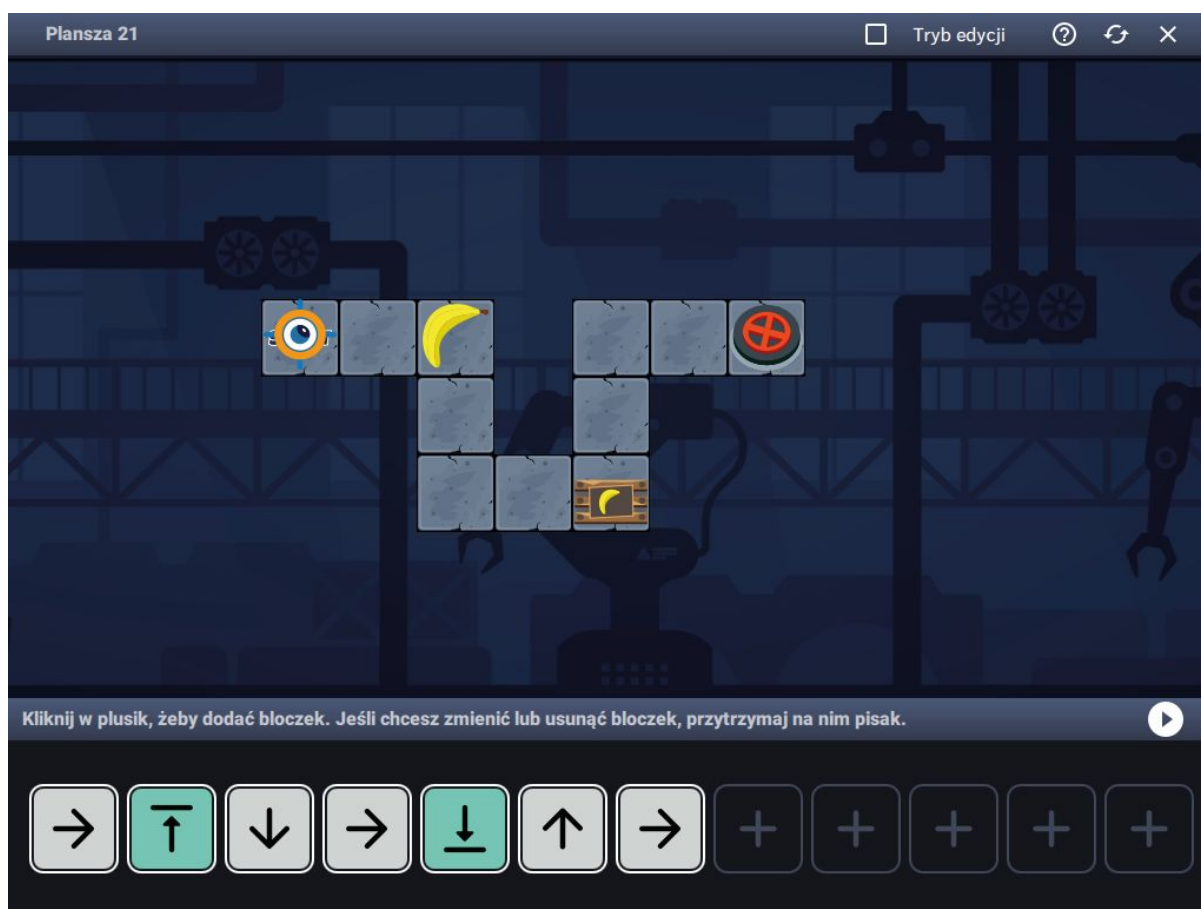
Akcje w programowaniu mają postać dodatkowych instrukcji - bloków kodu - ze z góry określoną funkcjonalnością. O akcjach można myśleć jak o pewnych ułatwieniach - wbudowanych funkcjach, czyli sekwencjach kodu, które zostały wcześniej przygotowane i zapisane w postaci pojedynczego bloku kodu. Takie podejście bardzo usprawnia programowanie, ponieważ daje programiście na start zbiór “narzędzi”, którymi może się posłużyć do rozwiązywania właściwego zadania. Akcje często są udostępniane jako składniki środowiska pracy (tzw. biblioteki funkcji), które są znane w momencie rozpoczęcia rozwiązywania danego zadania.

W kursie *Jedź Lumi!* zostało założone, że takimi akcjami, które są znane i dostępne do użycia są predefiniowane możliwości robota - podnoszenie przedmiotów i odkładania/upuszczanie ich. W przygotowanych ćwiczeniach założono dostęp do akcji *Podnieś* i *Upuść*. By wymusić użycie

(przećwiczenie) tych akcji przez ucznia, przygotowano zadania, w których wymagane jest przeniesienie wirtualnego przedmiotu do wskazanych miejsc. Przedmioty znajdują się na końcu korytarzy - ścieżek, gdzie z definicji kończy się ruch robota i czeka on na kolejną instrukcję.

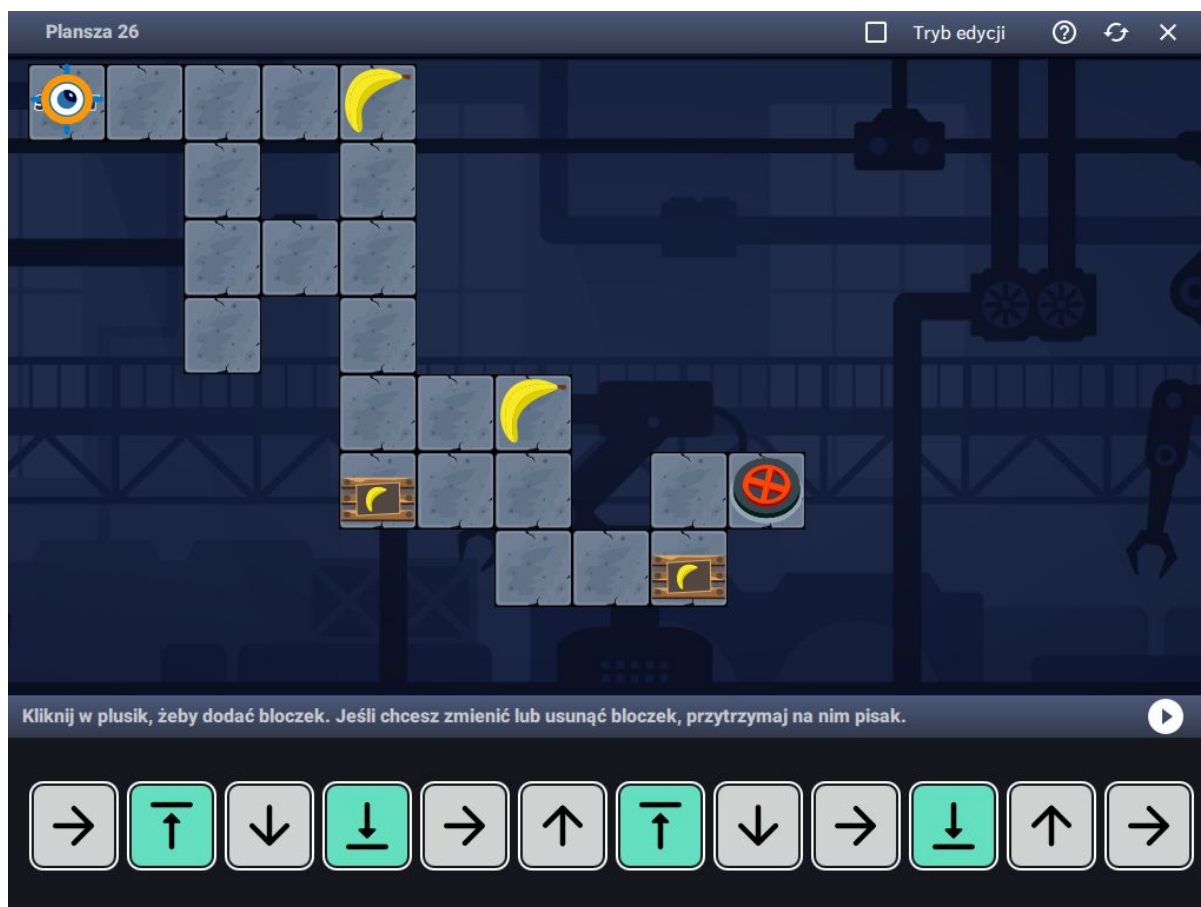
W ramach kursu *Jedź Lumi!* jest przygotowanych dziesięć zadań wprowadzających w to zagadnienie (plansze 21-30), a łącznie jest pięćdziesiąt ćwiczeń z wykorzystaniem akcji *Podnieś* i *Upuść* i innych technik programowania.

Przykładowe zadanie wymagające użycia akcji *Podnieś* i *Upuść* pokazano na poniższym rysunku (Plansza 21.).



W powyższym zadaniu robot ma do przewiezienia jeden przedmiot. Akcje dla robota zostały zdefiniowane w sekwencji dodatkowymi bloczkami: *Podnieś* i *Upuść*.

Bardziej złożony przykład zadania z akcjami *Podnieś* i *Upuść*, z koniecznością przeniesienia dwóch przedmiotów do miejsc docelowych pokazano poniżej.



W powyższym zadaniu Lumi ma do przewiezienia dwa przedmioty. Dla każdego z przedmiotów użyto zielonych bloczków z instrukcjami *Podnieś* i *Upuść*.

Pętle iteracyjne / Powtarzanie sekwencji w pętli

Cele operacyjne, uczeń:

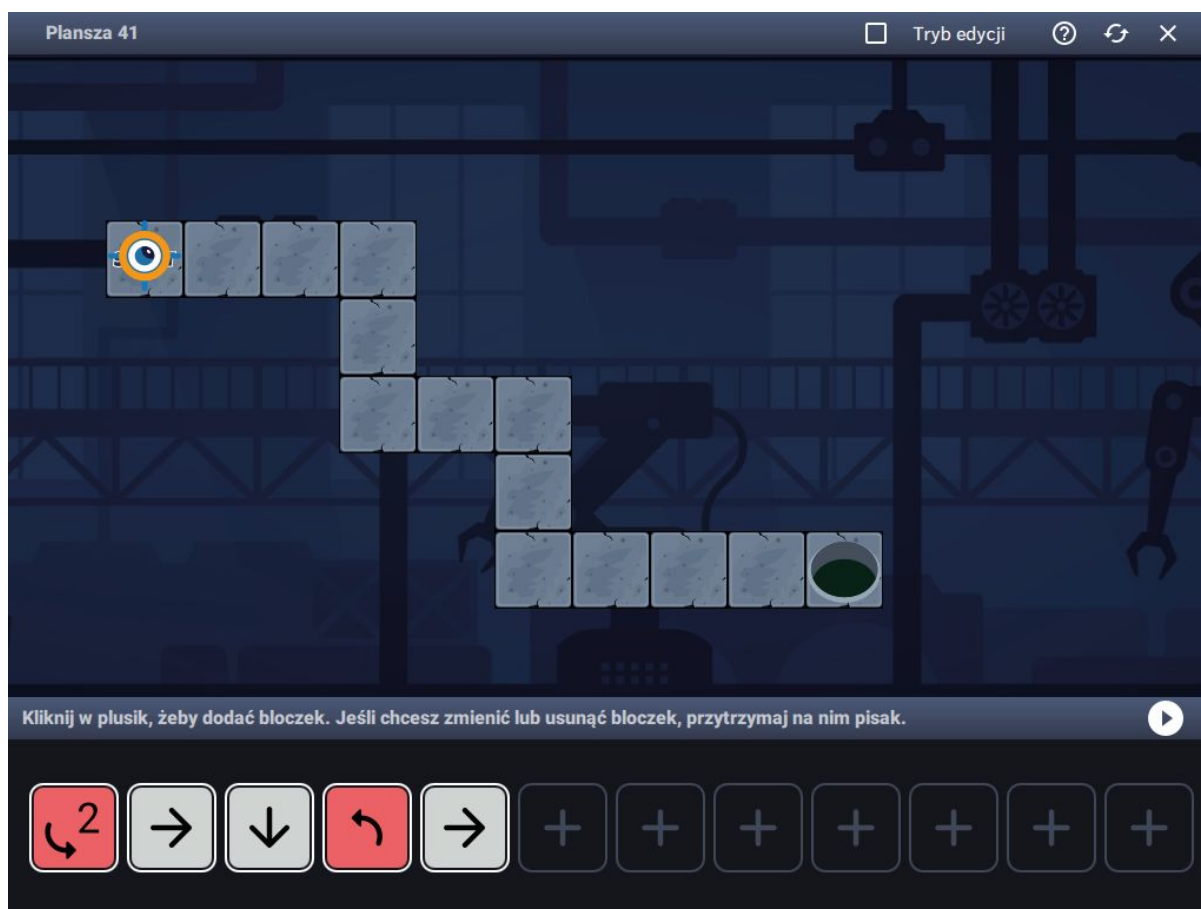
- wyodrębnia w długiej sekwencji instrukcji powtarzaną ciągle sekwencję
- układa sekwencję instrukcji, stosując pętlę iteracyjną

Pętla iteracyjna (zwana czasami pętlą licznikową) jest techniką programowania, która pozwala na wielokrotne wykonanie tych samych instrukcji, bez konieczności wypisywania ich wszystkich po kolei. Podejście to jest szczególnie przydatne tam, gdzie w ramach pełnej sekwencji kodu występują obok siebie powtarzalne fragmenty kodu. Dobrą praktyką jest poszukiwanie rytmicznych fragmentów kodu, zidentyfikowanie pojedynczego powtarzanego wzoru w tym fragmencie; wydzielenie go jako osobnego bloku kodu i określenie, ile razy ma się wykonać. Jest to możliwe dzięki instrukcji typu *Powtarzaj* (ang. *Loop*), w której można wskazać, ile razy program ma wykonać dany fragment sekwencji. Stosując instrukcję *Powtarzaj*, należy pamiętać o ustawieniu zakresu, czyli początku i końca sekwencji, która ma być wykonana w pętli. W praktyce taki zakres rozpoczyna się zaraz po bloczku *Powtarzaj*, a kończy się na specjalnym bloczku oznaczającym koniec pętli (ang. *End-Loop*).

Ze względu na specyfikę ruchu robota w aplikacji *Jedź Lumi!*, który po otrzymaniu instrukcji jazdy w określonym kierunku, kieruje się zawsze do końca korytarza, identyfikacja powtarzanych sekwencji ruchu nie zawsze jest oczywista przy pierwszym spojrzeniu na planszę. Okazuje się, że długość korytarza nie ma tutaj znaczenia dla określania sekwencji kierunków ruchu w pętli, ponieważ nie liczymy pojedynczych kroków-pól na trasie przejazdu.

