



MoboLab – roboty i tablety w Twojej szkole

Obszar I. „Zakoduj robota”

Scenariusze lekcji i zajęć pozalekcyjnych

SCENARIUSZ 3. POZNAJEMY I PROGRAMUJEMY CZUJNIK ŚWIATŁA

scenariusz lekcji informatyki (możliwy do realizacji także na zajęciach pozalekcyjnych)

autor: Kamil Kociszewski

redakcja: Agnieszka Koszowska

SŁOWA KLUCZOWE:

mBot, mBlock, czujnik, natężenie światła, światło, RGB, led, warunki, instrukcje

KRÓTKI OPIS LEKCJI:

Podczas lekcji uczniowie i uczennice rozwijają wiedzę o środowisku programistycznym mBlock opartym na **języku Scratch**. Poznają **czujnik światła** oraz jego zastosowanie, utrwalają wiedzę na temat wykorzystania diod robota. Uczą się wykorzystywać działania matematyczne w sterowaniu programem. Na zakończenie wykonują zadanie: tworzą skrypt, który sprawia, że robot reaguje na zwiększenie oraz zmniejszenie natężenia światła w pomieszczeniu.

WIEDZA I UMIEJĘTNOŚCI ZDOBYTE PRZEZ UCZNIĄ / UCZENNICĘ:

- zna budowę robota mBot,
- zna podstawowe elementy interfejsu programu mBlock,
- wie, co to jest czujnik światła,
- swobodnie porusza się po środowisku mBlock, wie, gdzie szukać bloków do tworzenia skryptów sterujących czujnikiem światła,
- potrafi stworzyć prosty skrypt sterujący czujnikiem światła,
- zna pojęcie kalibracji, potrafi skalibrować czujnik światła,
- umie wykorzystać pomiar z czujnika do sterowania innymi częściami robota,
- potrafi stworzyć prosty program w środowisku programistycznym mBlock, wykorzystując działania matematyczne i instrukcje warunkowe.

GRUPA DOCELOWA:

Starsze klasy szkoły podstawowej (VII-) i klasy gimnazjalne (po dostosowaniu: możliwość realizacji w młodszych klasach: I-III i IV-VI szkoły podstawowej)

LICZBA UCZNIÓW/UCZENNIC W KLASIE:

do 25 osób (z możliwością dostosowania scenariusza do potrzeb klas o różnej liczbie osób)

CZAS TRWANIA ZAJĘĆ:

90 min (lub 2 x 45 minut)

STOPIEŃ TRUDNOŚCI/SKOMPLIKOWANIA

(w skali od 1 do 5 dla obszaru I. „Zakoduj robota”):

2

POTRZEBNY SPRZĘT I OPROGRAMOWANIE:

- komputer (przenośny lub stacjonarny),
- program mBlock (do pobrania ze strony: <http://www.mblock.cc/download/>),
- roboty mBot (złożone) – 1 robot na 1 ucznia / uczennicę, a w przypadku mniejszej liczby robotów: 1 robot na 2 lub 3 uczniów / uczennic,
- kable USB (po 1 dla każdego robota),
- projektor i laptop (w części teoretycznej).

CO NALEŻY PRZYGOTOWAĆ PRZED ZAJĘCIAMI:

- zainstalować program mBlock,
- sprawdzić poprawne działanie robota mBot oraz połączenie z programem mBlock (jeśli wystąpią problemy, warto zainstalować ponownie sterownik Arduino),
- dopasować stopień trudności zadania do potrzeb i możliwości klasy, dla której organizowana jest lekcja według wskazówek zawartych w scenariuszu,
- sprawdzić, czy wszystkie elementy robota są prawidłowo podpięte i czy diody działają poprawnie,
- sprawdzić stan baterii zasilających robota.

