

Scratch, matematyka i klocki LEGO

Agnieszka BOROWIECKA

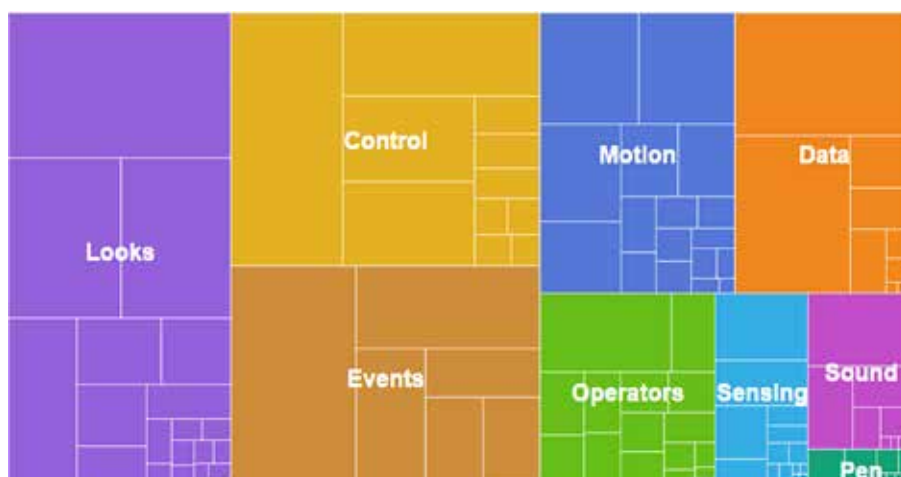
Wprowadzając uczniów w świat programowania, zwykle zaczynamy od środowiska wizualnego. Składanie programów z gotowych bloczków przyspiesza pracę szczególnie młodszych dzieci, mało jeszcze sprawnych w pisaniu i czytaniu. Bloczki łatwo ze sobą łączyć, zmieniać ich kolejność, zastępować innymi bez konieczności uczenia się skomplikowanej składni poleceń czy wpisywania ich z klawiatury.

Najbardziej popularnym środowiskiem do nauki programowania wizualnego jest Scratch. Został on zaprojektowany przez Mitchela Resnicka, twórcę języka StarLogo i pomysłodawcę serii zabawek Lego Mindstorms. Obecnie Scratch jest rozwijany przez grupę Lifelong Kindergarten z MIT Media Lab.

Scratch nie jest jedynym środowiskiem umożliwiającym budowanie programów z bloczków, ale inne podobne środowiska są zwykle mniej atrakcyjne wizualnie lub mają mniejsze możliwości.

Scratch i statystyki

We wrześniu 2017 roku społeczność Scratcha liczyła ponad dwadzieścia milionów zarejestrowanych użytkowników. Większość osób podczas zakładania konta deklaruje wiek poniżej 20 lat, najwięcej jest jedenasto- (prawie dwa miliony), dwunasto- (dwa miliony sto tysięcy) i trzynastolatków (prawie dwa miliony). Na platformie <https://scratch.mit.edu> można znaleźć ponad dwadzieścia cztery miliony udostępnionych projektów. Dużo z nich utworzyli samodzielnie członkowie społeczności Scratcha, ale wiele to wynik współpracy lub modyfikacji projektów udostępnionych przez innych (tzw. remiksy). Najczęściej wykorzystywane w projektach bloczki należą do kategorii *Wygląd* (22%), *Zdarzenia i Kontrola* (po 19%). Najrzadziej używane są bloczki z kategorii *Dźwięk* (3%) oraz *Pisak* (1%).



Rysunek 1. Wykorzystanie bloczków w projektach, strona ze statystykami Scratch.