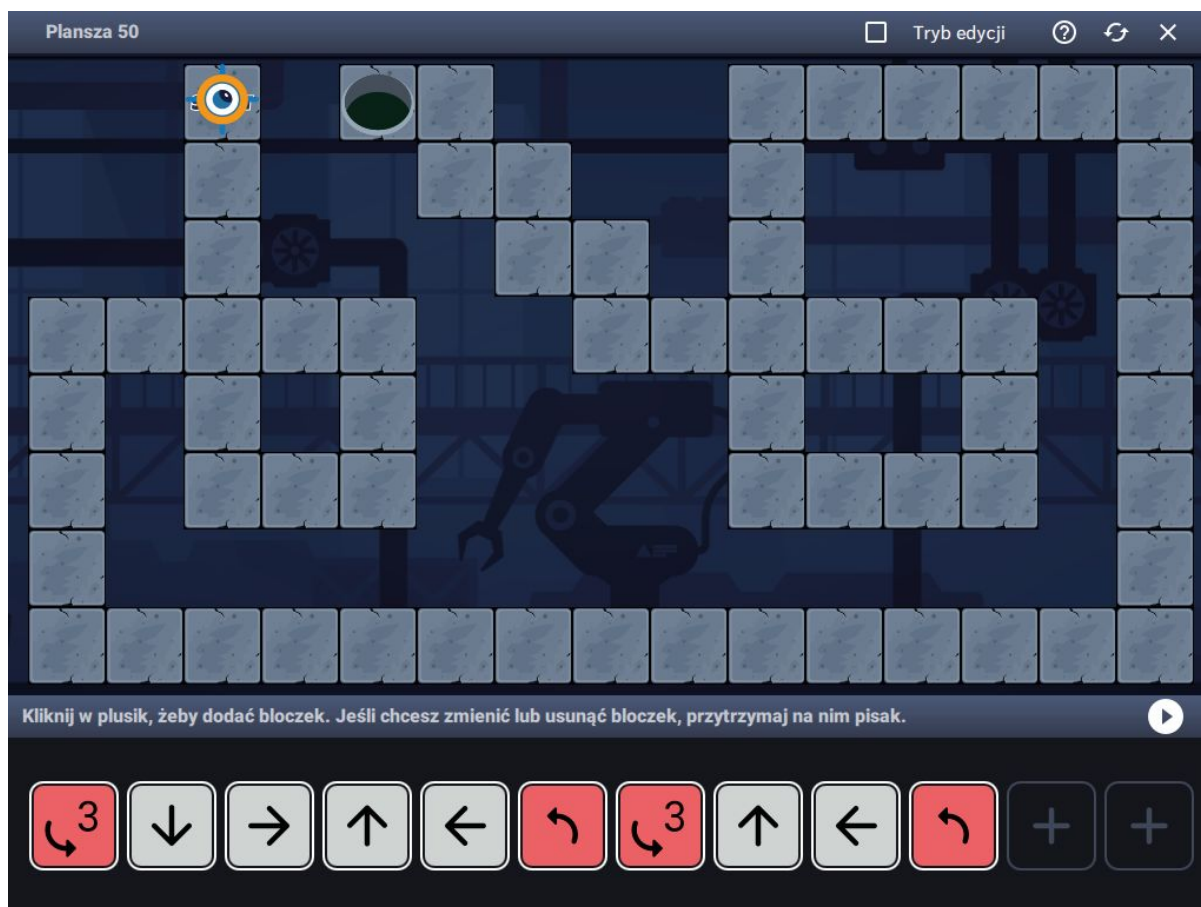
W ramach kursu *Jedź Lumi!* jest przygotowanych dziesięć zadań wprowadzających w zagadnienie pętli iteracyjnej (poziomy 41-50), a łącznie jest czterdzieści ćwiczeń z użyciem pętli iteracyjnej i innych technik programowania.

Przykład planszy wprowadzającej pojęcie pętli iteracyjnej zamieszczono poniżej.



W powyższym rozwiązaniu, zakres pętli oznaczono dwoma czerwonymi blokami. Początek pętli zaznaczono blokiem *Powtarzaj 2 razy*, a koniec zakresu pętli blokiem z instrukcją *Koniec pętli*. Ilość powtórzeń, czyli iteracji, wynosi 2 i jest to określone w bloku z instrukcją początku pętli.

Przykład planszy z bardziej złożonym zadaniem z dwiema pętlami iteracyjnymi (plansza 50.) występującymi po sobie zamieszczono poniżej.



W powyższym rozwiązaniu, zakresy pętli także oznaczono dwoma czerwonymi bloczkami. Ilość powtórzeń, czyli iteracji w każdej pętli wynosi 3.

Funkcje / Definiowanie funkcji

Cele operacyjne, uczeń:

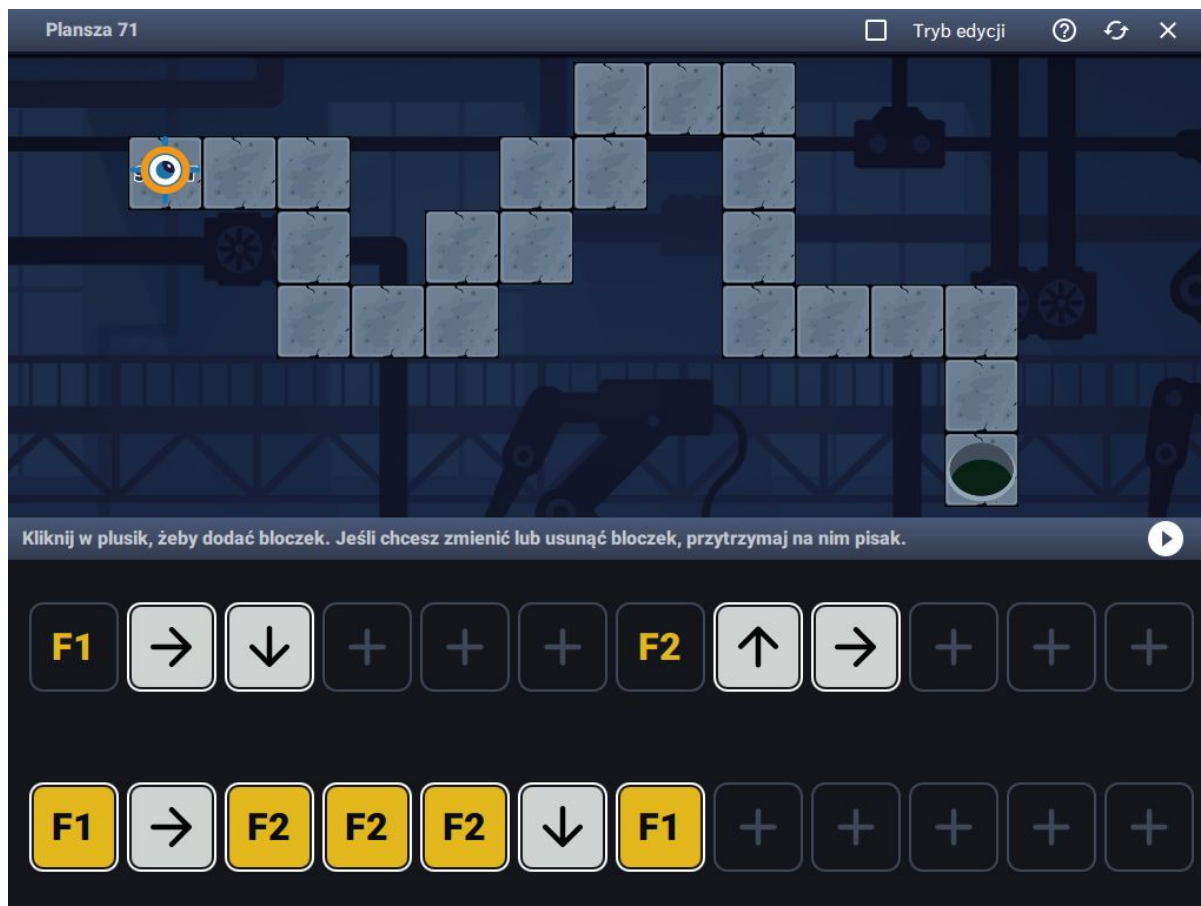
- potrafi wyodrębnić z długiej sekwencji największą powtarzalną sekwencję,
- umie zmienić fragment kodu w funkcję,
- układa sekwencję instrukcji z wywołaniem funkcji.

Definiowanie funkcji jest techniką w programowaniu, która pozwala na wydzielenie z kodu tych fragmentów sekwencji, które powtarzają się w różnych miejscach w kodzie, a następnie zapisanie ich w postaci osobnych sekwencji instrukcji (oddzielnego kodu). Dodatkowa zdefiniowana sekwencja otrzymuje swoją nazwę, dzięki czemu można korzystać ze zdefiniowanej funkcji w innych miejscach w kodzie programu, wstawiając nazwę funkcji, podobnie jak bloczki akcji - nazywamy to wywołaniem funkcji. Dodatkowo, łącząc funkcje z pętlami iteracyjnymi oraz instrukcjami warunkowymi, programista może tworzyć krótki i przejrzysty kod programu, który jest łatwy w dalszej rozbudowie. Stosowanie funkcji podejście ułatwia również wprowadzanie zmian, naprawę błędów w kodzie i współpracę z innymi programistami, ponieważ łatwiej jest analizować mniejsze fragmenty kodu zamknięte w

funkcjach, niż jedną długą sekwencję kodu rozwiązującą całe zadanie. Zmiana, którą wprowadzimy w kodzie źródłowym funkcji, jest dostępna i jednakowa dla wszystkich kodów źródłowych programów, które używają tej funkcji.

W ramach kursu *Jedź Lumi!* jest przygotowanych dziesięć zadań wprowadzających w zagadnienie definiowania funkcji (poziomy 71-80), a łącznie jest czterdzieści ćwiczeń z wykorzystaniem funkcji i innych technik programowania.

Przykład zadania:



Zasoby programu nauczania

Język programowania

Ogólna koncepcja programowania na podłodze interaktywnej

Programowanie w aplikacji *Jedź Lumi!* polega na układaniu wirtualnych bloczków z instrukcją w jeden poziomy ciąg znaków. Robimy to na podłodze interaktywnej, używając świetlnych pisaków (pisaków interaktywnych). Cechą charakterystyczną w tym języku programowania bohatera jest brak konieczności liczenia pól i każdego kroku robota na planszy. Jeśli mówimy

“Jedź w prawo”, to robot będzie poruszał się dotąd, dopóki nie napotka na drodze wirtualnej przeszkody, np. ściany labiryntu.

Składnia i semantyka języka programowania w aplikacji *Jedź Lumi!*

Składnia - zasady tworzenia sekwencji instrukcji



Główna sekwencja programu (inaczej zwana sekwencją podstawową) w aplikacji *Jedź Lumi!* może składać się maksymalnie z dwunastu blozków z instrukcjami.

W sekwencji można rozróżnić dwa rodzaje blozków-instrukcji:

- instrukcje proste, wyrażone w postaci jednego bločka z rozkazem np: *Jedź do dołu*, *Podnieś*, *Jeśli jesteś na polu warunkowym, to jedź w prawo*.
- instrukcje złożone, wyrażone w postaci jednego lub kilku blozków z rozkazami, np: *Powtórz następującą sekwencję 3 razy*, *Wykonaj pierwszą funkcję*. Dotyczą one pętli i funkcji.

Teoretycznie dla każdej planszy można ułożyć sekwencję składającą się wyłącznie z blozków z prostymi rozkazami, bez stosowanie pętli i funkcji. Z racji narzuconego ograniczenia liczby blozków w sekwencji głównej do dwunastu, w niektórych zadaniach trzeba zastosować bardziej złożone struktury kodu, takie jak pętla iteracyjna lub zdefiniować funkcję, tworząc dla niej sekwencję pomocniczą.

Sekwencję dla funkcji można zdefiniować używając maksymalnie sześciu blozków z instrukcją. Można zdefiniować dwie funkcje dla zadania. Początek definicji funkcji oznaczony jest

bloczkiem  albo . Zdefiniowanie funkcji pozwala ograniczyć zużycie miejsca w

sekwencji głównej do jednego bločka  albo , za pomocą którego odwołujemy się do zdefiniowanej funkcji.







Stosując pętlę iteracyjną należy wskazać liczbę powtórzeń **n**, oznaczyć bloczkiem 

początek sekwencji i bloczkiem  koniec sekwencji, którą ma powtarzać robot.

Podsumowanie zasad budowania kodu zaprezentowano w Tabeli: *Składnia języka w aplikacji Jedź Lumi!*.

Tabela: Składnia języka w aplikacji Jedź Lumi!






Element języka	Zasada stosowania
Długość podstawowej sekwencji instrukcji	Maksymalnie 12 blozków z instrukcją.

Długość sekwencji instrukcji dla funkcji	Maksymalnie 6 blozków z instrukcją.
Długość sekwencji instrukcji dla pętli	Maksymalnie 10 blozków w ramach sekwencji głównej.
Definiowanie funkcji	<p>Oznaczamy blozkiem  albo  początek sekwencji pomocniczej dla definiwanej funkcji.</p> <p>Odwołujemy się do funkcji w sekwencji głównej jednym blozkiem  albo .</p>
Definiowanie pętli iteracyjnej	<p>Stosujemy dodatkowe 2 blozki w sekwencji głównej. Oznaczamy blozkiem  początek instrukcji z pętlą i definiujemy liczbę powtórzeń n dla instrukcji w pętli, a blozkiem  oznaczamy koniec instrukcji w pętli.</p>

Semantyka - znaczenie poszczególnych instrukcji

Wyjaśnienia dla blozków z instrukcjami zaprezentowano w Tabeli: *Polecenia-blozki w aplikacji Jedź Lumi!*

Tabela: Polecenia-bloczki w aplikacji *Jedź Lumi!*

Symbol / Bloczek	Znaczenie - instrukcja dla robota
	Jedź w prawo
	Jedź w lewo
	Jedź do góry
	Jedź do dołu
	Jeśli jesteś na polu warunkowym, to jedź w prawo
	Jeśli jesteś na polu warunkowym, to jedź w lewo
	Jeśli jesteś na polu warunkowym, to jedź do góry
	Jeśli jesteś na polu warunkowym, to jedź do dołu
	Podnieś przedmiot (ang. Pick up)
	Upuść przedmiot (ang. Drop)










	Wykonaj sekwencję n razy. Początek pętli iteracyjnej (n-liczba powtórzeń/ iteracji, ang. number).
	Koniec sekwencji w pętli iteracyjnej
	Wykonaj funkcję F1
	Wykonaj funkcję F2

Tabela: Oznaczenia pól na planszy w aplikacji *Jedź Lumi!*

Pole na planszy	Znaczenie - Informacja dla użytkownika
	Pole Start, skąd rusza robot
	Pole Meta, do którego ma dotrzeć robot
	Przedmiot do podniesienia - pole, na którym trzeba podnieść przedmiot
	Miejsce do upuszczenia przedmiotu - pole, na którym trzeba położyć przedmiot
	Pole warunkowe - pole, na którym można zmienić kierunek jazdy robota



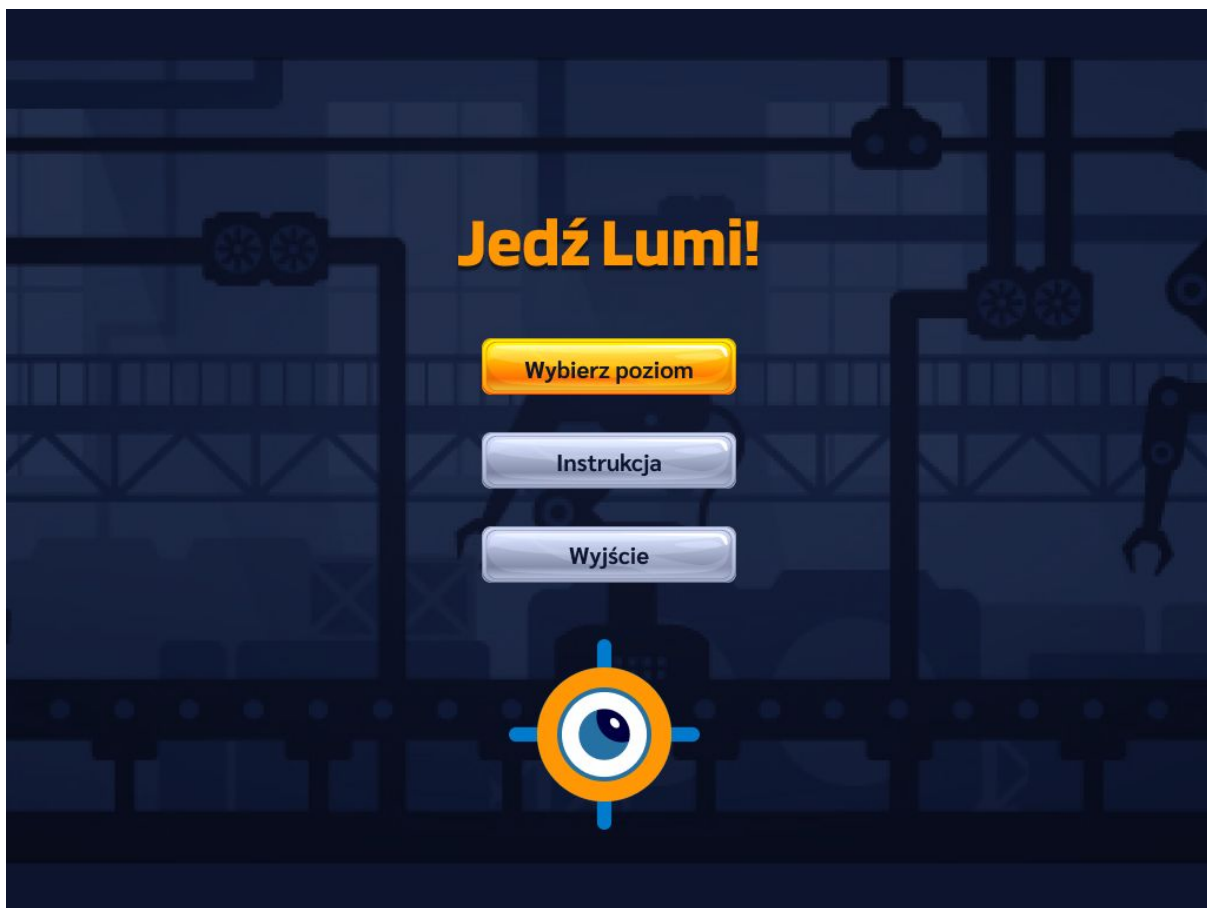
Ścieżka - droga, po której może poruszać się robot

Praca z aplikacją *Jedź Lumi!*

Po uruchomieniu aplikacji *Jedź Lumi!* widoczny jest ekran głównego menu. Z tego miejsca możesz:

- wybrać poziom i rozpocząć grę (przycisk **Wybierz poziom**),
- przejść do instrukcji aplikacji (przycisk **Instrukcja**),
- wyjść z aplikacji (przycisk **Wyjście**).