KOMPETENCJE OSOBY PROWADZĄCEJ:

- zna i rozumie działanie wykorzystywanych bloków w programach Scratch i mBlock,
- potrafi podłączyć robota do komputera, używając kabla USB,
- wie, jakich bloków należy użyć do sterowania czujnikiem światła, włączania i wyłączania diod,
- zna podstawowe pojęcia programistyczne (skrypt, program, pętla, instrukcja warunkowa),
- wie, dlaczego warto uczyć się programowania i jakie korzyści daje posiadanie umiejętności programistycznych,
- potrafi zachęcić do nauki programowania zarówno chłopców, jak i dziewczynki.

PRZEBIEG ZAJĘĆ:

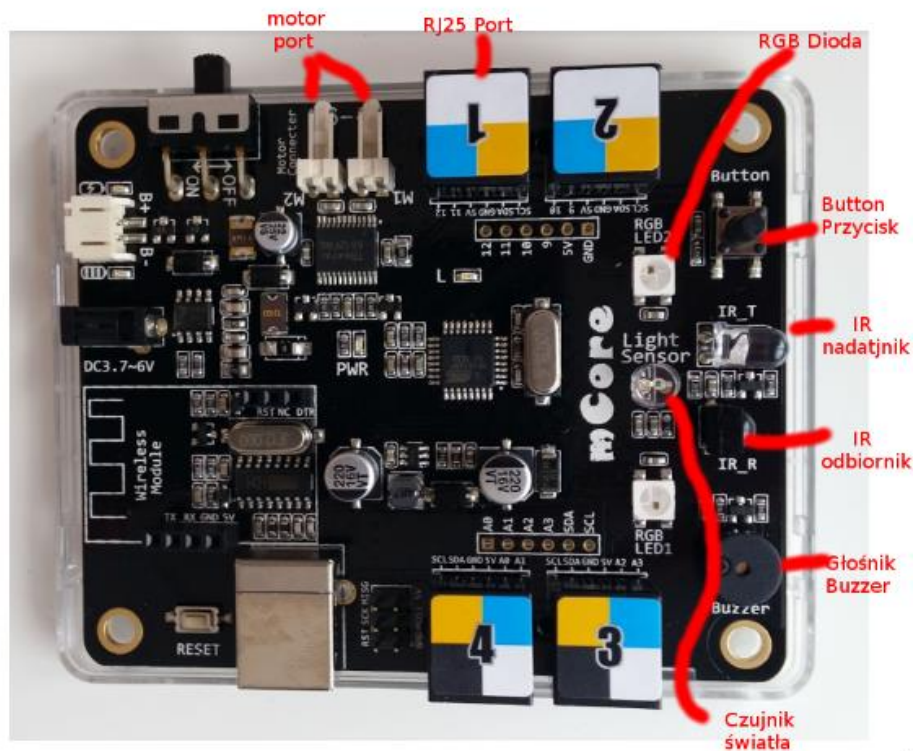
Część 1. – 45 minut

Wprowadzenie, omówienie czujnika światła – 10 minut

Cel: wprowadzenie uczniów i uczennic w tematykę lekcji, rozmowa o zastosowaniach czujnika światła.

Przedstawiamy tematykę lekcji. Zapowiadamy, że na dwóch kolejnych lekcjach uczniowie będą badać natężenie światła korzystając ze specjalnego urządzenia, w które jest wyposażony nasz robot – czujnika światła. Pojęcie „natężenie światła” można wyjaśnić na przykładzie lampki: im jaśniej świeci lampka, tym większe jest natężenie światła. Zadajemy uczniom pytania, czy znają przykłady czujników natężenia światła i jakie może być ich wykorzystanie. Jest to na przykład zapalanie się latarni podczas zmierzchu, czujnik zmierzchu w aucie czy czujnik dobierający automatycznie jasność ekranu w smartfonie.

Następnie krótko omawiamy czujnik znajdujący się w robocie, korzystając z poniższego zdjęcia (lub zrobionych samodzielnie zdjęć).