Według założeń twórców Scratch miał być środowiskiem do nauki programowania, tymczasem większość przygotowanych w tym środowisku projektów to bardzo proste animacje lub gry, w których elementy programistyczne stosowane są w minimalnym zakresie. Może to wynikać ze struktury wiekowej użytkowników Scratcha lub z tego, że wiele pojęć, takich jak np. elementy logiki lub losowość, jest trudnych do samodzielnego zrozumienia bez wsparcia nauczyciela. Jednak mimo ograniczeń języka Scratch można przy jego użyciu tworzyć ciekawe i wartościowe projekty, rozwiązując konkretne problemy z różnych dziedzin, m.in. z matematyki.

Matematyka ze Scratchem

W środowisku Scratch można programować już w klasach I-III szkoły podstawowej. Warto zacząć zabawę od wersji dostępnej na urządzenia mobilne – Scratcha Juniora. Z jego pomocą uczniowie tworzą różne historyjki i animacje. Projekty mogą składać się z kilku scen i wielu duszków, dostępne bloczki podzielone są na sześć grup i nie wymagają od uczniów umiejętności czytania. Tworząc projekty w „dorosłej” wersji Scratcha, warto pamiętać, że możemy wykorzystać go na kilka sposobów. Prostsze projekty można tworzyć w całości razem z uczniami, przechodząc stopniowo do coraz trudniejszych konstrukcji i pojęć. Bardziej rozbudowane projekty możemy przygotować w formie szablonu zawierającego częściowe rozwiązanie problemu. Uczniowie uzupełniają brakujące elementy, dodają własną grafikę, testują działanie skryptów. Możliwe

jest także przygotowanie projektów w całości przez nauczyciela, rola uczniów sprowadza się wtedy tylko do badania działania projektu lub poznawania w atrakcyjny sposób ważnych pojęć. Młodzi uczniowie mogą na przykład porównywać ze sobą przedmioty lub liczby, sprawdzać umiejętność wykonywania w pamięci różnych działań, rozpoznawać figury geometryczne itp. Inną propozycją jest wyszukanie przydatnych projektów na stronie <https://scratch.mit.edu> i po ewentualnej korekcie wykorzystanie ich w swojej pracy. Spróbujmy przyjrzeć się bliżej kilku przykładowym projektom łączącym matematykę i programowanie w środowisku Scratch.

Tabliczka mnożenia

Jednym z pierwszych projektów, jaki realizujemy z uczniami, jest zwykle dialog między postaciami. Może być to fragment wiersza, np. „Zoo” Jana Brzechwy, rozmowa o patronie szkoły lub wydarzeniach historycznych. Podobny schemat możemy wykorzystać do odpytywania z działań na liczbach: dodawania, odejmowania i tabliczki mnożenia. W najprostszej wersji projektu wystarczy dodać tło i jednego duszka, który będzie zadawał pytanie. Po udzieleniu odpowiedzi wyświetlany będzie odpowiedni komentarz, np. „Brawo” lub „Niestety nie, spróbuj ponownie”. Liczby, na których wykonywane jest działanie, losujemy z podanego zakresu i zapamiętujemy w zmiennych. Możemy także sprawdzać umiejętność mnożenia przez konkretną wartość, wtedy losowana jest tylko jedna liczba.



Rysunek 2. Sprawdzamy umiejętność mnożenia

By zabawa była ciekawsza, można dodać drugiego duszka. Jego zadaniem będzie wyświetlenie liczby podanej przez użytkownika. Pierwszy z duszków zadaje pytanie i po odpowiednim czasie wyświetla gratulacje lub zachęca do dalszej nauki.



Rysunek 3. Wyświetlenie odpowiedzi przez duszka.

Najtrudniejszym elementem dla uczniów może okazać się wyświetlenie pytania. By duszek wyświetlił całe zdanie typu „Ile to jest $4 * 5$?” musimy użyć aż 6 bloczków. Podobne problemy wystąpią, gdy będziemy budować złożone warunki lub działania matematyczne do wykonania przez duszka.



Rysunek 4. Przykładowy skrypt duszka zadającego pytanie.