Ekran startowy aplikacji *Jedź Lumi!* pokazano na poniższym rysunku.



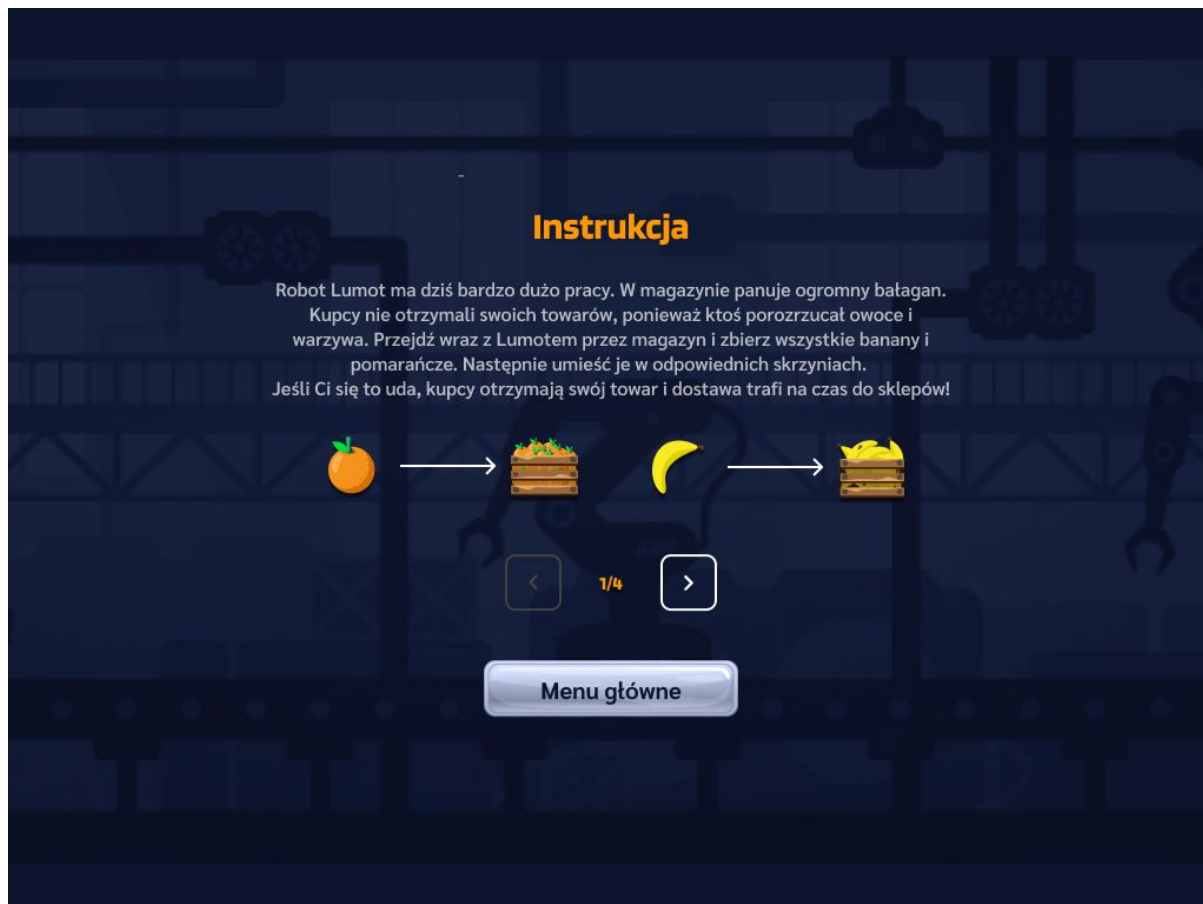
W aplikacji *Jedź Lumi!* jest do wyboru 110 poziomów gry, a każdy poziom to jedna plansza do rozwiązania. Plansze, które zostały rozwiązane, są podświetlone kolorem pomarańczowym. Po wybraniu poziomu gry, wokół kafelka pojawia się biała ramka. Możesz teraz rozpocząć grę, klikając w przycisk **Rozpocznij** w prawym dolnym rogu ekranu lub powrócić do menu głównego, wybierając **Menu główne**.

Ekran wyboru poziomu gry pokazano na poniższym rysunku.






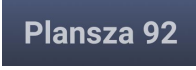
Po zapoznaniu się z instrukcją dowiesz się m.in. dlaczego w magazynie robota Lumi panuje bałagan oraz jakie misje czekają na Ciebie w trakcie gry. Znajdziesz również wskazówki dotyczące tworzenia kodu oraz stosowania podstawowych konstrukcji programistycznych, takich jak warunki, pętle iteracyjne oraz funkcje.

Widok instrukcji dla zadania pokazano na poniższym rysunku.

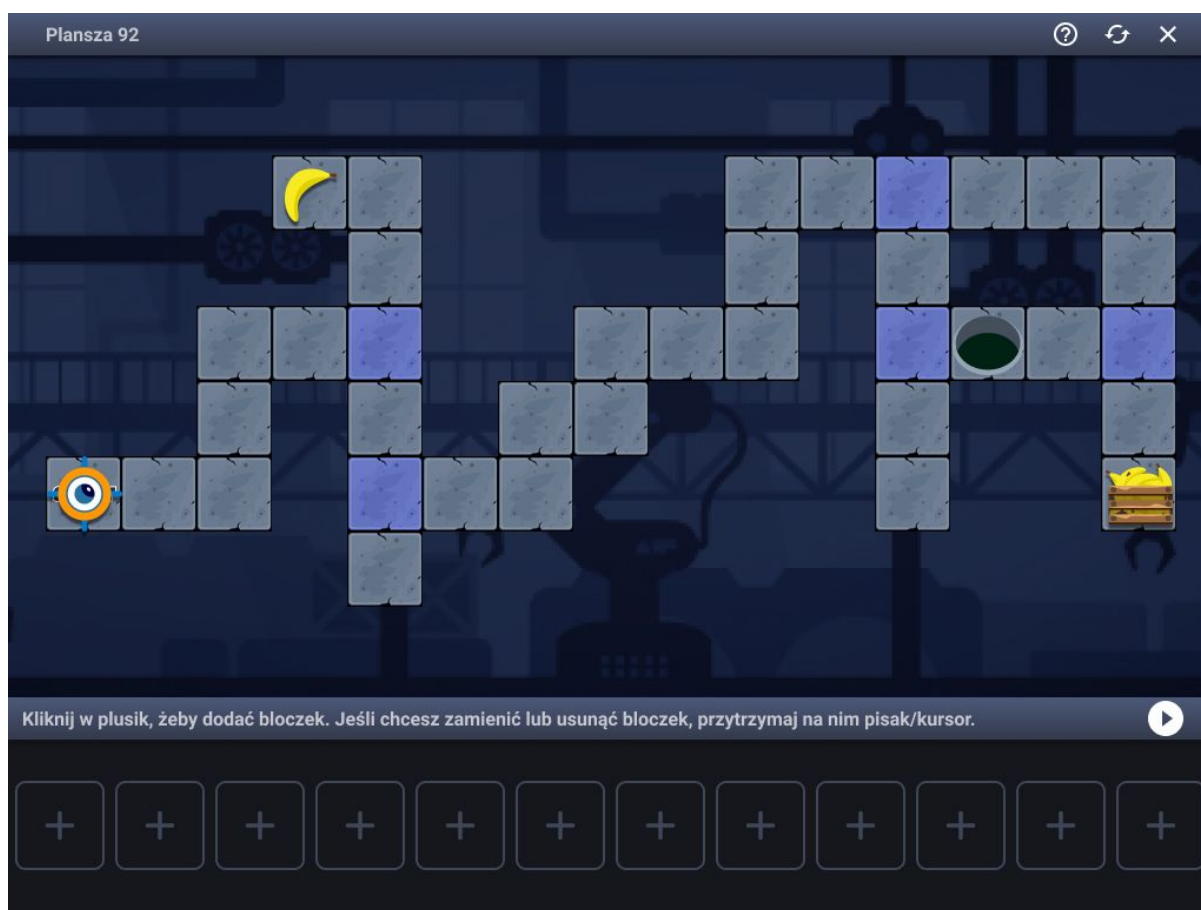


Po wybraniu poziomu i rozpoczęciu gry przechodzisz do ekranu z widokiem planszy i edycją kodu. Na górnej belce w oknie aplikacji znajdują się przyciski opisane w Tabeli: *Opcje podstawowe zarządzania aplikacją*.

Tabela: Opcje podstawowe zarządzania aplikacją.


Opcja	Akcja po naciśnięciu
	Po kliknięciu w tą ikonkę możesz zakończyć grę i przejść do głównego menu.
	Po kliknięciu w tą ikonkę możesz zrestartować grę i powrócić do początkowego widoku planszy. Uwaga: Utracisz wówczas stworzony do tej pory algorytm.
	Po kliknięciu w tą ikonkę możesz przejść do instrukcji gry, nie tracąc dotychczas ułożonej sekwencji na planszy.
	Po kliknięciu w nazwę planszy możesz przejść do ekranu z wyborem poziomu gry.


Na poniższym rysunku pokazano widok okna z przykładową planszą.



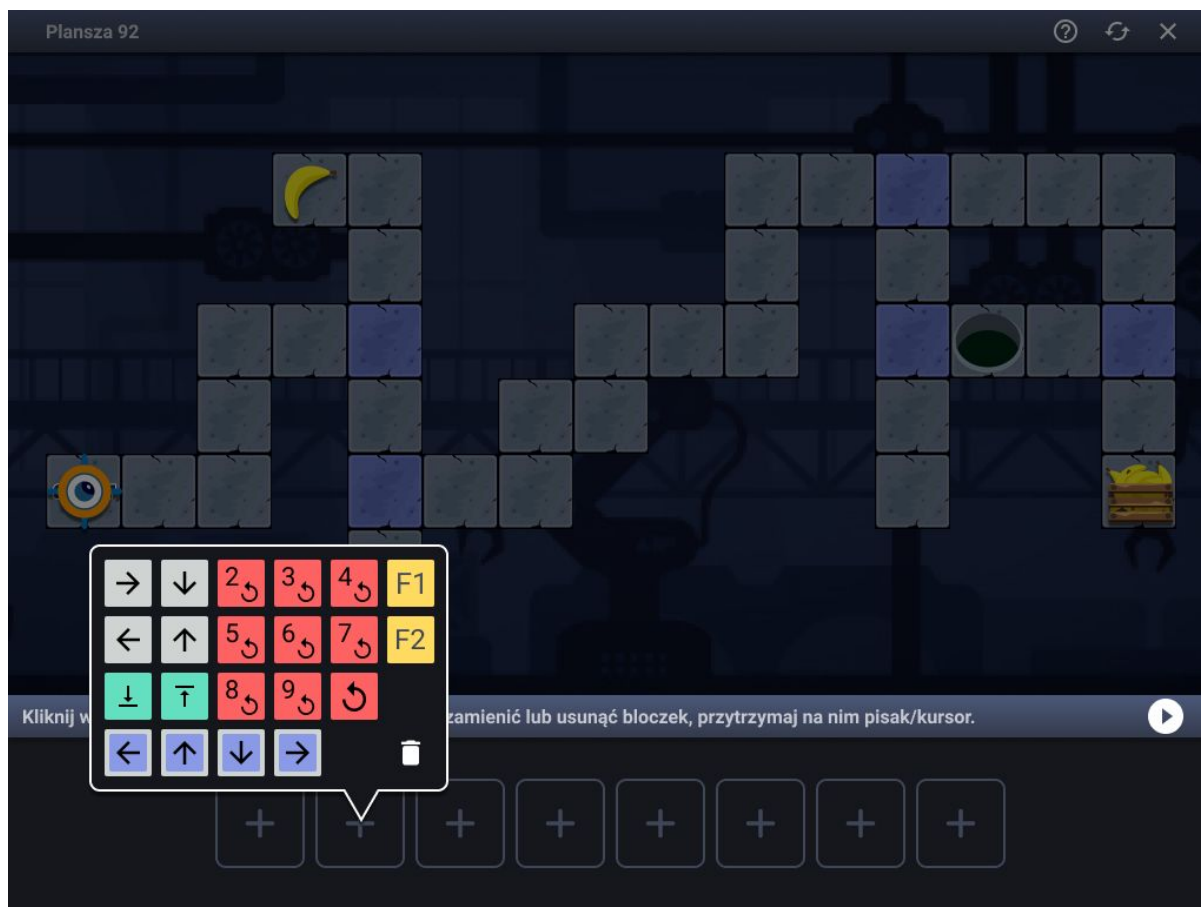
Na powyższym rysunku widać, że na belce tuż nad edycją kodu wyświetlane są komunikaty dotyczące aktualnego stanu gry.

Aby rozpocząć tworzenie kodu, kliknij w kwadrat z plusem w dolnej części ekranu i wybierz odpowiedni bloczek. Nazwy oraz przykłady użycia każdego bloczka zostały opisane w instrukcji.

W celu skasowania wybranego bloczka przytrzymaj na nim pisak, a następnie wybierz ikonę kosza. 

Po utworzeniu algorytmu z rozwiązaniem dla planszy, sprawdź swoje rozwiązanie, klikając w przycisk Uruchom , aby uruchomić program i bohatera na planszy. Jeśli stworzony przez Ciebie kod będzie poprawny, przejdziesz do następnego poziomu gry.