Na zdjęciu widać płytę Arduino, czyli „mózg” naszego robota. Oprócz wejść, do których możemy podłączyć różne moduły, posiada ona także wbudowane na stałe czujniki. Na tej lekcji będziemy korzystać z czujnika światła, znajdującego się w przedniej części płytki. Jego działanie można prosto wyjaśnić w następujący sposób: przez czujnik przepływa prąd, którego natężenie zmienia się pod wpływem światła – im więcej światła pada na czujnik, tym większe natężenie prądu.

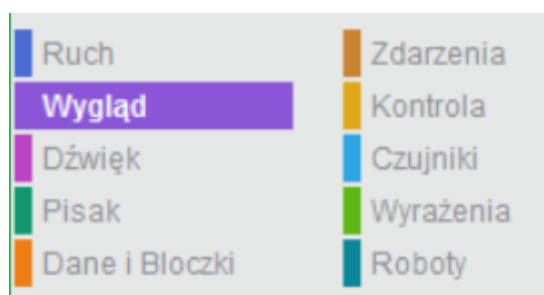
Stworzenie programu pozwalającego na kalibrację czujnika – 25 minut

Cel: uczniowie dowiadują się o potrzebie i sposobie kalibracji czujnika światła. Tworzą program, który umożliwi przeprowadzenie kalibracji.

Prowadzący wyjaśnia uczestnikom potrzebę kalibracji czujnika. Wiele czujników wymaga kalibracji do poprawnego działania w danym miejscu. Nasz czujnik będzie pokazywał liczbę odpowiadającą poziomowi oświetlenia czujnika. Musimy znaleźć taką liczbę, którą czujnik pokaże przy zwykłym oświetleniu. Będzie to nasza wartość „zero”, która pozwoli nam tak zaprogramować czujnik, by zareagował na zmianę natężenia padającego na niego światła.

Do przeprowadzenia kalibracji czujnika będzie nam potrzebny blok odpowiadający za sprawdzanie jego wartości oraz coś, co pozwoli pokazać tę wartość na ekranie. W tym celu wykorzystamy duszka – „Pandę”.

Warto w tym miejscu przypomnieć kategorie bloków, którymi się posługujemy programując naszego robota.



Zdarzenia: bloki z tej kategorii służą do programowania interakcji z użytkownikami – tworzenia skryptów, które reagują na określone działania użytkownika.

Roboty: bloki z tej kategorii służą do programowania interakcji z robotem – tworzenia skryptów, które umożliwiają sterowanie robotem i reakcję na zdarzenia oraz inicjowanie i kontrolę zdarzeń z udziałem poszczególnych elementów robota (np. czujników).

Kontrola: bloki z tej kategorii pozwalają sterować programem, na przykład dodawać do skryptu warunek, pętlę albo opóźnić wykonanie skryptu.

Wyrażenia: bloki z tej kategorii pozwalają wprowadzać do skryptu działania matematyczne lub wyrażenia logiczne.

Dodatkowo, w celu wyświetlenia wartości czujnika, skorzystamy z kategorii **Wygląd**. Bloki z tej kategorii umożliwiają graficzną interakcję z duszkami w lewym oknie

programu. Pozwalają zmienić wygląd duszka i komunikować się z nim, np. wyświetlić jego wypowiedź.



Za pomocą bloku „Powiedz...” możemy wyświetlić tekst, wartości zmiennych, a także to, czego teraz potrzebujemy: wartość czujnika.

W tym miejscu prosimy uczniów o znalezienie bloku odpowiedzialnego za pomiar natężenia światła. Jeśli pamiętają materiał z poprzednich lekcji, znajdą go w kategorii „Roboty”.



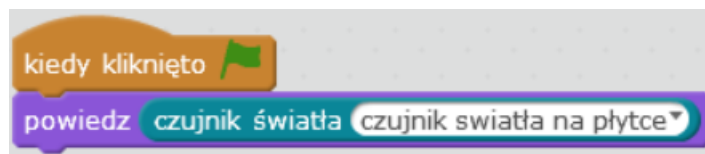
Blok ten daje możliwość pomiaru z zewnętrznych czujników, lecz na naszych zajęciach będziemy wykorzystywać czujnik na płytce. Ostatnim elementem, który będzie nam potrzebny do stworzenia skryptu, jest coś, co uruchomi program. Takie bloki znajdują się w kategorii „Zdarzenia”. Na poprzednich lekcjach korzystaliśmy z bloków reagujących na przycisk klawiatury, natomiast w tym przypadku potrzebujemy ciągłego pomiaru. Umożliwi nam to blok „kiedy kliknięto [zieloną flagę]”.