Interesującym zadaniem może być taka modyfikacja projektu, by duszek wielokrotnie pytał się o wynik mnożenia bez konieczności ponownego uruchamiania projektu za pomocą zielonej flagi. Możemy także dodać zliczanie poprawnych odpowiedzi i w zależności od uzyskanych wyników wyświetlać podsumowanie znajomości tabliczki mnożenia przez ucznia.

Ustawiamy zegar

W projekcie **Zegar** podajemy dwie liczby określające czas pokazywany przez zegar analogowy. Tło sceny zawiera tarczę zegara z zaznaczoną podziałką godzinową. Dwa duszki reprezentują wskazówki: godzinową i minutową. Na początku obie wskazówki skierowane są do góry, ponieważ zegar nie został jeszcze ustawiony.



Rysunek 5. Podawanie godziny ustawianej na zegarze

Duszek-kot prosi o podanie kolejno dwóch liczb – godziny i minut. Podane liczby zostają zapamiętane w pomocniczych zmiennych *godzina* i *minuty*, wyświetlanych w lewym górnym rogu sceny. Zostaje wysłany komunikat do duszków z prośbą o ustawienie zegara.



Rysunek 6. Skrypt ustawiający wskazówkę godzinową.

Możemy przedyskutować z uczniami, w jaki sposób duszki-wskazówki powinny zachować się po otrzymaniu komunikatu. W najprostszej wersji projektu każda ze wskazówek reaguje na wartość odpowiedniej zmiennej, ustawiając kierunek lub obracając się o właściwy kąt.

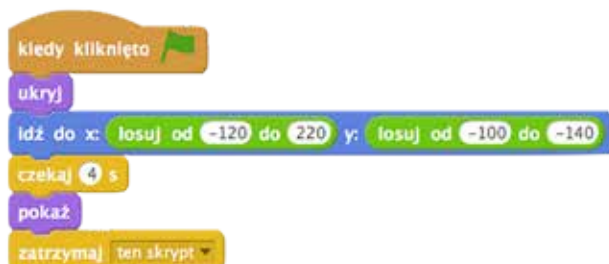


Rysunek 7. Zegar wyświetla podany czas

Projekt **Zegar** można rozszerzyć, poprawiając położenie wskazówki godzinowej. Powinna się ona nie tylko przesunąć skokowo co godzinę, ale także podczas danej godziny o pewien kąt, w zależności od tego, ile upłynęło minut. Inną modyfikacją może być odczytywanie przez ucznia czasu z tarczy zegara po wylosowaniu położenia wskazówek. Możemy także przygotować projekt, w którym wskazówki pokazują bieżący czas, korzystając z czujników *aktualna godzina* i *aktualna minuta* .

Rozpoznajemy figury

Bardzo uniwersalnym zadaniem jest podział duszków na dwie grupy lub więcej. Możemy na przykład rozpoznawać figury geometryczne, dzielić liczby na dodatnie i ujemne lub parzyste i nieparzyste. Ten sam projekt po niewielkiej modyfikacji można wykorzystać na lekcji języka polskiego (np. pisanie wyrazów z **rz** i **ź**), przyrody (podział na owoce i warzywa, drzewa liściaste i iglaste itp.) czy dowolnego innego przedmiotu. Do rozpoznania, czy przedmioty zostały prawidłowo podzielone na grupy, można użyć tła sceny.



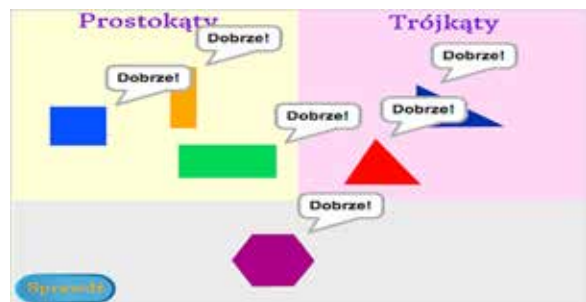
Rysunek 8. Losowanie położenia figury.