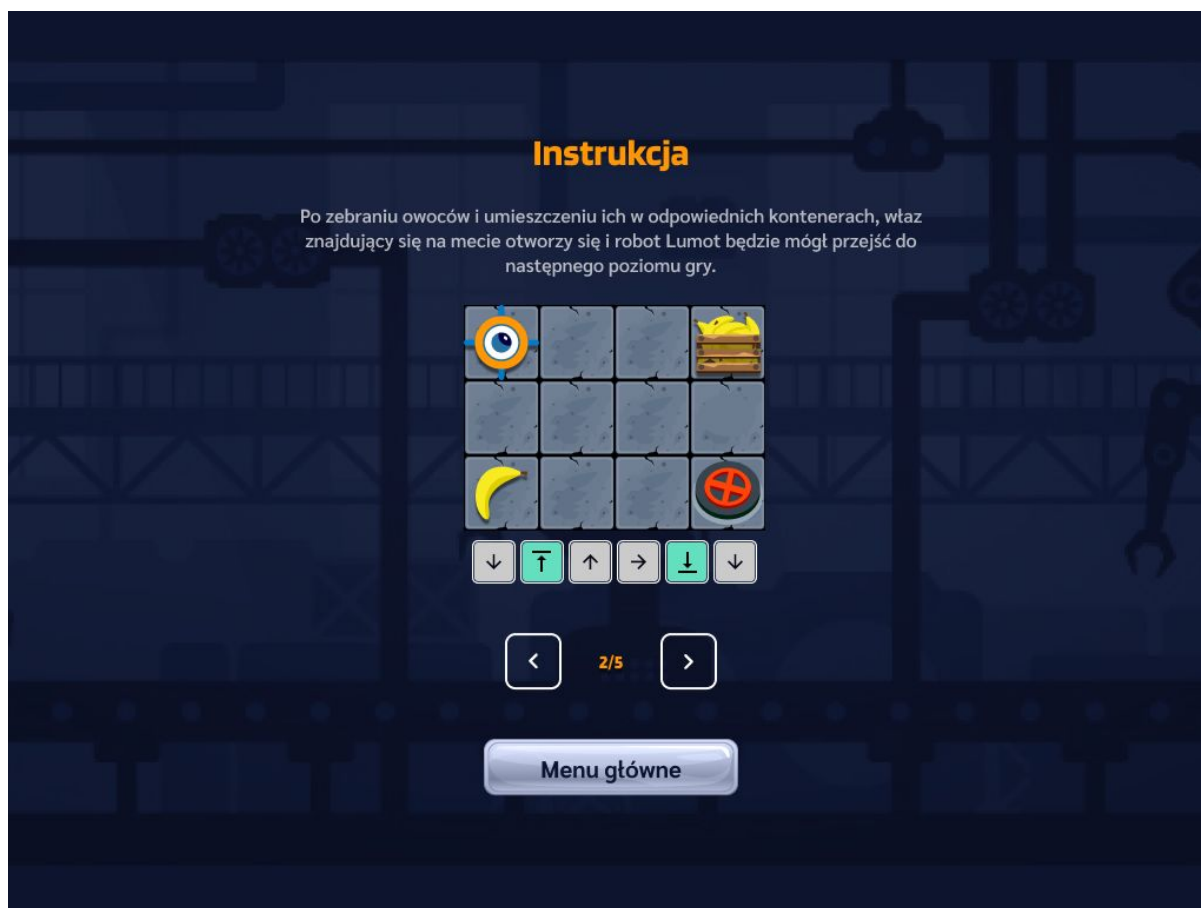
Na poniższym rysunku pokazano menu dostępne po kliknięciu pisakiem na pusty bloczek z plusikiem.



Na powyższym rysunku widać, że w podręcznym menu mamy do wyboru wszystkie podstawowe instrukcje do budowy algorytmu dla przejazdu robota.

W instrukcji wewnątrz aplikacji *Jedź Lumi!* zostały przedstawione i opisane przykłady tworzenia algorytmów z użyciem instrukcji warunkowej, akcji *Podnieś* i *Upuść*, zastosowaniem pętli iteracyjnej i funkcji występujących w grze.

Na poniższym rysunku pokazano widok okna z instrukcją.



Powyższy rysunek pokazuje okno z instrukcją gry. Aby wrócić do wyboru poziomu gry, trzeba wybrać **Menu główne**.

Struktura kursu

Kurs składa się z jedenastu lekcji, patrz Tabela *Lekcje - zakres programu nauczania*. Do każdej lekcji przyporządkowany jest zestaw dziesięciu ćwiczeń o zróżnicowanym poziomie trudności. Ćwiczenia interaktywne na podłodze są realizowane w oparciu o aplikację *Jedź Lumi!*. Lekcje mają zaproponowany scenariusz zajęć. Każda plansza ma jedno lub więcej poprawnych rozwiązań. Nauczyciel może dokonać wyboru spośród zaproponowanych ćwiczeń lub opracować własne ćwiczenie w aplikacji *Jedź Lumi!* za pomocą edytora plansz. Uczniowie realizują gotowy scenariusz lub zmodyfikowany przez nauczyciela.

Tabela: Objętość kursu

Element kursu	Ilość
Liczba lekcji w kursie	12
Liczba ćwiczeń - plansz dla 1 lekcji	10
Ilość poziomów trudności plansz dla 1 lekcji	10 (1 plansza = 1 poziom)
Łączna liczba ćwiczeń - plansz w kursie	110
Tryb projektowania plansz	1

W Tabeli: *Lekcje - zakres programu nauczania* podano dziedziny, które obejmuje dana lekcja. Wskazano również grupę wiekową, dla której dedykowany jest zakres kursu.

Tabela: Lekcja - zakres programu nauczania

Numer lekcji	Zakres programu nauczania / Dziedziny (zapis skrótowy)	Ilość ćwiczeń - plansz	Numery plansz	Grupa wiekowa
1	Sekwencje (S)	10	1-10	7-14 lat
2	Sekwencje, Warunki (S+W)	10	11-20	7-14 lat
3	Sekwencje, Akcje (S+A)	10	21-30	7-14 lat
4	Sekwencje, Warunki, Akcje (S+W+A)	10	31-40	7-14 lat
5	Sekwencje, Pętle iteracyjne (S+P)	10	41-50	7-14 lat
6	Sekwencje, Pętle iteracyjne, Warunki (S+P+W)	10	51-60	7-14 lat
7	Sekwencje, Pętle iteracyjne, Warunki, Akcje (S+P+W+A)	10	61-70	7-14 lat
8	Sekwencje, Funkcje (S+F)	10	71-80	7-14 lat
9	Sekwencje, Funkcje, Warunki (S+F+W)	10	81-90	7-14 lat
10	Sekwencje, Funkcje, Warunki, Akcje (S+F+W+A)	10	91-100	7-14 lat
11	Sekwencje, Funkcje, Pętle iteracyjne, Warunki, Akcje (S+F+P+W+A)	10	101-110	7-14 lat

W Tabeli *Zakres programu nauczania - lekcja* podano numery lekcji, które realizują dany zakres materiału.

Tabela: Zakres programu nauczania - Lekcje - Poziomy zadań

Zakres programu nauczania / Dziedziny	Numer lekcji	Poziomy zadań w aplikacji <i>Jedź Lumi!</i>
Sekwencje	1-11	1-110
Warunki	2, 4, 6, 7, 9, 11	11-20, 31-40, 51-70, 81-110
Akcje	3, 4, 7, 10, 11	21-40, 61-70, 91-110
Pętle	5, 6, 7, 11	41-70, 101-110
Funkcje	8, 9, 10, 11	71-110

W Tabeli: *Zakres programu - Grupa wiekowa* wskazano sugerowany wiek uczniów dla zakresu nauczania.

Tabela: Zakres programu - Grupa wiekowa

Zakres nauczania	Grupa wiekowa
Sekwencje	7-14 lat
Warunki	7-14 lat
Akcje	7-14 lat
Pętle	8-14 lat
Funkcje	10-14 lat

Scenariusze zajęć

Do kursu dołączone są przykładowe scenariusze zajęć dla lekcji wprowadzających nowe zagadnienia.

Lekcja 1: Tworzenie sekwencji instrukcji

Cele operacyjne

Uczeń:

- określa kierunki i ustala położenie robota w stosunku do własnej osoby, a także w stosunku do innych przedmiotów, rozróżnia stronę lewą i prawą, górę, dół,
- układa sekwencję instrukcji dla robota za pomocą bloczków ze strzałkami kierunkowymi,

- odczytuje sekwencję instrukcji zapisaną za pomocą bloczków ze strzałkami kierunkowymi,
- omija przeszkody w postaci ścian labiryntu.

Opis skrótowy

Stosując proste ćwiczenia w aplikacji *Jedź Lumi!* uczniowie będą tworzyć algorytmy sekwencyjne, aby poruszać robotem na interaktywnej planszy i pokierować nim w labiryncie od pola Start do pola Meta. Aby to zrobić, uczniowie ustawią odpowiednie bloczki z instrukcjami na interaktywnej planszy w jeden ciąg znaków - tworząc w ten sposób algorytm - sekwencję instrukcji dla robota.

Cele ćwiczeń

Podczas tej lekcji uczniowie rozwiną umiejętności kodowania i debugowania. Oparta na bloczkach forma instrukcji pomaga uczniom poznać sekwencję i ideę budowania kodu programu bez martwienia się o doskonalenie składni.

Plan zajęć

Czas trwania: 45 min

Rozgrzewka

Czas trwania: 20 min

Dyskusja: Zasady zachowywania się w pracowni interaktywnej

1. Poznaj oczekiwania uczniów i porozmawiaj z nimi na temat odpowiedniego zachowania po wejściu do pracowni z podłogą interaktywną, robotem i pisakami świetlnymi.
2. Zastanówcie się, jakie środki ostrożności i higieny warto stosować, aby ćwiczenia przebiegały w bezpiecznych warunkach.
3. Ustalcie zasady współpracy, aby ćwiczenia przebiegały w przyjaznej atmosferze.

Zabawa: Mucha wyszła

W tej zabawie potrzebna jest kwadratowa siatka pól 3x3 lub większa. Wystarczy kartka papieru w rozmiarze A4. Uczniowie mogą narysować siatkę samodzielnie. Upewnij się, że wszyscy mają kwadratową siatkę z jednakową ilością pól. Wyjaśnij dzieciom zasady gry:

1. Muchę umieszczamy na jednym z pól (można użyć małego przedmiotu symbolizującego owada-muchę, lub przechowywać jej położenie w wyobraźni).
2. Następnie nauczyciel posługując się wyrażeniami: *w prawo*, *w lewo*, *do góry*, *do dołu*, kieruje muchą po polach planszy, bez fizycznego przesuwania jej.

3. Uczniowie śledzą drogę muchy w swoich myślach lub przesuwając palcem po polach. Gdy mucha wyjdzie poza obszar siatki, uczniowie wołają *Wyszła!*

Zabawę można powtórzyć. Postępując się strzałkami kierunkowymi, narysuj z uczniami dwie przykładowe drogi muchy na planszy. Zaznaczcie w waszych instrukcjach, w którym momencie mucha wyszła poza planszę.

Nowe pojęcia

Sekwencja - Lista kroków.

Algorytm - Lista kroków (sekwencja) potrzebna do wykonania zadania.

Program - Algorytm, który został zakodowany w czymś, co może być uruchomione przez maszynę.

Debugowanie - Poszukiwanie i usuwanie błędów w algorytmie.

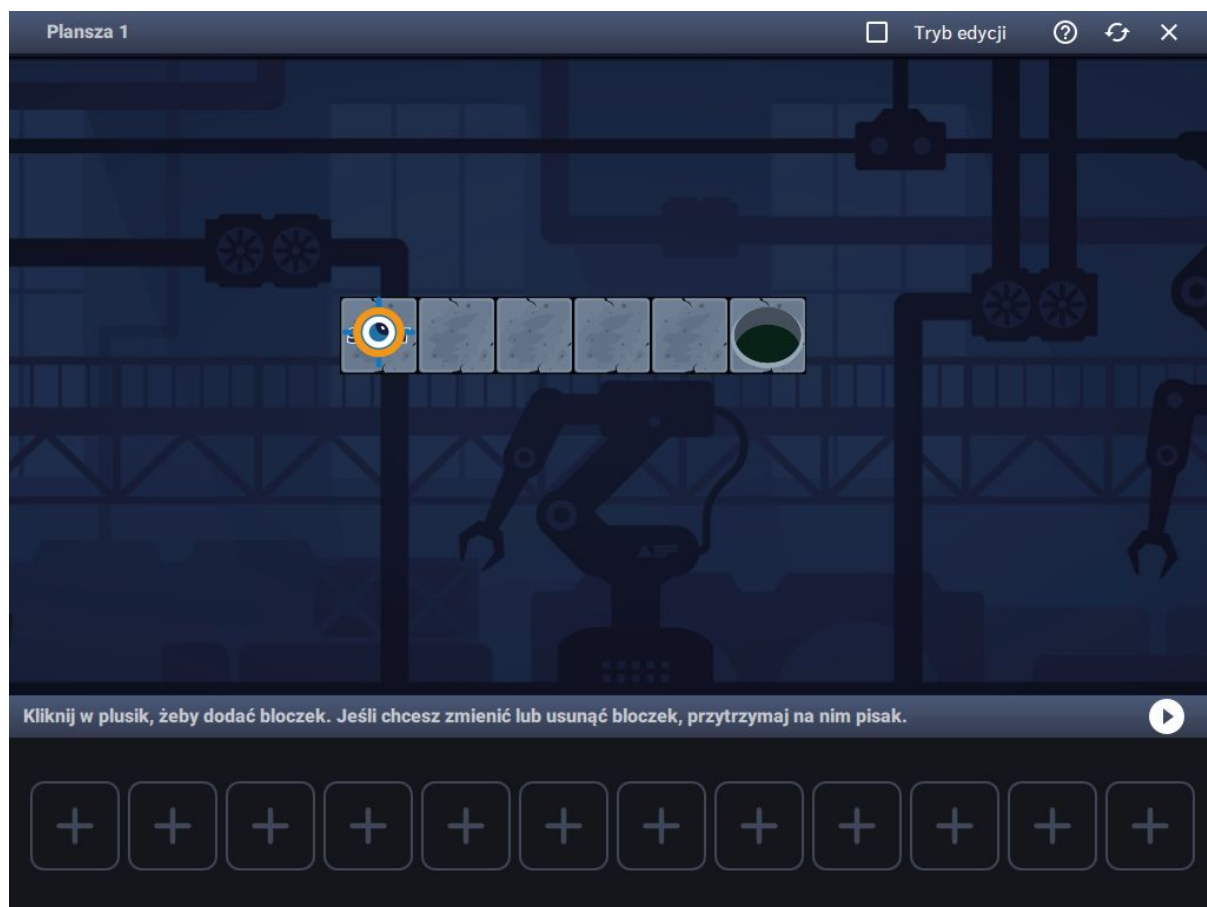
Ćwiczenia główne

Czas trwania: 20 min

Jedź Lumi! Poziom 1-10

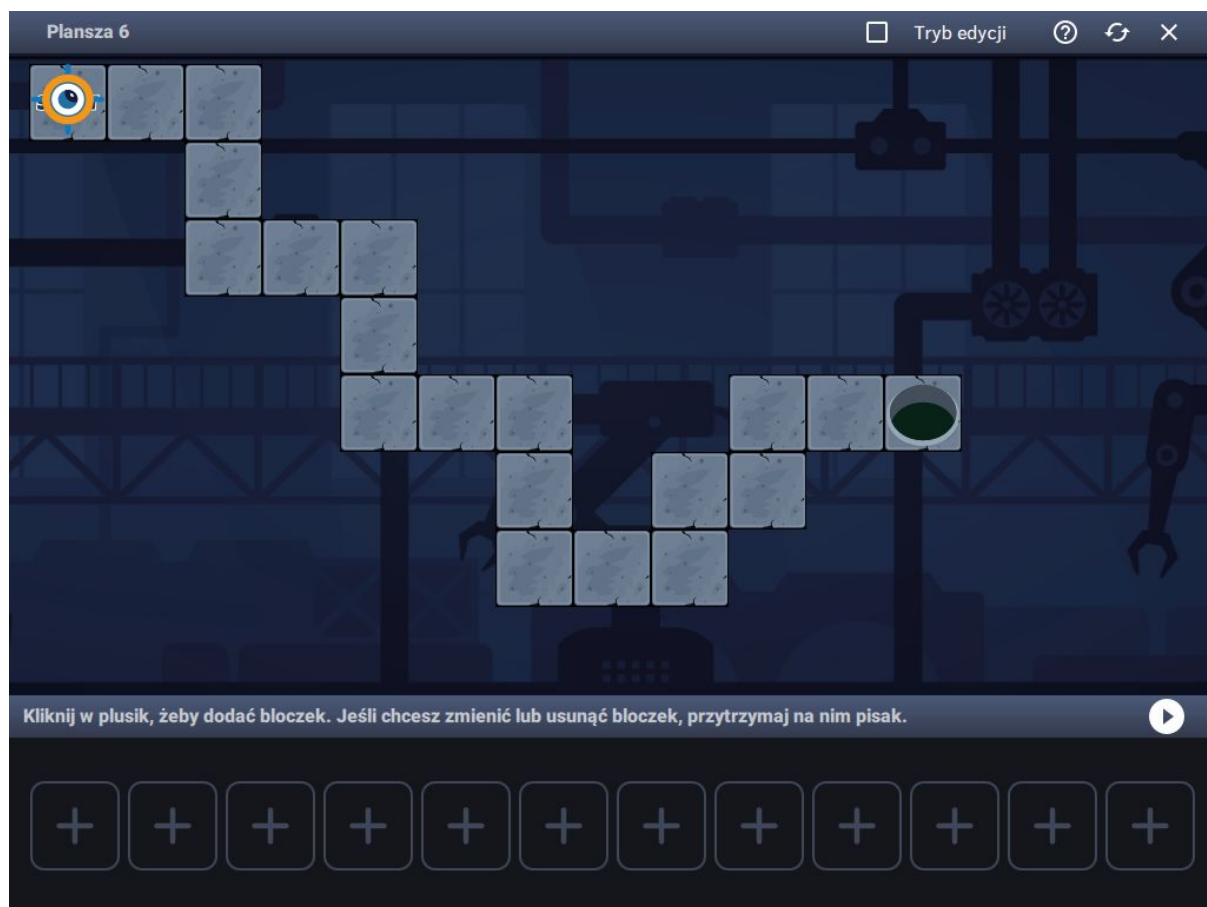
Ćwiczenia w aplikacji *Jedź Lumi!* dedykowane do lekcji nr 1: Poziom 1 - Poziom 10.