Będzie on uruchamiał działanie programu, kiedy klikniemy zieloną flagę nad pandą, a przerywał, gdy klikniemy czerwony sześciokąt.



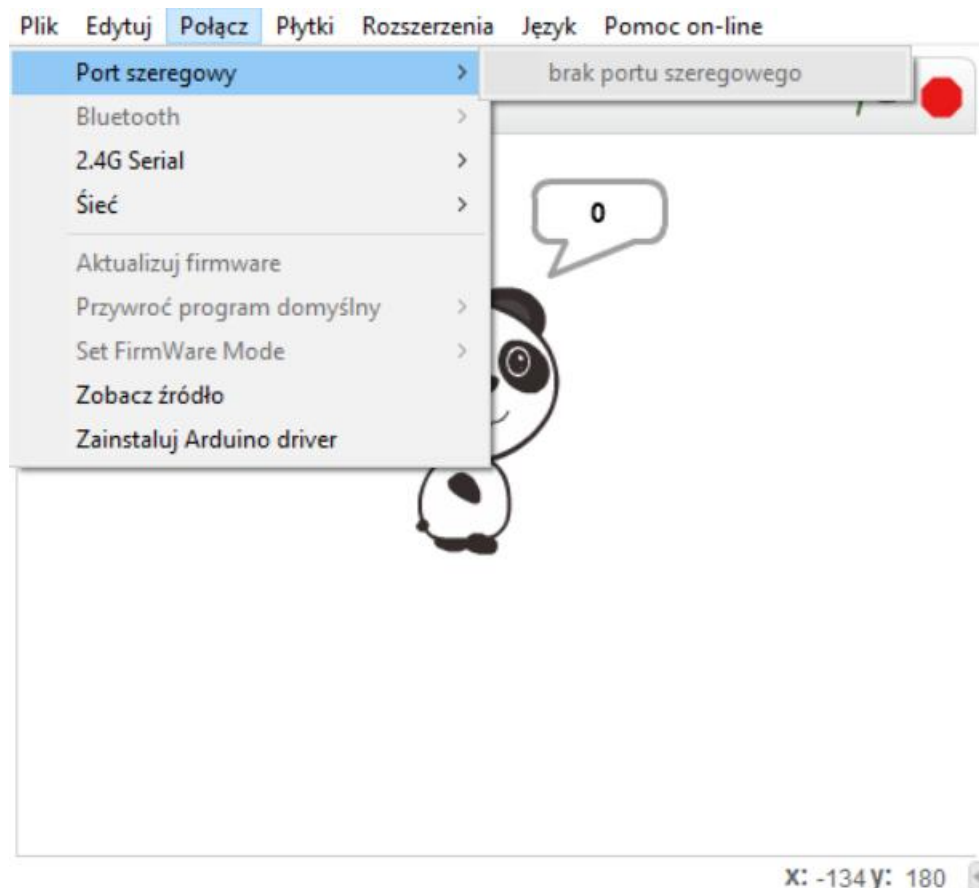
Prosimy uczniów o stworzenie programu składającego się z trzech omówionych wyżej bloków – na wykonanie tego zadania przeznaczamy 5 minut. Skrypt powinien wyglądać tak:



Podłączenie robota i kalibracja czujnika – 10 minut

Cel: powtórzenie procesu podłączania robota oraz kalibrowanie czujnika do dalszych zadań.

Uczniowie samodzielnie podłączają robota poprzez kabel USB oraz sprawdzają wartość „zero” czujnika natężenia światła w sali, w której odbywa się lekcja. Jeśli uczniowie wcześniej nie podłączali robota, krótko wyjaśniamy, jak to zrobić.