Po uruchomieniu projektu za pomocą zielonej flagi, duszki-figury rozsypują się w sposób losowy w dolnej części sceny. Analizujemy z uczniami skrypt zmieniający położenie duszków, zastanawiając się wspólnie nad zakresem losowanych współrzędnych. Uczniowie mogą samodzielnie przygotować tło sceny, wtedy odczytują za pomocą kursora myszki, jakie współrzędne mogą mieć duszki, by mieścić się w całości na szarym tle. Zauważamy, że jeśli prawidłowo dobraliśmy zakres losowania, to wszystkie duszki-figury mogą korzystać z tego samego skryptu.



Rysunek 9. Sprawdzenie, czy trójkąt został prawidłowo umieszczony na scenie.

Naszym zadaniem jest przesunięcie myszką figur do właściwej części sceny. Należy pamiętać, by włączyć opcję *można przeciągać w odtwarzaczu* dla każdego duszka-figury. Po naciśnięciu przycisku **Sprawdź**, zostaje nadany komunikat *sprawdź figurę*. Każdy duszek-figura reaguje na komunikat, sprawdzając, jakiego koloru tła dotyka. Następnie wypowiada odpowiednio komentarz „Dobrze” lub „Źle”, w zależności od tego, w jakiej części sceny się znalazł.



Rysunek 10. Wybieramy trójkąty i prostokąty.

Łatwo sprawdzić, że proponowane rozwiązanie nie jest idealne. Jeśli figurę umieścimy w taki sposób, że dotyka kilku kolorów (nie leży całkowicie wewnątrz właściwej części sceny), to również uzyskamy komentarz „Dobrze”. Warto spróbować z bardziej zaawansowanymi uczniami zmienić warunki sprawdzane przez duszki, by rozwiązać ten problem. Projekt można urozmaicić, dodając dodatkowe kostiumy dla duszków. Po kliknięciu w zieloną flagę będzie losowany kostium dla każdego duszka. Uczeń za każdym razem otrzyma nowy zestaw figur różniących się kształtem lub kolorem od poprzedniego.

Wielokrotności liczb

Przygotowując specjalne kostiumy dla duszków, możemy tworzyć ciekawe projekty i przyspieszać jednocześnie proces ich budowania. Jeden ze scenariuszy omawianych na zajęciach w ramach projektu **Warszawa programuje!** dotyczy układania wyrazu z poruszających się po ekranie literek. Aby można było automatycznie tworzyć różne wyrazy, niezbędne jest przygotowanie duszka-litera, którego kostiumy to kolejne litery alfabetu. W standardowej bibliotece kostiumów Scratcha dostępne są jedynie pojedyncze litery, nie ma także znaków diakrytycznych potrzebnych do tworzenia słów w języku polskim. Dlatego kostiumy dla duszka musimy wcześniej samodzielnie przygotować lub pobrać z internetu.

Do projektu **Wielokrotności liczb** warto najpierw przygotować duszka, którego kostiumami będą kolejne liczby od jeden do stu. Podobne duszki możemy później wykorzystać w innych projektach związanych z liczeniem.

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

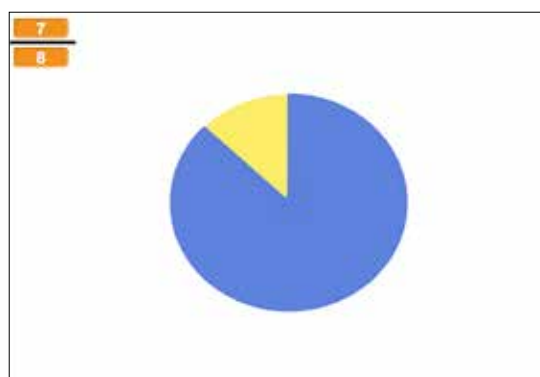
Rysunek 11. Wielokrotności liczby 3.