1. Prawdopodobnie część uczniów miała okazję zetknąć się z podłogą interaktywną, mimo to zrób krótkie wprowadzenie dla tych, którzy jeszcze jej nie znają.
2. Wybierz najprostszą planszę odpowiednią do tematu lekcji - Poziom1- i przy pomocy uczniów zademonstruj zasadę pracy w aplikacji. Wspólnie rozwiążcie pierwszą planszę.



3. Zapytaj uczniów, jak chcieliby nazwać swojego prawdziwego robota, jeśli nie ma on jeszcze imienia. Zachęć uczniów, by przy uruchomieniu kodu zawołali do robota: *Jedź + imię robota!* Możecie też wołać: *Jedź Lumi!*
4. Wybierz dogodny sposób pracy z grupą na interaktywnej podłodze:
 - Podziel uczniów na zespoły lub pary, które będą rozwiązywać poszczególne plansze.
 - Wszystkie zespoły mogą pracować równolegle, projektując sekwencję instrukcji dla robota na kartce papieru lub za pomocą samodzielnie przygotowanych tekturowych bloczków. Wybrany zespół przepisuje lub układa kod na planszy.
 - Sprawdźcie wspólnie wszystkie lub wybrane rozwiązania zespołów na podłodze interaktywnej.
 - Wybierzcie osobę, która będzie naśladować robota i wykonywać polecenia według zaprojektowanych sekwencji, poruszając się w labiryncie na planszy. Następnie uruchomcie kod i porównajcie rezultaty. Jeśli się różnią, to poszukajcie, gdzie się wkradł błąd.
 - Śledźcie przebieg ruchu robota w labiryncie i porównajcie go z Waszym kodem na planszy lub na kartce. W przypadku błędnego zachowania robota w labiryncie lub innego przebiegu jazdy, niż w sekwencjach instrukcji uczniów poszukajcie przyczyny w kodzie-sekwencji.

5. Dobieraj kolejne poziomy plansz do umiejętności uczniów.



Podpowiedź: Nie wszystkie plansze z danej lekcji muszą być zrealizowane podczas zajęć. Rozwiązanie dwóch plansz na lekcji również może być wystarczające. Dostosuj tempo pracy do potrzeb uczniów.

Podsumowanie

Czas trwania: 5 min

Poproś uczniów o narysowanie prostego rysunku twarzy, wyrażającej ich odczucia, wybraną emocję podczas lekcji.

Nauczanie rozszerzone

Jeśli jest taka potrzeba, zdefiniuj zadanie domowe dla chętnych, np. rysunek robota lub dodatkowe ćwiczenie utrwalające.

Przygotowanie do zajęć

- Upewnij się, że uczniowie posiadają ołówek lub długopis, zeszyt lub notatnik, w którym mogą zapisywać notatki, przemyślenia, rysować.

- Jeśli planujesz użyć podczas zajęć dodatkowych kart pracy lub innych materiałów, upewnij się, że masz wystarczającą ich ilość dla uczniów.
- Jeśli planujesz przeprowadzić dyskusję, przygotuj dodatkowe pytania.
- Przejrzyj ćwiczenia przed lekcją, aby zidentyfikować potencjalne problematyczne obszary dla Twoich uczniów.

Lekcja 2: Dodajemy warunek na trasie

Cele operacyjne

Uczeń:

- określa kierunki i ustala położenie robota w stosunku do własnej osoby, a także w stosunku do innych przedmiotów, rozróżnia stronę lewą i prawą, góra, dół,
- układa sekwencję instrukcji dla robota za pomocą bloczków ze strzałkami kierunkowymi,
- odczytuje sekwencję instrukcji zapisaną za pomocą bloczków ze strzałkami kierunkowymi,
- omija przeszkody w postaci ścian labiryntu,
- stosuje w sekwencji instrukcji warunek *“jeśli jesteś na polu oznaczonym kolorem, to zmień kierunek jazdy”* (nowy cel).

Opis skrótowy

Stosując proste ćwiczenia w aplikacji *Jedź Lumi!* uczniowie będą tworzyć algorytmy sekwencyjne z warunkiem, aby poruszać robotem na interaktywnej planszy i pokierować nim w labiryncie od pola Start do pola Meta, stosując warunek na trasie przejazdu. Aby to zrobić, uczniowie ustawią odpowiednie bloczki z instrukcjami na interaktywnej planszy w jeden ciąg znaków - tworząc w ten sposób algorytm - sekwencję instrukcji dla robota.

Cele ćwiczeń

Podczas tej lekcji uczniowie rozwiną umiejętności programowania z prostymi warunkami i debugowania. Oparta na bloczkach forma instrukcji pomaga uczniom poznać sekwencję i ideę budowania kodu programu bez martwienia się o doskonalenie składni.

Plan zajęć

Czas trwania: 45 min

Rozgrzewka

Czas trwania: 15 min

Zabawa: Jestem robotem

Ćwiczenie w grupie:

Wybrana osoba jest robotem, który wykonuje proste polecenia pozostałych osób. Zabawa może nawiązywać do poleceń w aplikacji *Jedź Lumi!*: *idź w prawo*, *idź w lewo*, *idź do góry*, *idź do dołu*. Ustalcie punkt odniesienia dla określania kierunków. Wprowadźcie do zabawy nowe warunki, np: *Jeśli spotkasz koleżankę, przywitaj się z nią*. *Jeśli spotkasz kolegę, skręć w lewo* itd.

Dyskusja

Zawiązanie dyskusji na temat rozumienia, interpretacji poleceń, w tym rozumienia zdarzeń warunkowych - sytuacji, w których coś może się zdarzyć, pod jakimś warunkiem.

Nowe pojęcia

Warunek - powtórzcie razem Wa-ru-nek

Warunek to instrukcja, która wykonuje się tylko w określonych warunkach.

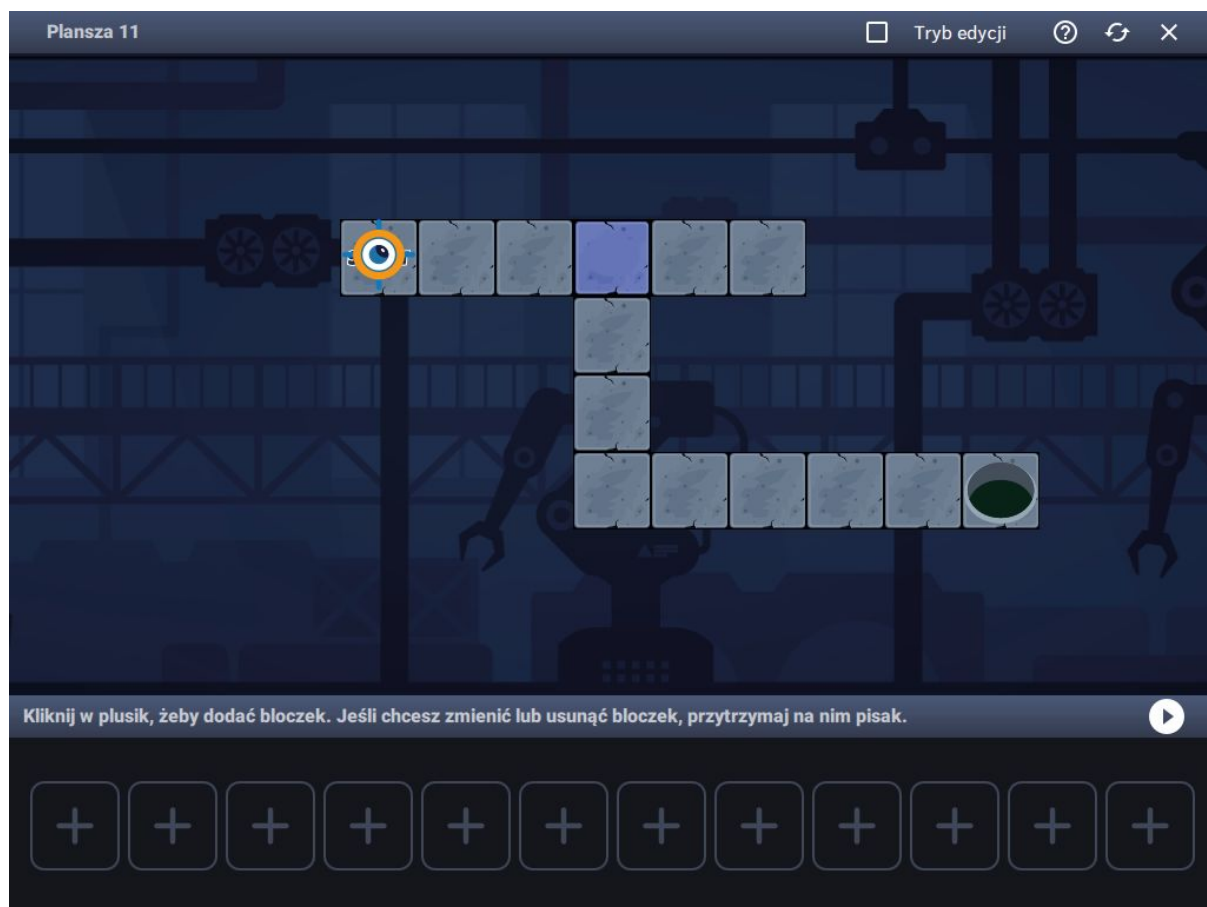
Ćwiczenia główne

Czas trwania: 25 min

Jedź Lumi! Poziom 11-20

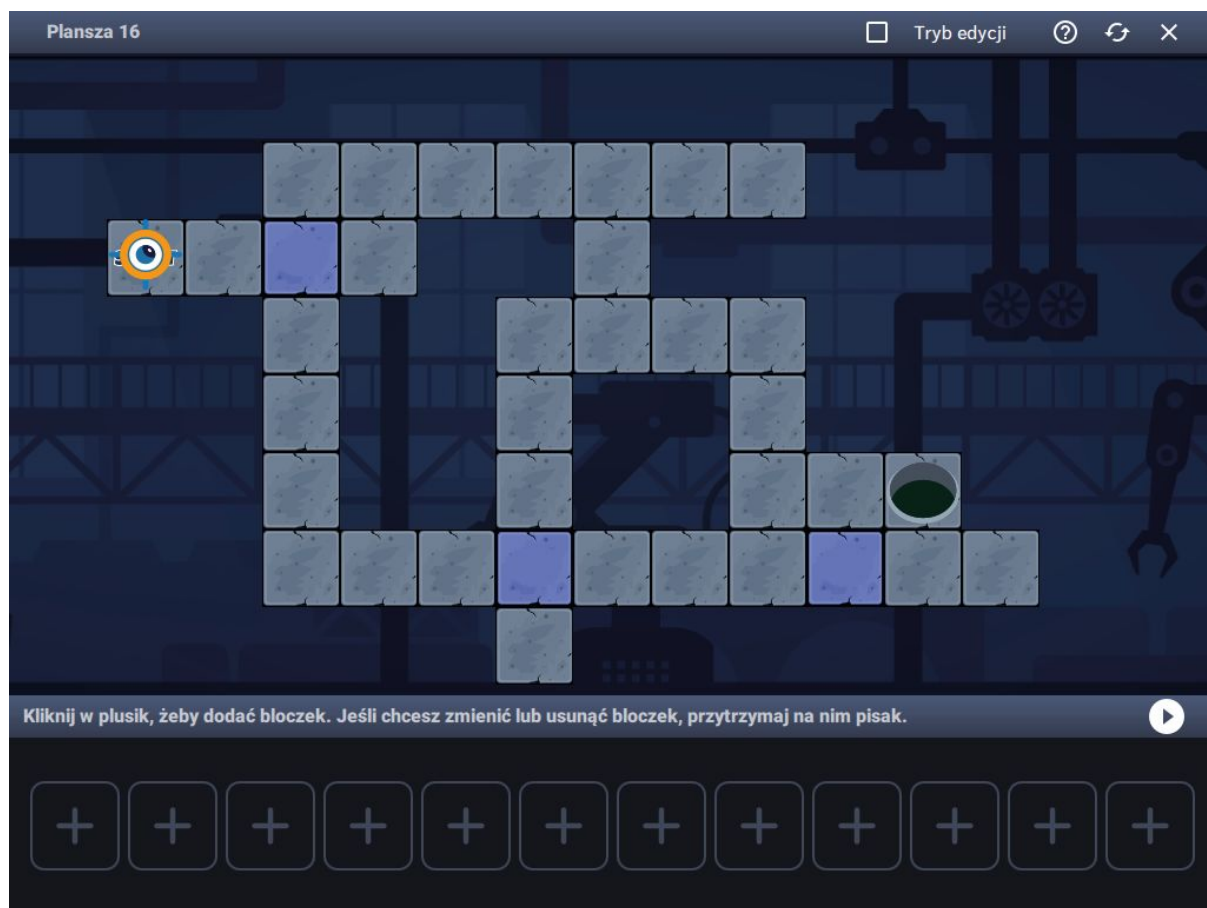
Ćwiczenia w aplikacji *Jedź Lumi!* dedykowane do lekcji nr 2: Poziom 11 - Poziom 20.

1. Wybierz najprostszą planszę odpowiednią do tematu lekcji - Poziom 11 - i przy pomocy uczniów zademonstruj zasadę pracy w aplikacji. Wspólnie rozwiążcie pierwszą planszę.