Robota podłączamy do komputera przez kabel USB, a następnie wybieramy w menu programu mBlock polecenia „Połącz” -> „Port szeregowy” -> Port pod którym podłączony jest robot. W wypadku, gdy do wyboru będzie więcej niż jeden port, należy odłączyć robota i sprawdzić który port zniknie. Następnie podłączyć go ponownie i wybrać ten port, który zniknął po odłączeniu robota. Gdy podłączymy robota poprawnie, na płytce – „mózgu” robota zaświeci się zielona dioda.

Po podłączeniu uczniowie uruchamiają stworzony program poprzez kliknięcie zielonej flagi i sprawdzają, jakie wartości pokazują się nad pandą. Prosimy o sprawdzenie wartości w różnych miejscach sali, o ile pozwala na to długość kabla. Na sprawdzenie wartości czujnika w różnych (różnie oświetlonych) miejscach przeznaczamy 5 minut. Można zasłonić górną część robota i sprawdzić, jaki będzie odczyt. Można też sprawdzić odczyt pod ławką i przy oświetleniu robota latarką lub telefonem.

Na zakończenie tej części, każdy z zespołów pracujących z robotem zapisuje na kartce wartość czujnika, gdy robot stoi na biurku przed monitorem. Będzie to nasza wartość wyjściowa do kolejnych zadań, wartość „zero”. Uczniowie mogą porównać swoje wartości między sobą i spróbować je skomentować (np. jedna grupa siedzi blisko okna, druga po słabiej oświetlonej stronie sali).

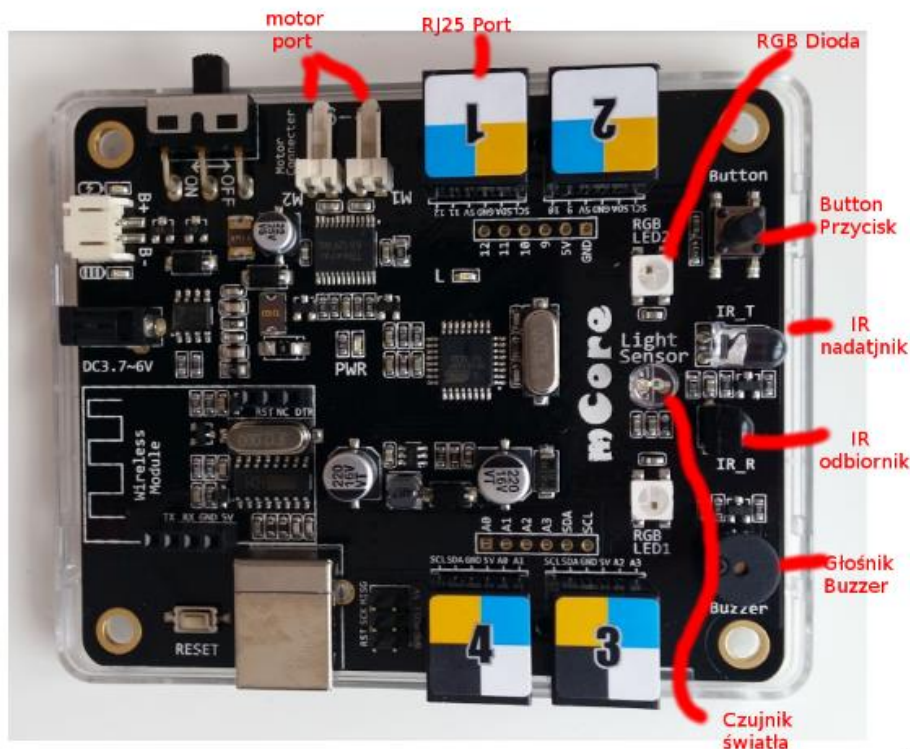
W tym miejscu możliwa jest przerwa (kolejna część scenariusza będzie realizowana na następnej lekcji).