Dla wzorcowego duszka-liczby definiujemy zmienną prywatną *numer klona*, a następnie klonujemy go i ustawiamy na właściwej pozycji. Uczniowie mogą samodzielnie ustawić sto duszków, zaczynając od pojedynczego wiersza z dziesięcioma liczbami, a potem dodając drugą pętlę i tworząc kolejne klony. Wspólnie tworzymy skrypt określający zachowanie duszków – każdy z nich powinien mieć inną wartość zmiennej *numer klona* i inny kostium. Jeśli dany klon dotyka wskaźnika myszki, traktowany jest jako wybrany przez użytkownika. Wszystkie pozostałe duszki, o numerze będącym wielokrotnością wybranego, zmieniają swój rozmiar lub dodają efekt *kolor* albo *jasność*, aby się wyróżnić. Reszta klonów stosuje standardowe ustawienia.

Tablicę liczb wyświetlanych w projekcie można wykorzystać także na inne sposoby. Możemy na przykład wrócić do tabliczki mnożenia. Użytkownik podaje dwie liczby, a klony odpowiadające wynikowi mnożenia zostają wyróżnione. Starsi uczniowie mogą spróbować przygotować wizualizację wyszukiwania liczby pierwszą metodą sita Eratostenesa.

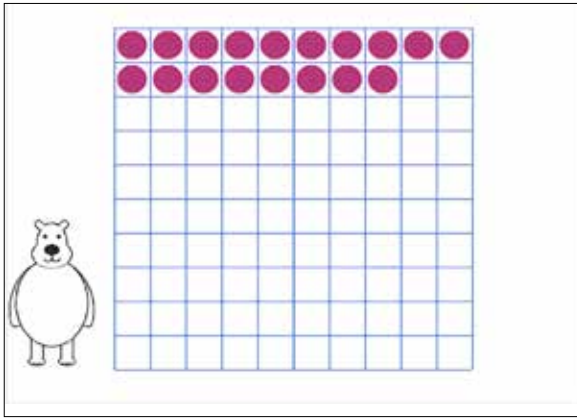
Ułamki

W Scratchu możemy przygotowywać skrypty pomagające uczniom rozumieć ułamki zwykłe i dziesiętne. Do wizualizacji ułamków zwykłych można wykorzystać bloczki z grupy *Pisak*. Po otrzymaniu informacji o liczniku i mianowniku ułamka duszek rysuje odcinki w dwóch kolorach. Niebieski kolor to nasz ułamek, żółty – dopełnienie do jedności. Możemy ilustrować ułamek za pomocą wycinka koła lub jako dwukolorowy prostokąt.



Rysunek 12. Ułamek 7/8.

Rysowania można także użyć do przedstawiania ułamków dziesiętnych. Przygotowujemy tło sceny zawierające siatkę kwadratów o wymiarach 10 na 10. Po podaniu wartości ułamka, duszek skacze do środka kolejnych kwadratów i w odpowiedniej części z nich rysuje dużą kropkę.



Rysunek 13. Ułamek 0.18

Oba przedstawione projekty można prosto rozbudować. Po dodaniu wyświetlania drugiego ułamka zwykłego możemy porównywać ułamki i wskazywać na przykład, który z nich jest większy. Siatki kwadratów z drugiego projektu można użyć do ilustrowania dodawania lub odejmowania ułamków dziesiętnych.