2. Uruchamiając kod programu, zawołajcie *Jedź + imię waszego robota!*
3. Wybierz dogodny sposób pracy z grupą na interaktywnej podłodze:
 - Podziel uczniów na zespoły lub pary, które będą rozwiązywać poszczególne plansze.
 - Wszystkie zespoły mogą pracować równolegle, projektując sekwencję instrukcji dla robota na kartce papieru lub za pomocą samodzielnie przygotowanych tekturowych bloczków. Wybrany zespół przepisuje lub układa kod na planszy.
 - Sprawdźcie wspólnie wszystkie lub wybrane rozwiązania zespołów na podłodze interaktywnej.
 - Wybierzcie osobę, która będzie naśladować robota i wykonywać polecenia według zaprojektowanych sekwencji, poruszając się w labiryncie na planszy. Następnie uruchomcie kod i porównajcie rezultaty. Jeśli się różnią, to poszukajcie, gdzie się wkradł błąd.
 - Śledźcie przebieg ruchu robota w labiryncie i porównajcie go z Waszym kodem na planszy lub na kartce. W przypadku błędnego zachowania robota w labiryncie lub innego przebiegu jazdy, niż w sekwencjach instrukcji uczniów poszukajcie przyczyny w kodzie-sekwencji.

4. Dobieraj kolejne poziomy plansz do umiejętności uczniów.



Podpowiedź: Nie wszystkie plansze z danej lekcji muszą być zrealizowane podczas zajęć. Rozwiązanie dwóch plansz na lekcji również może być wystarczające. Dostosuj tempo pracy do potrzeb uczniów.

Podsumowanie

Czas trwania: 5 min

Poproś uczniów o narysowanie lub napisanie o tym, czego się nauczyli i jakie emocje towarzyszyły im podczas zajęć.

Nauczanie rozszerzone

Jeśli jest taka potrzeba, zdefiniuj zadanie domowe dla chętnych, np. Podaj przykłady z życia, w których wykonujemy jakąś czynność, gdy spełnione są określone warunki.

Przygotowanie do zajęć

- Upewnij się, że uczniowie posiadają ołówek, zeszyt lub notatnik, w którym mogą zapisywać notatki, przemyślenia, rysować.

- Jeśli planujesz użyć podczas zajęć dodatkowe karty pracy lub inne materiały, upewnij się, że masz wystarczającą ich ilość dla uczniów.
- Jeśli planujesz przeprowadzić dyskusję, przygotuj dodatkowe pytania.
- Przejrzyj ćwiczenia przed lekcją, aby zidentyfikować potencjalne problematyczne obszary dla Twoich uczniów.

Lekcja 3: Metoda *podnieś* i *upuść*

Cele operacyjne

Uczeń:

- określa kierunki i ustala położenie robota w stosunku do własnej osoby, a także w stosunku do innych przedmiotów, rozróżnia stronę lewą i prawą, górę, dół,
- układa sekwencję instrukcji dla robota za pomocą bloczków ze strzałkami kierunkowymi,
- odczytuje sekwencję instrukcji zapisaną za pomocą bloczków ze strzałkami kierunkowymi,
- omija przeszkody w postaci ścian labiryntu,
- określa wzajemne położenie przedmiotów na płaszczyźnie, (nowy cel)
- stosuje w sekwencji instrukcji akcję “podnieś”(nowy cel),
- stosuje w sekwencji instrukcji akcję “upuść” (nowy cel).

Opis skrótowy

Stosując proste ćwiczenia w aplikacji *Jedź Lumi!* uczniowie będą tworzyć algorytmy sekwencyjne z akcjami *podnieś* i *upuść*. Będą poruszać robotem na interaktywnej planszy i kierować nim w labiryncie od pola Start do pola Meta, po drodze zabierając przedmioty i odkładając je we wskazane miejsce w labiryncie. Aby to zrobić, uczniowie ustawią odpowiednie bloczki z instrukcjami na interaktywnej planszy w jeden ciąg znaków - tworząc w ten sposób algorytm - sekwencję instrukcji dla robota.

Cele ćwiczeń

Podczas tej lekcji uczniowie rozwiną umiejętności programowania z prostymi akcjami typu *podnieś* i *upuść* oraz debugowania, tworząc w ten sposób bazę dla przyszłych ćwiczeń. Zrozumieją najbardziej podstawowe funkcje maszyn-robotów, takie jak podnoszenie i odkładanie/upuszczanie. Oparta na bloczkach forma instrukcji pomaga uczniom poznać idee metody *podnieś* i *upuść* oraz budowania kodu programu bez martwienia się o doskonalenie składni.

Plan zajęć

Czas trwania: 45 min

Rozgrzewka

Czas trwania: 15 min

Wybierz jedną z poniższych zabaw.

Zabawa: Układanie puzzli

Potrzebnych jest kilka prostych układanek (maksymalnie 20 elementów w układance): 1 zestaw na dwoje uczniów. Dobierz dzieci w pary. Poproś je o ułożenie puzzli w całość. Następnie poproś o wyjęcie 5 elementów z układanki i odłożenie ich na bok. Jedna osoba w parze przyjmie rolę ręki robota, za pomocą której dokończy układankę. Zadaniem drugiej osoby jest pokierowanie ręką robota za pomocą poleceń: *przesuń w prawo, przesuń w lewo, przesuń w górę, przesuń w dół, podnieś element, odtóż element*.

Zabawa: Jestem robotem

Zabawa odpowiednia dla młodszych dzieci.

Ćwiczenie w grupie:

Wybrana osoba jest robotem, który wykonuje proste polecenia pozostałych osób. Zabawa może nawiązywać do poleceń w aplikacji *Jedź Lumi!*: *idź w prawo, idź w lewo, idź do góry, idź do dołu, jeśli spotkasz kolegę, idź do góry*, itd. Ustalcie punkt odniesienia dla określania kierunków. Wprowadźcie do zabawy nowe akcje, np: *podnieś książkę z podłogi, odtóż książkę na półkę*, itd.

Zabawa: Kodowanie graficzne

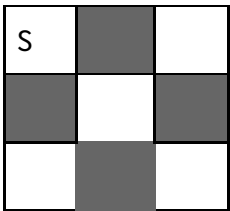
Zabawa dedykowana dla starszych uczniów.

W tej zabawie stworzymy instrukcję pokolorowania planszy. Potrzebna jest kwadratowa siatka pól 3x3 lub większa. Wystarczy kartka papieru w rozmiarze A4. Uczniowie mogą narysować siatkę samodzielnie. Upewnij się, że wszyscy mają kwadratową siatkę z jednakową ilością pól. Co drugie pole na siatce należy wypełnić kolorem (przykład - patrz Rysunek 1). Punkt startowy zaznaczcie w lewym górnym rogu literą S. Wyjaśnij dzieciom zasady kodowania:

Stwórzcie instrukcję pokolorowania planszy według wzoru. Dostępne instrukcje: strzałka kierunkowa w prawo $>$, w lewo $<$, do góry \wedge , na dół \vee , kolor lub symbol reprezentujący akcję, np. A.

Tabela: Kodowanie graficzne

Wzór do zakodowania	Kod rozwiązania
Rysunek 1	Rozwiązanie 1 $> A > \vee A < < A \vee > A >$

	<p>Rozwiązanie 2 $\vee A \vee > A > \wedge A \wedge < A \vee$</p> <p>Rozwiązanie n Jest wiele poprawnych rozwiązań tego zadania</p>
---	---

Dyskusja

Zawiązanie dyskusji na temat rozumienia, interpretacji poleceń, w szczególności akcji do wykonania typu: *podnieś, odłóż/upuść, wypełnij kolorem*, itp.

Nowe pojęcia

Podnieś i upuść - odpowiednik metody przeciągnij i upuść (z ang. Pick up and Drop, Drag and Drop - przeciągnij i upuść). Reprezentuje najbardziej podstawowe funkcje komputera, takie jak klikanie, przeciąganie i upuszczanie, a w przypadku robotów - podnoszenie i upuszczanie/odkładanie.

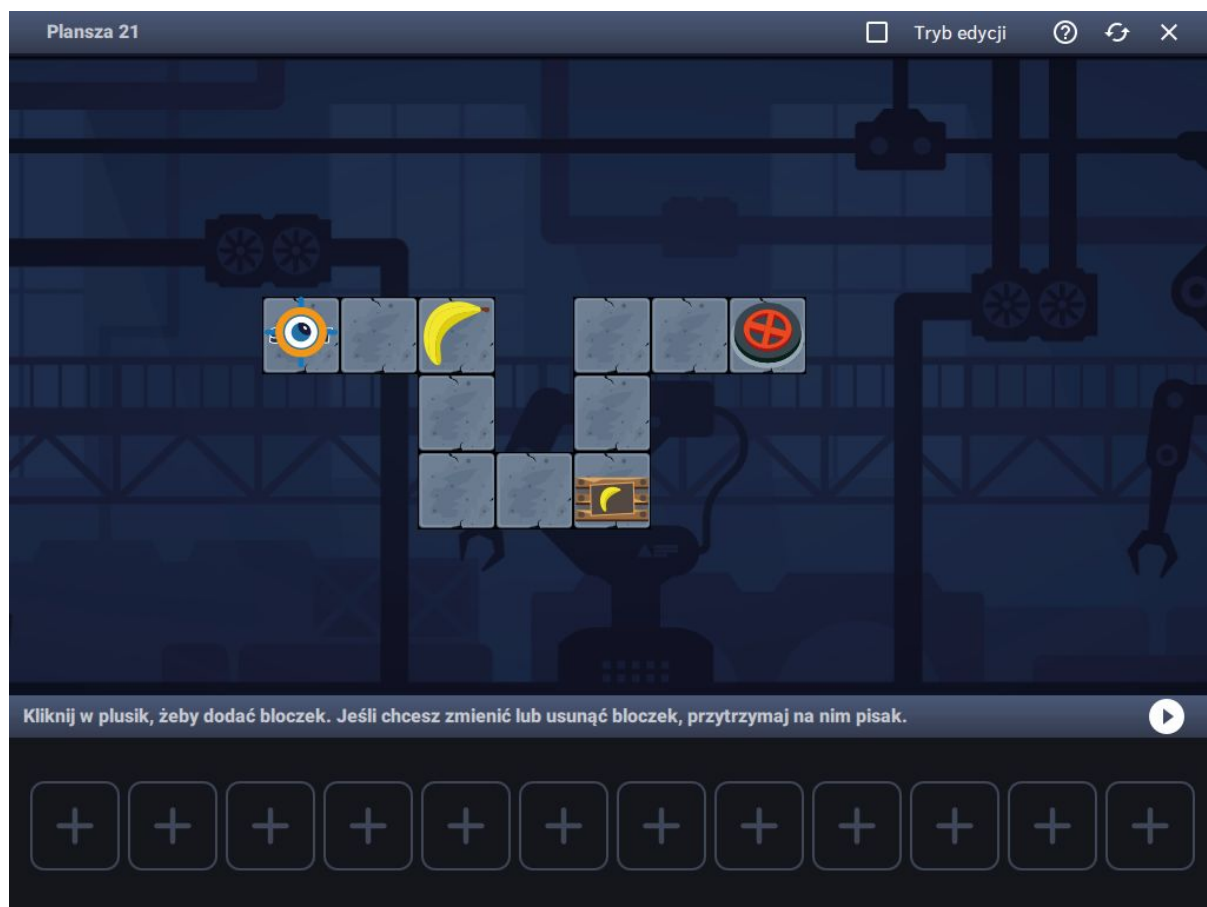
Ćwiczenia główne

Czas trwania: 25 min

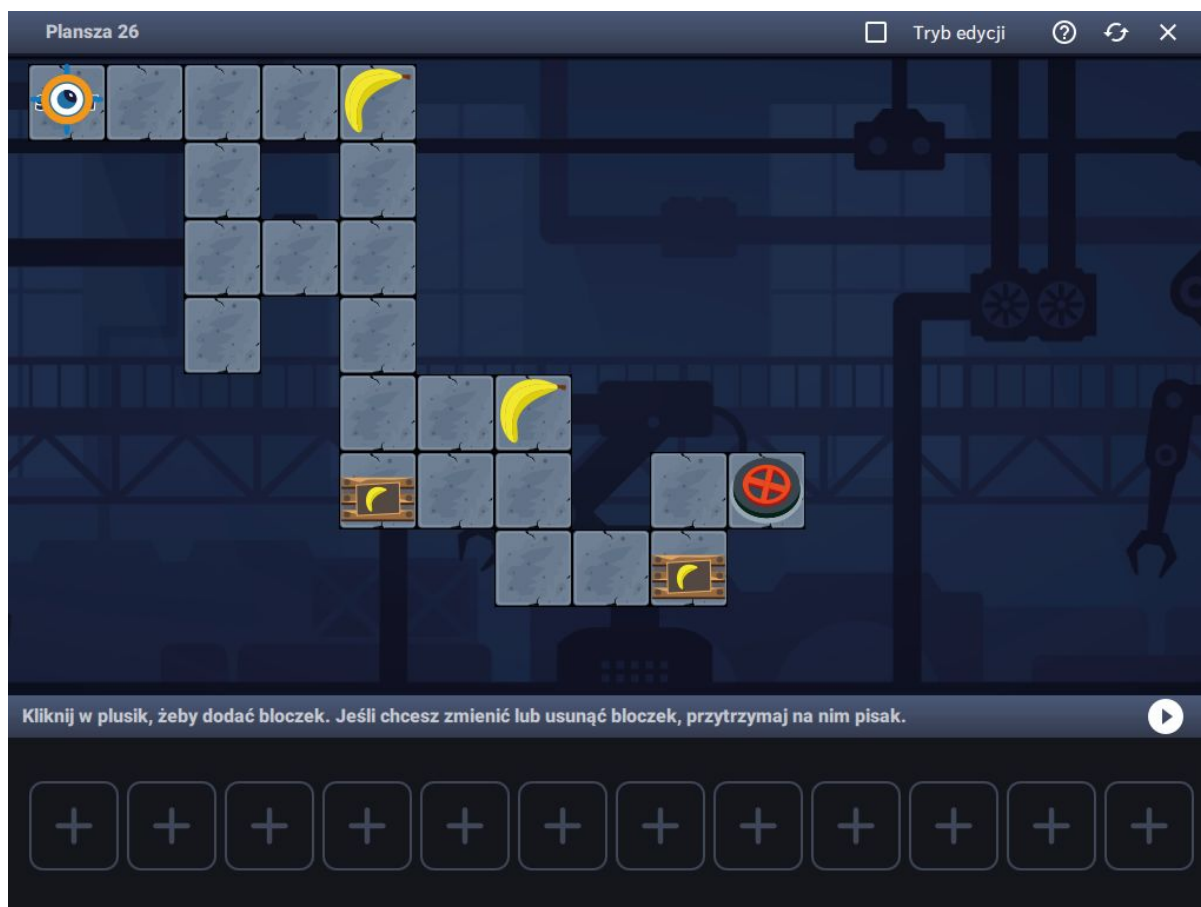
Jedź Lumi! Poziom 21-30

Ćwiczenia w aplikacji *Jedź Lumi!* dedykowane do lekcji nr 3: Poziom 21 - Poziom 30.

1. Wybierz najprostszą planszę odpowiednią do tematu lekcji - Poziom 21 - i przy pomocy uczniów zademonstruj zasadę pracy w aplikacji. Wspólnie rozwiążcie pierwszą planszę.
2. Uruchamiając kod programu, zawołajcie *Jedź + imię waszego robota!*



3. Wybierz dogodny sposób pracy z grupą na interaktywnej podłodze:
 - Podziel uczniów na zespoły lub pary, które będą rozwiązywać poszczególne plansze.
 - Wszystkie zespoły mogą pracować równolegle, projektując sekwencję instrukcji dla robota na kartce papieru lub za pomocą samodzielnie przygotowanych tekturowych bloczków. Wybrany zespół przepisuje lub układa kod na planszy.
 - Sprawdźcie wspólnie wszystkie lub wybrane rozwiązania zespołów na podłodze interaktywnej.
 - Wybierzcie osobę, która będzie naśladować robota i wykonywać polecenia według zaprojektowanych sekwencji, poruszając się w labiryncie na planszy. Następnie uruchomcie kod i porównajcie rezultaty. Jeśli się różnią, to poszukajcie, gdzie się wkradł błąd.
 - Śledźcie przebieg ruchu robota w labiryncie i porównajcie go z Waszym kodem na planszy lub na kartce. W przypadku błędnego zachowania robota w labiryncie lub innego przebiegu jazdy, niż w sekwencjach instrukcji uczniów poszukajcie przyczyny w kodzie-sekwencji.
4. Dobieraj kolejne poziomy plansz do umiejętności uczniów.