Część 2. – 45 minut

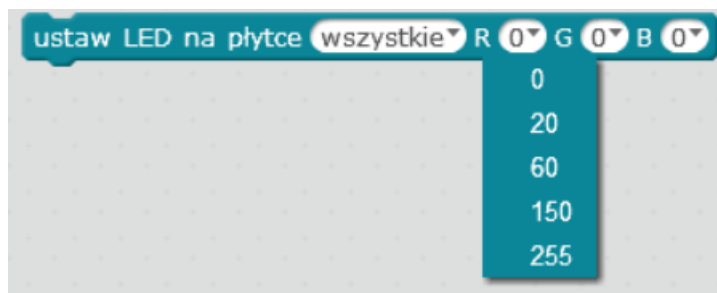
Przypomnienie zasad działania diody RGB LED – 5 minut

Cel: odświeżenie wiedzy na temat diod LED RGB.

Rozpoczynamy lekcję od krótkiego przypomnienia materiału z poprzednich lekcji dotyczącego działania diody RGB LED umieszczonej na płytce.



Na płytce robota umieszczone są dwie diody RGB LED. Mogą one świecić w różnych kolorach. Kolor diody uzyskujemy poprzez ustawienie odpowiednich „natężeń” 3 kolorów: czerwonego (R), zielonego (G) i niebieskiego (B). Kolory ustawiamy za pomocą bloku „ustaw LED na płytce... R... G... B...” z kategorii „Roboty”. Blok ten pozwala na ustawienie na lewej/prawej diodzie lub obydwu diodach koloru składającego się z trzech składowych, których natężenie wybieramy z zakresu od 0 do 255. Będziemy z nich korzystali „współpracując” z czujnikiem natężenia światła.



Program włączający diody, gdy robot znajduje się w ciemności – 20 minut.

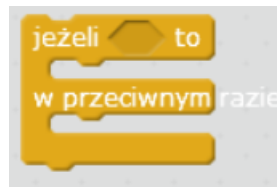
Cel: wykorzystanie operacji matematycznych i instrukcji warunkowej do realizacji zadania.

Zadaniem uczniów jest stworzenie programu, który zaświeci diody, jeśli robot znajdzie się w ciemniejszym miejscu niż biurko, na którym ustawili wartość „zero”. Najpierw prosimy o narysowanie takiego skryptu na tablicy, słuchamy propozycji uczniów, ewentualnie naprowadzamy ich na wykorzystanie instrukcji warunkowej – bloku „Jeżeli..., to..., w przeciwnym razie...”. Program powinien wyglądać następująco:
Start programu -> jeżeli wartość czujnika jest mniejsza od wartości „zero”, zapal dwie diody, w przeciwnym wypadku diody są zgaszone.

Aby wykonać to zadanie, trzeba wykorzystać blok umożliwiający wykonanie działań matematycznych, w tym przypadku porównywania wartości. Bloki pozwalające na działania matematyczne znajdziemy w kategorii „Wyrażenia”. Znajdują się tam bloki umożliwiające dodawanie/odejmowanie, porównywanie wartości, przeprowadzanie operacji logicznych typu „i”, „lub” itp. W naszym zadaniu potrzebny jest blok, który sprawdzi, czy wartość czujnika jest mniejsza od naszej wartości „zero”.



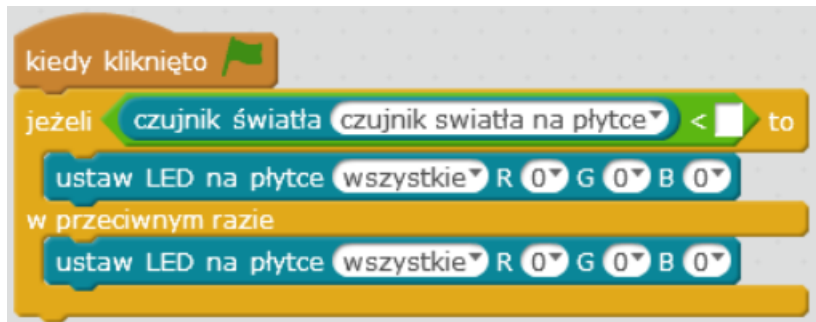
Klikając prawym przyciskiem myszki na ten blok, mamy możliwość zmiany operacji którą wykonujemy. Potrzebny jest nam także blok wykorzystujący instrukcję warunkową, czyli „Jeżeli..., to..., w przeciwnym razie...” z kategorii „Kontrola”.