Cyfry i systemy liczbowe

Korzystając z operacji dzielenia możemy przygotować projekt, w którym duszek „czyta” kolejne cyfry podanej przez nas liczby. Potrzebna będzie zmienna *liczba* i bloczki *zapytaj* oraz *odpowiedz* do komunikacji z użytkownikiem. Wykorzystamy operację *mod* – reszta z dzielenia oraz część całkowitą z dzielenia. Niestety w środowisku Scratch nie ma operacji dzielenia całkowitego. Zamiast tego można skorzystać z funkcji *podłoga*, która dla danej liczby obcina jej część ułamkową, zostawiając największą liczbę całkowitą mniejszą lub równą podanej wartości. Warto zwrócić uwagę uczniów na kolejność wyświetlanych cyfr.

W przedstawionym na rysunku 14 skrypcie algorytm można także wykorzystać do przeliczania z systemu dziesiętnego na inny system liczbowy. Wystarczy zastąpić dzielenie przez dziesięć dzieleniem przez podstawę tego systemu. Zamiast kazać



Rysunek 14. Wyświetlenie kolejnych cyfr liczby.

duszkowi wyświetlać kolejne cyfry otrzymanej liczby, można je zapamiętać w liście albo wyświetlić na scenie jako kolejne duszki. Każdy z duszków będzie miał tyle kostiumów, ile jest różnych cyfr w systemie liczbowym, na który przeliczamy. Dla systemu binarnego będą to zero i jedynka, dla systemu trójkowego zero, jedynka i dwójka itd.



Rysunek 15. Przeliczenie na system dwójkowy.

Prostszym projektem dla uczniów może być przeliczanie z systemu dwójkowego na dziesiętny. Wystarczy, jeśli przygotują oni tyle duszków, ile maksymalnie cyfr może liczyć wyświetlana wartość binarna. Każdy duszek ma dwa kostiumy – zero i jedynkę. Po kliknięciu w duszka następuje przetączenie kostiumu. Z duszkami związane są zmienne odpowiadające potęgze liczby 2. Jeśli duszek pierwszy z prawej ma kostium z liczbą zero – to właściwa zmienna ma wartość zero, jeśli kostium z jedynką – to zmienna ma wartość jeden. Dla kolejnego duszka zero odpowiada zero, a jedynce dwójka. U duszków następnych są to zero i czwórka, zero i ósemka itd. Po każdej zmianie

kostiumu dowolnego z duszków liczona jest suma wszystkich zmiennych i przypisywana pomocniczej zmiennej *dziesiątka*.

Uczniowie mogą samodzielnie przygotować analogiczny projekt do przeliczania z systemu trójkowego na dziesiętny.

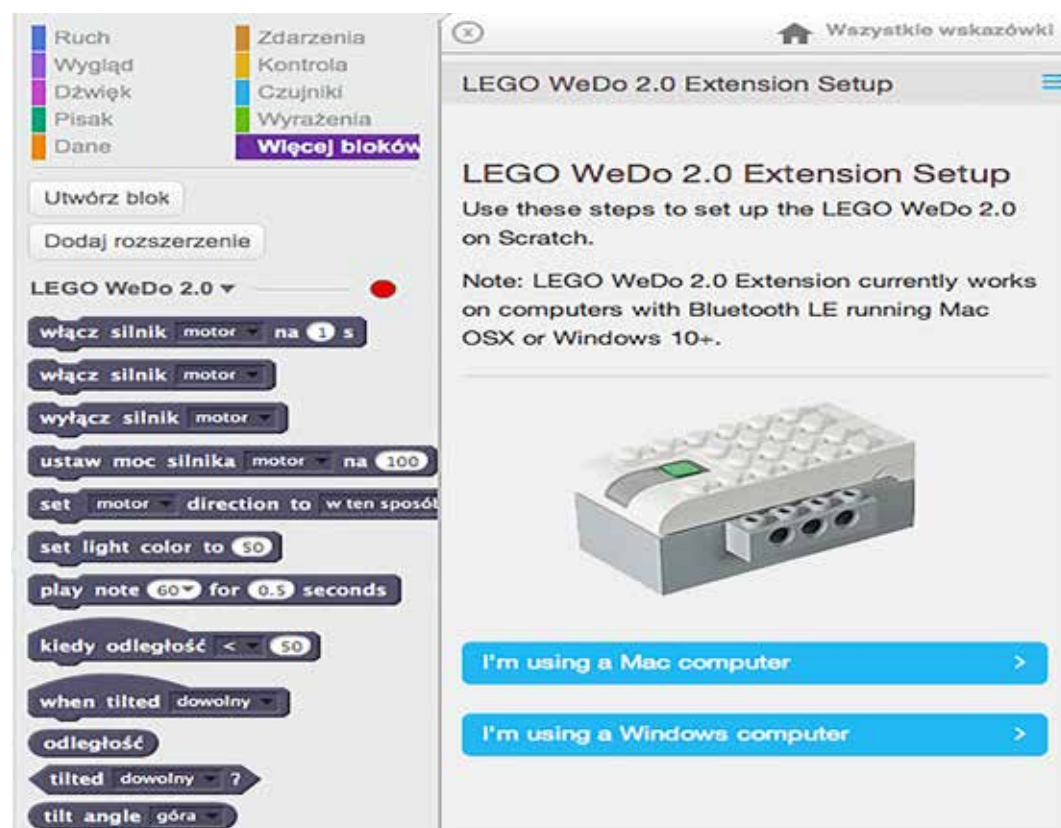
Zbuduj z klocków LEGO i baw się ze Scratchem