Podpowiedź: Nie wszystkie plansze z danej lekcji muszą być zrealizowane podczas zajęć. Rozwiązanie dwóch plansz na lekcji również może być wystarczające. Dostosuj tempo pracy do potrzeb uczniów.

Podsumowanie

Czas trwania: 5 min

Porozmawiaj z uczniami lub poproś ich o narysowanie lub napisanie o tym, czego się nauczyli i jakie emocje towarzyszyły im podczas zajęć.

Nauczanie rozszerzone

Jeśli jest taka potrzeba, zdefiniuj zadanie domowe dla chętnych, np.

- Narysuj inny obraz, który możesz zakodować. Czy możesz napisać program, który będzie pasował do tego rysunku?
- Jakie rodzaje robotów moglibyśmy zaprogramować, gdybyśmy zmienili znaczenie strzałek lub akcji w waszym kodzie graficznym?
- Czy znasz przykłady maszyn, których sposób pracy moglibyśmy opisać za pomocą kodu graficznego?

Przygotowanie do zajęć

- Upewnij się, że uczniowie posiadają ołówek lub długopis, zeszyt lub notatnik, w którym mogą zapisywać notatki, przemyślenia, rysować.
- Jeśli planujesz użyć podczas zajęć dodatkowych kart pracy lub układanek, puzzli, upewnij się, że masz wystarczającą ich ilość dla uczniów.
- Jeśli planujesz przeprowadzić dyskusję, przygotuj dodatkowe pytania.
- Przejrzyj ćwiczenia przed lekcją, aby zidentyfikować potencjalne problematyczne obszary dla Twoich uczniów.

Lekcja 5: Wprowadzamy pętlę iteracyjną

Cele operacyjne

Uczeń:

- określa kierunki i ustala położenie robota w stosunku do własnej osoby, a także w stosunku do innych przedmiotów, rozróżnia stronę lewą i prawą, górę, dół,
- układa sekwencję instrukcji dla robota za pomocą bloczków ze strzałkami kierunkowymi,
- odczytuje sekwencję instrukcji zapisaną za pomocą bloczków ze strzałkami kierunkowymi,
- omija przeszkody w postaci ścian labiryntu,
- wyodrębnia w długiej sekwencji instrukcji powtarzaną ciągle sekwencję (nowy cel),
- układa sekwencję instrukcji, stosując pętlę iteracyjną (nowy cel).

Opis skrótowy

Stosując proste ćwiczenia w aplikacji *Jedź Lumi!* uczniowie będą tworzyć algorytmy sekwencyjne z pętlą iteracyjną, aby poruszać robotem na interaktywnej planszy. Pokierują robotem w labiryncie od pola Start do pola Meta. Aby to zrobić, uczniowie ustawią odpowiednie bloczki z instrukcjami na interaktywnej planszy w jeden ciąg znaków - tworząc w ten sposób sekwencję instrukcji dla robota.

Cele ćwiczeń

Podczas tej lekcji uczniowie rozwiną umiejętności programowania z pętlą iteracyjną i debugowania, aby dotrzeć do końca labiryntu. Zrozumieją korzyści wynikające ze stosowania struktury pętli, zamiast ręcznego powtarzania tych samych instrukcji. Nauczą się wyodrębniać

w długiej sekwencji instrukcji powtarzaną ciągle sekwencją. Oparta na bloczkach forma instrukcji pomaga uczniom poznać i zrozumieć sekwencję, ideę pętli iteracyjnej i budowania oraz uproszczania kodu programu bez martwienia się o doskonalenie składni.

Plan zajęć

Czas trwania: 45 min

Rozgrzewka

Czas trwania: 15 min

Wybierz jedną z poniższych zabaw.

Zabawa: Jestem robotem

Zabawa dedykowana dla starszych uczniów.

Ćwiczenie w grupie. Wybrana osoba jest robotem, który wykonuje proste polecenia jednej z osób. Podstawowa zasada w ćwiczeniu to stosować to samo polecenie kilka razy z rzędu.

Po ostatnim wykonaniu polecenia:

Zapytaj: Czy byłoby tobie łatwiej po prostu poprosić naszego kolegę o kilkukrotne wykonanie tej czynności?

Zachęć do przemyślenia: Chcemy, aby kolega wykonał czynność dziesięć razy. Jak zmienilibyście waszą instrukcję, aby nie trzeba było tak często powtarzać tego samego polecenia? Zapiszcie swoje instrukcje na kartkach.

Dzielcie się pomysłami: Poproś kilkoro uczniów, aby podzielili się instrukcjami z klasą, wskazując, w jaki sposób każde rozwiązanie uprościło podejście do wydawania poleceń.

Podsumuj: Dzisiaj będziemy pracować nad znalezieniem sposobów na ułatwienie wydawania wielu instrukcji, zwłaszcza gdy te instrukcje często się powtarzają.

Zabawa: Zapętlone Koderki

Zabawa dedykowana dla młodszych dzieci.

Jeśli na podłodze interaktywnej jest zainstalowana aplikacja *Zapętlone Koderki* lub *Buduj połączenia* możesz z niej skorzystać, aby dzieci bawiąc się, poznały ideę pętli iteracyjnej. W zabawie tej należy wykonać określoną ilość ruchów - tupnięć, aby połączyć dwa Koderki.

Jeśli nie dysponujesz taką aplikacją interaktywną, wykonaj podobne ćwiczenie na podłodze. Pary dzieci ustawiają się naprzeciwko siebie w różnych odległościach. Jedno dziecko z pary powtarza polecenie: *zrób krok*, a drugie robi tylko jeden krok w kierunku pierwszego. Osoba, która wykonuje kroki, liczy je na głos. Polecenie jest powtarzane dopóki dziecko nie dotrze do kolegi z pary. Dzieci mówią, ile zrobiły kroków i wracają do początkowego ustawienia.

Następnie poproś ucznia, aby dał koledze tylko jedno polecenie zamiast kilku, w którym mu powie ile ma zrobić kroków. Drugie dziecko z pary wykonuje polecenie.

Zapytaj: Czy było tobie łatwiej poprosić raz swojego kolegę o zrobienie kilku kroków, zamiast mówić mu za każdym razem co ma zrobić?

Podsumuj: Dzisiaj poznamy sposoby, jak ułatwić dawanie wielu takich samych instrukcji, aby nie trzeba było ich wszystkich wypisywać.

Nowe pojęcia

Pętla - czynność polegająca na robieniu czegoś w kółko

Powtórzenie - Zrobić coś jeszcze raz

Iteracja - Powtarzanie

Pętla iteracyjna - Powtarzanie czegoś ustaloną ilość razy

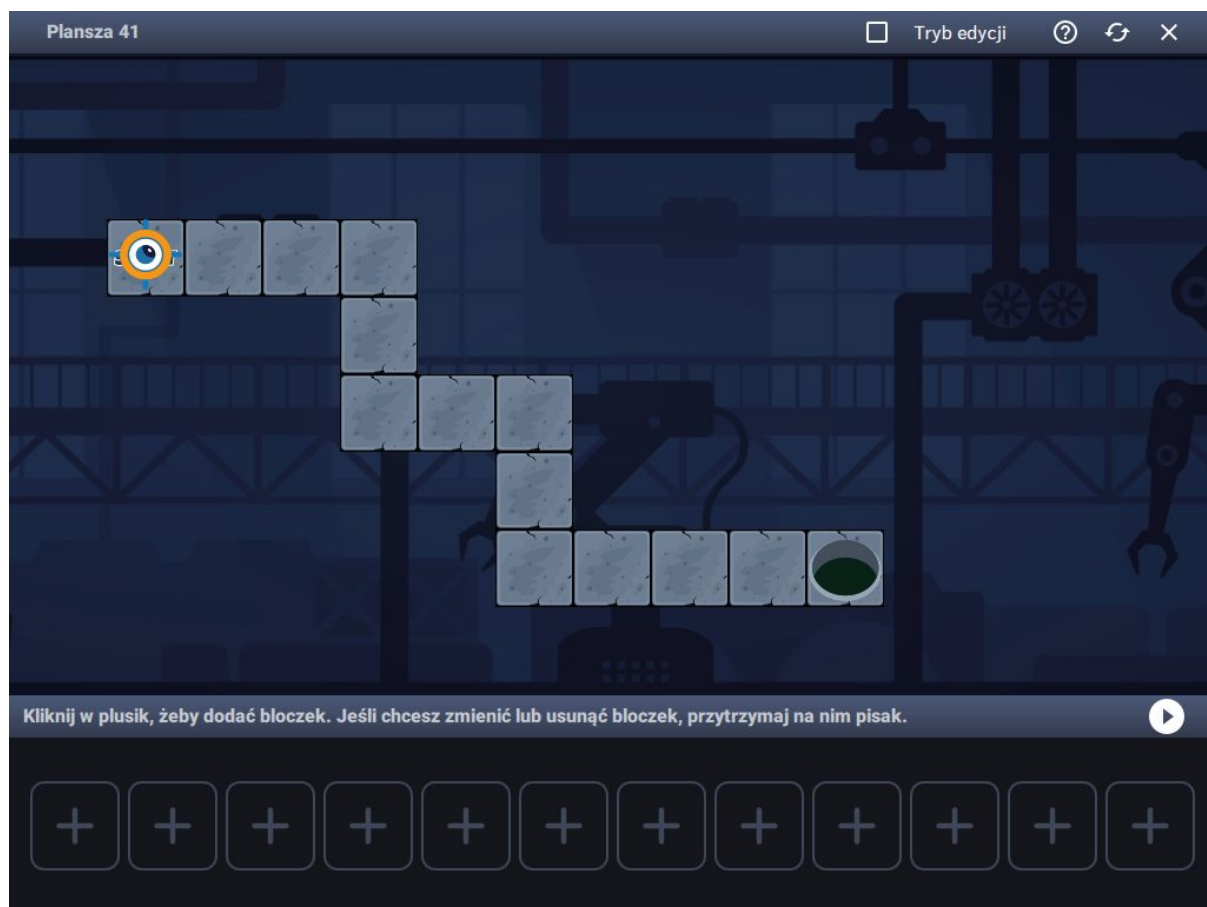
Ćwiczenia główne

Czas trwania: 25 min

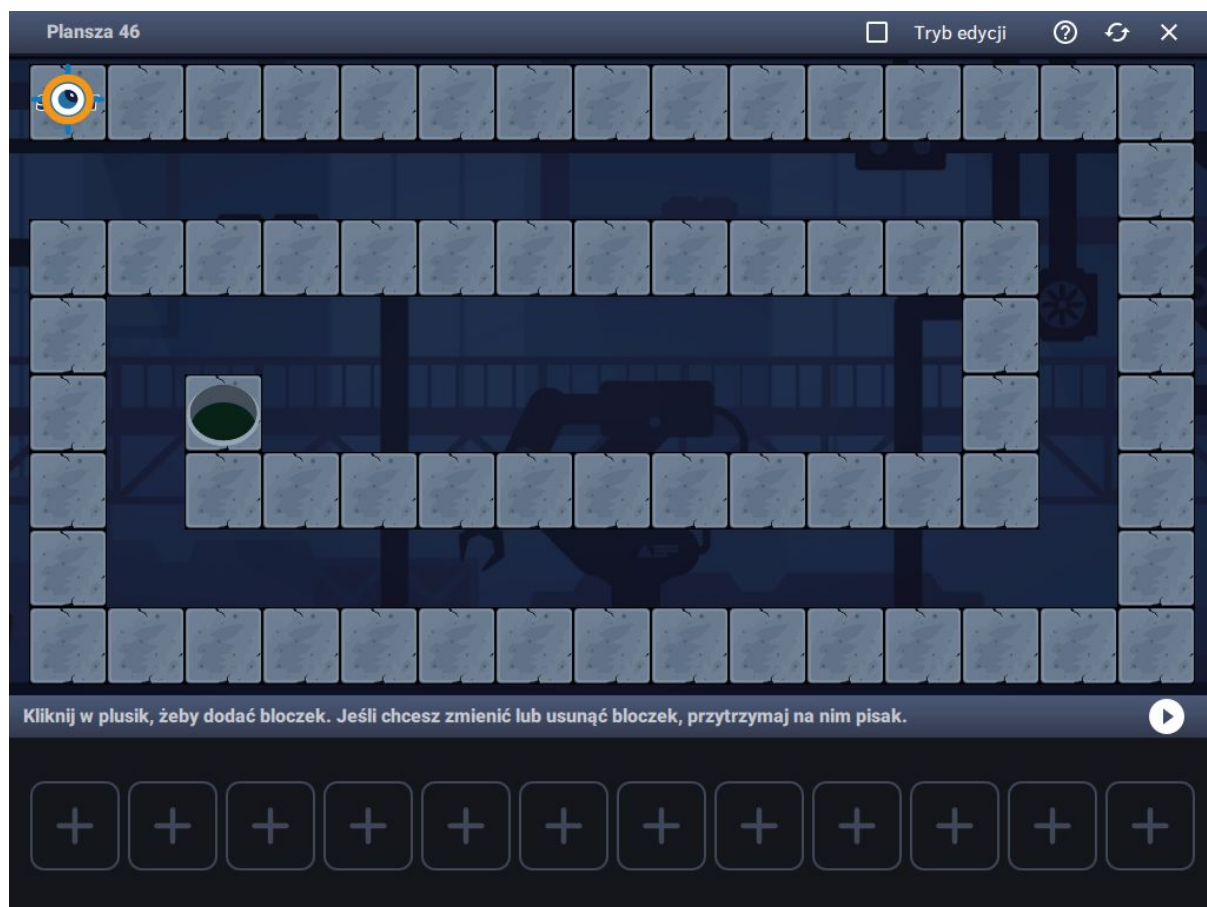
Jedź Lumi! Poziom 41 - 50

Ćwiczenia w aplikacji *Jedź Lumi!* dedykowane do lekcji nr 2: Poziom 41 - Poziom 50.

1. Wybierz najprostszą planszę odpowiednią do tematu lekcji - Poziom 51 - i przy pomocy uczniów zademonstruj zasadę pracy w aplikacji. Wspólnie rozwiążcie pierwszą planszę.



2. Uruchamiając kod programu, zawołajcie *Jedź + imię* waszego robota!
3. Wybierz dogodny sposób pracy z grupą na interaktywnej podłodze:
 - Podziel uczniów na zespoły lub pary, które będą rozwiązywać poszczególne plansze.
 - Wszystkie zespoły mogą pracować równolegle, projektując sekwencję instrukcji dla robota na kartce papieru lub za pomocą samodzielnie przygotowanych tekturowych bloczków. Wybrany zespół przepisuje lub układa kod na planszy.
 - Sprawdźcie wspólnie wszystkie lub wybrane rozwiązania zespołów na podłodze interaktywnej.
 - Wybierzcie osobę, która będzie naśladować robota i wykonywać polecenia według zaprojektowanych sekwencji, poruszając się w labiryncie na planszy. Następnie uruchomcie kod i porównajcie rezultaty. Jeśli się różnią, to poszukajcie, gdzie się wkradł błąd.
 - Śledźcie przebieg ruchu robota w labiryncie i porównajcie go z Waszym kodem na planszy lub na kartce. W przypadku błędnego zachowania robota w labiryncie lub innego przebiegu jazdy, niż w sekwencjach instrukcji uczniów poszukajcie przyczyny w kodzie-sekwencji.
4. Dobieraj kolejne poziomy plansz do umiejętności uczniów.