Instrukcja warunkowa, to część skryptu, w której wykonywany jest blok znajdujący się w pierwszym polu, jeśli zostanie spełniony określony warunek („jeżeli...”). Jeśli nie („w przeciwnym razie”), wykonywany jest blok znajdujący się w drugim polu.

Prosimy uczniów o stworzenie programu wykorzystującego omówione bloki – przeznaczamy na to 10 minut. Kolory diod mogą być dowolne, ważne jest dobranie takich wartości, aby diody nie świeciły się, jeśli warunek nie zostanie spełniony. Aby uzyskać taki efekt, należy ustawić wartości zerowe dla każdego koloru.

Po zakończeniu zadania uczniowie podłączają ponownie robota i sprawdzają działanie programu. Zespoły, które wykonają zadanie przed czasem, mają możliwość zabawy z kolorami diod.



(Puste pole to miejsce na wpisanie wartości „zero” uzyskanej podczas kalibracji.)

Ulepszenie działania programu – 25 minut

Cel: zadaniem uczniów jest przemyślenie sposobu ulepszenia działania programu oraz dodanie nowej operacji do wykonania w przypadku oświetlenia robota.

Pytamy uczniów, czy mają pomysł na ulepszenie działania programu. Jeśli któryś z zespołów przedstawi ciekawy pomysł, dajemy możliwość jego realizacji. Jeśli nie, dajemy zadanie: jak ulepszyć program poprzez zmniejszenie czułości oraz dodanie reakcji na dodatkowe oświetlenie. Możemy zapytać uczniów, czy mają pomysł na to, jak zmniejszyć czułość. Odpowiedź powinna proponować zmianę warunku wykonania instrukcji. Reakcję na dodatkowe oświetlenie można zapewnić poprzez skopiowanie (zduplikowanie) skryptu i zmianę warunków lub samodzielne dołożenie bloków.

Uczniowie mają 15 minut na realizację zadania. Po jego wykonaniu każdy z zespołów pokazuje działanie swojego robota innym grupom oraz przedstawia wprowadzone przez siebie modyfikacje. Przykładowe rozwiązania zadań:



W miejsce pustych pól powinny być wstawione zmodyfikowane wartości, w miejsce 0 przy dodawaniu powinny być wstawione nasze wartości „zero”. Kolory diod mogą być dowolne.

Podsumowanie - 5 minut

Zapowiadamy, że na kolejnych lekcjach uczniowie będą wykorzystywać pilota do sterowania robotem i poznawać elementy robota, które pomogą im stworzyć autonomiczny pojazd. Pozostały czas lekcji można poświęcić na pytania oraz zabawę z robotami. Uczniowie zapisują swoje skrypty i wyłączają robota.

MOŻLIWE MODYFIKACJE DLA KLAS I-III I IV-VI:

W klasach I-III można przeprowadzić lekcję w formie zabawy. Uczniowie bawią się robotami, następnie pokazujemy im skrypt, tłumaczymy działanie bloków oraz znaczenie parametrów. W części zadaniowej rozmawiamy z uczniami na temat sposobu wykonania zadań, realizujemy ich pomysły i tworząc program pokazujemy

działanie robota.

W klasach IV-VI można zachować odpowiednio uproszczoną część teoretyczną, a następnie pracować z programem bez parametrów. Uczniowie otrzymują zadanie, w którym dobierają parametry programu, tak aby wykonywał on narzucone zadania. Ułatwieniem może być wykorzystanie wcześniej przygotowanych szkieletów programów.