Każdy z nas w dzieciństwie bawił się klockami. Zapewne najbardziej atrakcyjne dla dzieci są klocki LEGO. Dzisiaj możemy połączyć zabawę z programowaniem, budując z klocków Lego WeDo różne konstrukcje, a następnie sterując nimi za pomocą oryginalnego oprogramowania lub specjalnego rozszerzenia dodawanego do Scratcha. Bazowy zestaw klocków WeDo 2.0 zawiera ponad 150 elementów, w tym silnik, dwa czujniki oraz jednostkę sterującą. Oprogramowanie można bezpłatnie zainstalować na tabletach, laptopach lub komputerach stacjonarnych. Jedynym wymaganiem jest wyposażenie sprzętu komputerowego w standard Bluetooth Low Energy. Jeśli nie mamy

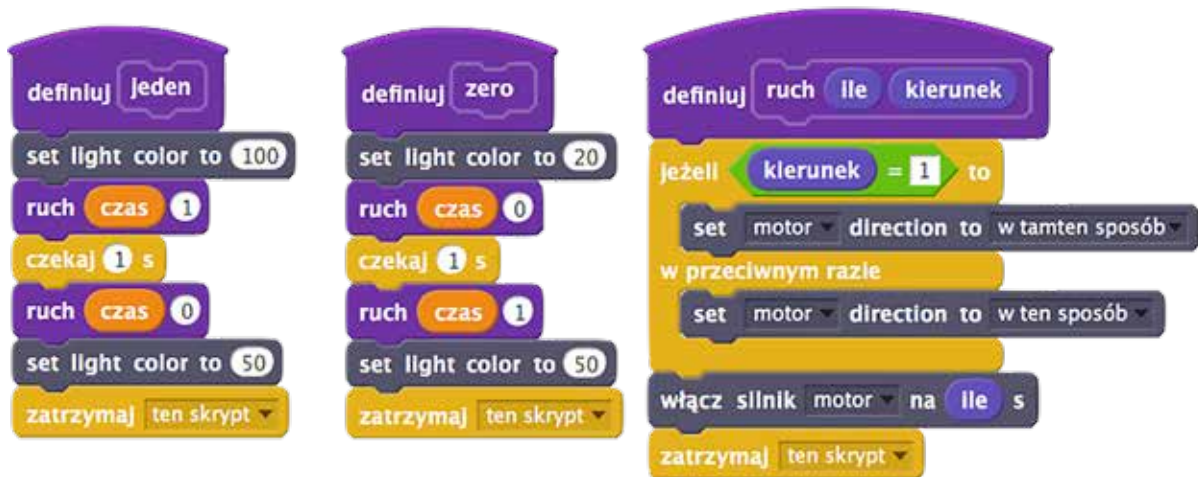
wbudowanego takiego interfejsu, możliwe jest podłączenie go zewnętrznym przez wejście USB.