Podpowiedź: Nie wszystkie plansze z danej lekcji muszą być zrealizowane podczas zajęć. Rozwiązanie dwóch plansz na lekcji również może być wystarczające. Dostosuj tempo pracy do potrzeb uczniów.

Podsumowanie

Czas trwania: 5 min

Poproś uczniów o narysowanie lub napisanie o tym, czego nauczyli się podczas lekcji i jakie emocje towarzyszyły im podczas wykonywania ćwiczeń. Mogą również zapisać swoje pytania.

Nauczanie rozszerzone

Jeśli jest taka potrzeba, zdefiniuj zadanie domowe dla chętnych, np. Zastanówcie się i odpowiedzcie na pytanie: W jakich sytuacjach w życiu codziennym spotykamy się z pętlą?

Przygotowanie do zajęć

- Upewnij się, że uczniowie posiadają ołówek lub długopis, zeszyt lub notatnik, w którym mogą zapisywać notatki, przemyślenia, rysować.
- Jeśli planujesz użyć podczas zajęć dodatkowych kart pracy lub innych materiałów, upewnij się, że masz wystarczającą ich ilość dla uczniów.
- Jeśli planujesz przeprowadzić dyskusję, przygotuj dodatkowe pytania.

- Przejrzyj ćwiczenia przed lekcją, aby zidentyfikować potencjalne problematyczne obszary dla Twoich uczniów.

Lekcja 8: Tworzymy funkcję

Cele operacyjne

Uczeń:

- określa kierunki i ustala położenie robota w stosunku do własnej osoby, a także w stosunku do innych przedmiotów, rozróżnia stronę lewą i prawą, górę, dół,
- układa sekwencję instrukcji dla robota za pomocą bloczków ze strzałkami kierunkowymi,
- odczytuje sekwencję instrukcji zapisaną za pomocą bloczków ze strzałkami kierunkowymi,
- omija przeszkody w postaci ścian labiryntu,
- potrafi wyodrębnić z długiej sekwencji największą powtarzalną sekwencję (nowy cel),
- umie zmienić fragment kodu w funkcję (nowy cel),
- układa sekwencję instrukcji z wywołaniem funkcji (nowy cel).

Opis skrótowy

Stosując proste ćwiczenia w aplikacji *Jedź Lumi!* uczniowie będą tworzyć algorytmy sekwencyjne z wykorzystaniem funkcji, aby poruszać robotem na interaktywnej planszy i pokierować nim w labiryncie od pola Start do pola Meta. Aby to zrobić, uczniowie ustawią odpowiednie bloczki z instrukcjami na interaktywnej planszy w ciągu znaków - tworząc w ten sposób definicję funkcji oraz algorytm - sekwencję instrukcji - dla robota.

Cele ćwiczeń

Podczas tej lekcji uczniowie rozwiną umiejętności definiowania funkcji i programowania z wykorzystaniem funkcji oraz umiejętność debugowania. Ćwiczenia pomogą uczniom zrozumieć dlaczego zmienianie fragmentów kodu w funkcje może być pomocną praktyką. Nauczą się wyodrębnić z długiej sekwencji największą powtarzalną sekwencję. Oparta na bloczkach forma instrukcji pomaga uczniom poznać i zrozumieć sekwencję, funkcję oraz ideę budowania i uproszczania kodu programu bez martwienia się o doskonalenie składni.

Plan zajęć

Czas trwania: 45 min

Rozgrzewka

Czas trwania: 15 min

Zabawa: Śpiewamy piosenkę

Ćwiczenie w grupie:

Zacznij od prostej piosenki, napisanej lub wyświetlonej na ekranie, najlepiej takiej, którą znają uczniowie. Ustal z nimi, że na znak refrenu wszyscy razem śpiewacie refren. Wskaż refren i upewnij się, że klasa rozumie zasadę, zanim zaczniesz utwór. Zaczynajcie śpiewać piosenkę.

Jest całkiem prawdopodobne, że większość klasy zaśpiewa słowa do refrenu, gdy pokażesz fragment wskazujący na refren. Po kolejnym refrenie zatrzymaj piosenkę.

Zapytaj: Co się właśnie wydarzyło?

Zachęć do przemyślenia: Zdefiniowaliśmy refren. Wywołałam refren. Zaśpiewaliście refren. Dlaczego refren mamy zapisany tylko jeden raz na górze utworu?

Dzielcie się pomysłami: Wystuchaj wszystkich odpowiedzi. Zadawaj pomocnicze pytania: Jakie są inne korzyści z zapisania refrenu tylko jeden raz, kiedy śpiewamy go wiele razy?

Podsumuj: Piosenka jest jak program komputerowy. Definiowanie nazwy takiej jak *Refren* dla małego fragmentu utworu, którego używasz często, nazywa się tworzeniem funkcji. Jest to pomocne dla informatyków z tych samych powodów, dla których jest pomocne dla autorów piosenek. Dzisiaj poznamy jak ta technika jest używana w programowaniu.

Nowe pojęcia

Funkcja - powtórzcie razem: Funkcja

Funkcja to kawałek kodu, pewna sekwencja instrukcji, którą można zdefiniować jeden raz i wywoływać (używać) wiele razy w programie.

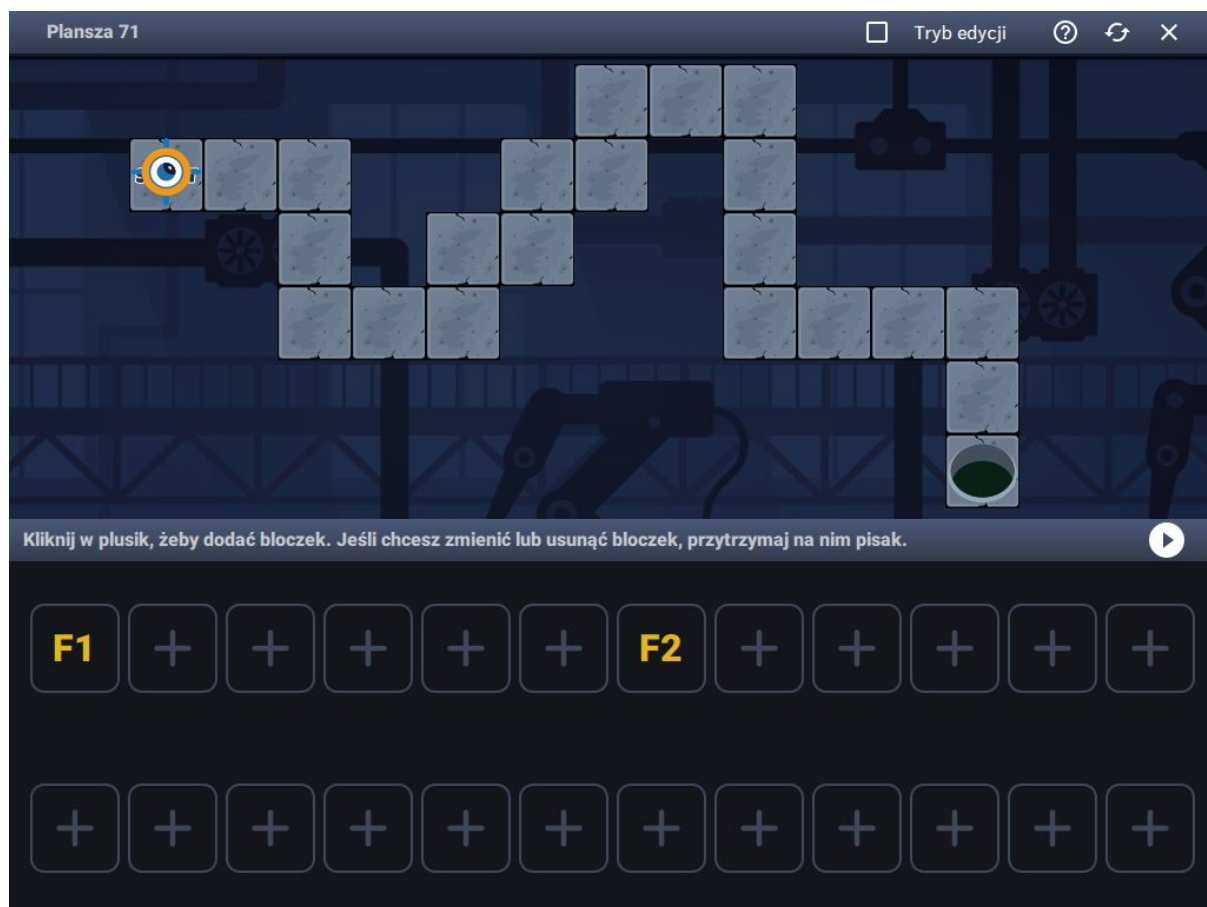
Ćwiczenia główne

Czas trwania: 25 min

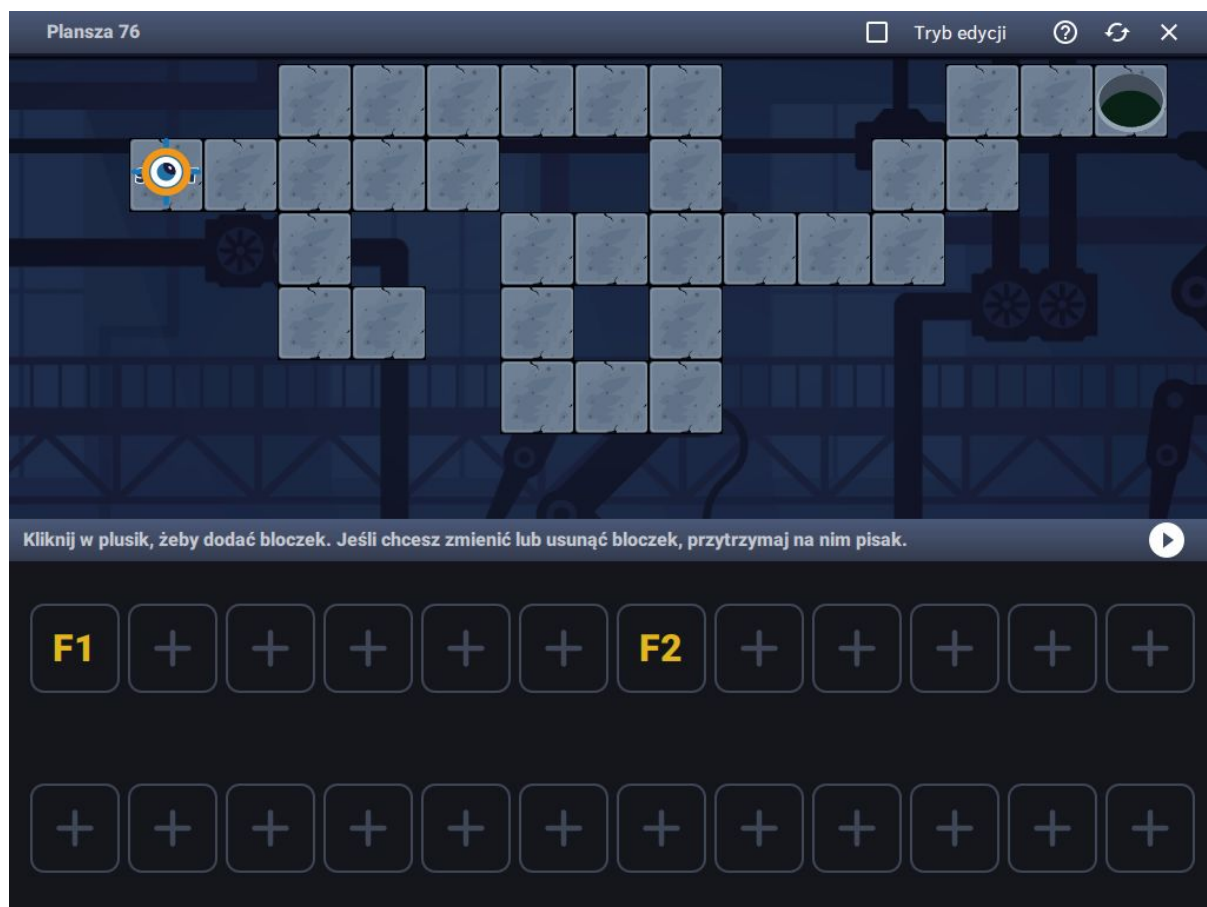
Jedź Lumi! Poziom 71 - 80

Ćwiczenia w aplikacji *Jedź Lumi!* dedykowane do lekcji nr 2: Poziom 71 - Poziom 80.

1. Wybierz najprostszą planszę odpowiednią do tematu lekcji - Poziom 71 - i przy pomocy uczniów zademonstruj zasadę pracy w aplikacji oraz wyjaśnij zasadę budowania funkcji. Rzuć uczniom wyzwanie, aby znaleźli na planszy układ drogi w labiryncie, który powtarza się. Pozwól uczniom zdefiniować pierwszą funkcję. A potem wspólnie rozwiążcie pierwszą planszę.



2. Uruchamiając kod programu, możecie wołać *Jedź + imię waszego robota!*
3. Wybierz dogodny sposób pracy z grupą na interaktywnej podłodze:
 - Podziel uczniów na zespoły lub pary, które będą rozwiązywać poszczególne plansze.
 - Wszystkie zespoły mogą pracować równolegle, projektując sekwencję instrukcji dla robota na kartce papieru lub za pomocą samodzielnie przygotowanych tekturowych bloczków. Wybrany zespół przepisuje lub układa kod na planszy.
 - Sprawdźcie wspólnie wszystkie lub wybrane rozwiązania zespołów na podłodze interaktywnej.
 - Wybierzcie osobę, która będzie naśladować robota i wykonywać polecenia według zaprojektowanych sekwencji, poruszając się w labiryncie na planszy. Następnie uruchomcie kod i porównajcie rezultaty. Jeśli się różnią, to poszukajcie, gdzie się wkraść błąd.
 - Śledźcie przebieg ruchu robota w labiryncie i porównajcie go z Waszym kodem na planszy lub na kartce. W przypadku błędnego zachowania robota w labiryncie lub innego przebiegu jazdy, niż w sekwencjach instrukcji uczniów poszukajcie przyczyny w kodzie-sekwencji.
4. Dobieraj kolejne poziomy plansz do umiejętności uczniów.



Podpowiedź: Nie wszystkie plansze z danej lekcji muszą być zrealizowane podczas zajęć. Rozwiązanie dwóch plansz na lekcji również może być wystarczające. Dostosuj tempo pracy do potrzeb uczniów.

Podsumowanie

Czas trwania: 5 min

Poproś uczniów o narysowanie lub napisanie o tym, czego nauczyli się podczas lekcji i jakie emocje towarzyszyły im podczas wykonywania ćwiczeń. Poproś ich o zapisanie pytań, jeśli je mają.

Nauczanie rozszerzone

Jeśli jest taka potrzeba, zdefiniuj zadanie domowe dla chętnych, np.

1. Wymień przykłady różnych urządzeń w domu. Jakie funkcje realizują te sprzęty?
2. Stwórz piosenkę z refrenem. Zapisz refren na górze i odwołuj się do niego po każdej zwrotce. Jeśli refren ma być powtarzany parę razy z rzędu po zwrotce, zaznacz to w utworze.

Przygotowanie do zajęć

- Możesz wyszukać teksty niektórych popularnych piosenek w Internecie. Pokaż uczniom, że standardem powtarzania słów w utworach jest zdefiniowanie refrenu u góry i wywołanie go w treści utworu, po każdej zwrotce.
- Upewnij się, że uczniowie posiadają ołówek lub długopis, zeszyt lub notatnik, w którym mogą zapisywać notatki, przemyślenia, rysować.
- Jeśli planujesz użyć podczas zajęć dodatkowe karty pracy lub inne materiały, upewnij się, że masz wystarczającą ich ilość dla uczniów.
- Jeśli planujesz przeprowadzić dyskusję, przygotuj dodatkowe pytania.
- Przejrzyj ćwiczenia przed lekcją, aby zidentyfikować potencjalne problematyczne obszary dla Twoich uczniów.