ZADANIE SPRAWDZAJĄCE UMIEJĘTNOŚCI ZDOBYTE PODCZAS LEKCJI:

Uczeń / uczennica, pracując samodzielnie albo w dwu- lub trzyosobowym zespole tworzy skrypt wykorzystujący właściwości czujnika światła. W wyniku działania skryptu robot reaguje na obniżenie natężenia światła oraz na zwiększenie natężenia światła zapalając lub gasząc diody w różnych kolorach.

PIGUŁKA WIEDZY I INSPIRACJI DLA OSÓB PROWADZĄCYCH:

Wykorzystywane kategorie bloków:

Zdarzenia: bloki z tej kategorii służą do programowania interakcji z użytkownikami – tworzenia skryptów, które reagują na określone działania użytkownika.

Roboty: bloki z tej kategorii służą do programowania interakcji z robotem – tworzenia skryptów, które umożliwiają sterowanie robotem i reakcję na zdarzenia oraz inicjowanie i kontrolę zdarzeń z udziałem poszczególnych elementów robota (np. czujników).

Kontrola: bloki z tej kategorii pozwalają sterować programem, na przykład dodawać do skryptu warunek, pętlę albo opóźniać wykonanie skryptu.

Wyrażenia: bloki z tej kategorii pozwalają wprowadzać do skryptu działania matematyczne lub wyrażenia logiczne.

Wygląd: bloki z tej kategorii umożliwiają graficzną interakcję z duszkami w lewym oknie programu. Pozwalają zmienić wygląd duszka i komunikować się z nim, np. wyświetlić jego wypowiedź.

Dodatkowe materiały - linki:

Nauka programowania w języku i środowisku Scratch – zasoby edukacyjne programu „Mistrzowie Kodowania”:

http://wiki.mistrzowiekodowania.pl/index.php?title=Strona_g%C5%82%C3%B3wna#Scratch

Nauka programowania w języku i środowisku Scratch – zasoby edukacyjne programu

„Koduj z Klasą”:

http://kodujzklasa.ceo.org.pl/sites/kodujzklasa.ceo.org.pl/files/scenariusze_scratch.zip

Nauka programowania w języku i środowisku Scratch – zasoby edukacyjne programu „Link do Przyszłości”:

www.linkdoprzyszlosci.pl/zasoby

Nauka programowania w języku i środowisku Scratch – zasoby edukacyjne programu „#SuperKoderzy”:

<http://superkoderzy.pl/scenariusze-lekcji/podstawy-scratcha/>

Nauka programowania w środowisku mBlock z wykorzystaniem robota mBot - zasoby edukacyjne programu „Mistrzowie Kodowania”:

[http://wiki.mistrzowiekodowania.pl/index.php?title=Pierwsze kroki -
zdalne sterowanie robotem](http://wiki.mistrzowiekodowania.pl/index.php?title=Pierwsze_kroki_-_zdalne_sterowanie_robotem)

Czujnik fotoelektryczny (artykuł z Wikipedii):

https://pl.wikipedia.org/wiki/Czujnik_fotoelektryczny

Scenariusz został opracowany na potrzeby projektu „MoboLab – roboty i tablety w Twojej szkole”. Celem projektu jest zwiększenie kompetencji informatycznych z zakresu programowania i wykorzystywania technologii mobilnych w uczeniu się, a także kreatywności, innowacyjności i umiejętności współpracy w zespole z wykorzystaniem TIK, uczniów/uczennic z (UCZ) z 6 szkół podnadgimnazjalnych i 4 gimnazjów Wołomina i Zielonki. Projekt dofinansowany jest ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego (Regionalny Program Operacyjny Województwa Mazowieckiego na lata 2014-2020, Oś Priorytetowa X. Edukacja dla rozwoju regionu, Działanie 10.1. Edukacja ogólna i przedszkolna, Poddziałanie 10.1.2. Edukacja ogólna w ramach ZIT).



Ten utwór jest dostępny na licencji [Creative Commons Uznanie autorstwa 4.0 Międzynarodowe](https://creativecommons.org/licenses/by/4.0/).