Do programu sterującego robotami WeDo dołączonych jest ponad 20 gotowych scenariuszy składających się z wprowadzenia, opisu budowania robota krok po kroku oraz krótkiego programu sterującego zbudowaną konstrukcją. Programy budujemy z bloczków, podobnie jak w Scratchu Juniorze. W internecie można także znaleźć przykłady oryginalnych konstrukcji wymyślonych i zbudowanych przez uczniów.

W Scratchu istnieje możliwość dodania rozszerzeń pozwalających łączyć się z urządzeniami zewnętrznymi, takimi jak PicoBoard, LEGO WeDo 1.0 i LEGO WeDo 2.0. Wybieramy grupę *Więcej bloczków* i naciskamy przycisk **Dodaj rozszerzenie**. Jeśli chcemy kierować robotem WeDo, musimy jeszcze połączyć się z jego jednostką sterującą. Należy zainstalować na komputerze program Scratch Device Manager, uruchomić go i włączyć robota. Po wykryciu i sparowaniu robota z komputerem możemy sterować jego zachowaniem za pomocą skryptów tworzonych w Scratchu.



Rysunek 16. Bloczki sterujące robotem WeDo i instrukcja podłączenia go do Scratcha.



Rysunek 17. Przykładowe skrypty sterujące robotem Milo.

Ponieważ zestaw WeDo zawiera tylko jeden silnik, roboty mają ograniczoną możliwość ruchu – mogą jeździć tylko do przodu i do tyłu, nie potrafią jednak skręcać. Istnieje możliwość dokupienia i podłączenia dodatkowego silniczka, co pozwoli zwiększyć możliwości sterowania robotem. Jednak już nawet standardowa konstrukcja zalecana do zbudowania jako pierwsza, tzw. robot Milo, może być wykorzystana do tworzenia ciekawych projektów. Wystarczy przygotować dwie plansze, po których będzie poruszał się Milo. Na jednej drukujemy lub piszemy cyfrę zero, na drugiej zaś jedynekę. Otwieramy w Scratchu projekt przeliczający liczby z systemu dziesiętnego na dwójkowy i zapamiętujący wynik w postaci listy. Dodajemy do niego skrypty sterujące robotem: po odczytaniu z listy cyfry zero Milo jedzie na planszę z cyfrą zero, a następnie wraca do wyjściowej pozycji. Dla cyfry jeden Milo jedzie w przeciwną stronę. Dodatkowo możemy zmieniać kolor światełka na jednostce sterującej.

Podsumowanie

Scratch jest ciekawym środowiskiem i warto z niego korzystać na lekcji. Uczniowie łatwo uczą się budować z bloczków skrypty, wymyślają różne historyjki, projektują scenę i kostiumy dla duszków. Należy zwrócić uwagę na to, jakie projekty będziemy tworzyć. Już samo sterowanie duszkami wprowadza nas w świat matematyki – ustalanie współrzędnych duszka, wyliczanie kątów obrotu, budowanie warunków logicznych. W swoich poszukiwaniach warto posunąć się jednak dalej i przygotować bardziej rozbudowane projekty matematyczne. Jeśli możemy dodatkowo połączyć się ze światem realnym, chociażby poprzez sterowanie robotami, nauka programowania stanie się ulubionym zajęciem naszych uczniów, a matematyka przestanie być strasznym i stresującym przedmiotem.

Agnieszka BOROWIECKA jest nauczycielem konsultantem w Ośrodku Edukacji Informatycznej i Zastosowań Komputerów w Warszawie.