

Robo-matematycy

MATEMATYKA

W tej ścieżce tematycznej zaprosimy #SuperKoderów do stworzenia gry planszowej (w rozmiarze maxi), w której zakodujemy zadania matematyczne odpowiednie dla każdego poziomu edukacyjnego. Lekcje zawierają elementy grywalizacji – uczniowie będą pracować w grupach i przygotowywać dla siebie nawzajem zadania. W grach wezmą udział także roboty... w charakterze sędziów rozstrzygających, która z grup wygra konkurencję!

Autorki:

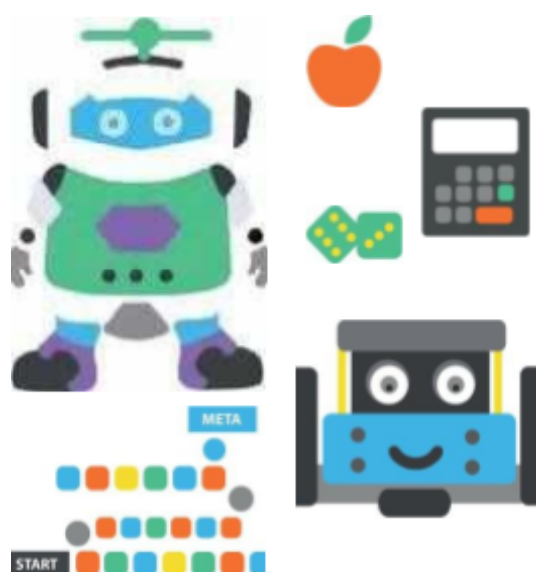
Joanna Płatkowska-Nęcka, Karolina Czerwińska

Spis lekcji:

- Lekcja 1:** Hello World! Pierwsza linijka kodu dla mBota2.
- Lekcja 2:** Silniki, głośniki i LEDy, czyli poznajemy się bliżej z mBotem2.
- Lekcja 3:** Programowanie - sposób na łatwe obliczenia.
- Lekcja 4:** Ułamki dziesiętne, procenty oraz... mBot2-matematyk.
- Lekcja 5:** Tworzymy grę matematyczną – część 1
- Lekcja 6:** Tworzymy grę matematyczną – część 2
- Lekcja 7:** Tworzymy grę matematyczną – część 3
- Lekcja 8:** Wielki turniej matematyczny – gramy!

Licencja:

Creative Commons Uznanie autorstwa Na tych samych warunkach 3.0



Robo-matematycy

Autorki: Joanna Płatkowska, Karolina Czerwińska

Lekcja 1:

Hello World! Pierwsze linijki kodu dla mBot2

Lekcja, podczas której uczniowie poznają mBot2: jego budowę i podstawowe funkcjonalności, piszą pierwszy program zarządzający ruchem robota. Uczniowie zapoznają się także z programem ścieżki i dowiedzą, nad jakim projektem będą pracować.

Cele zajęć:

Uczeń powinien:

- wyjaśniać, na czym będzie polegał #SuperKoderski projekt,
- opisywać, czym jest mBot2,
- wymieniać podstawowe elementy mBot2 i jego funkcje,
- stworzyć prosty program kontrolujący mBot przy użyciu aplikacji mBlock.

Materiały pomocnicze:

- roboty mBot2 (dla poprzedniej wersji robotów mBot należy skorzystać z archiwalnej wersji tych scenariuszy, dostępnych pod adresem https://superkoderzy.pl/wp-content/uploads/2018/05/Robo-matematycy_fv_2020-2.pdf);
- program mBlock (do kodowania robota) w wersji online <https://ide.mblock.cc/> lub offline <http://www.mblock.cc/download/>;
- komputery stacjonarne lub laptopy (ze sprawnym portem USB);
- laptop lub komputer nauczyciela podłączony do tablicy multimedialnej lub projektora.

Pojęcia kluczowe:

→ robot → silnik → czujnik → mikrokontroler CyberPi
→ mBlock

Czas na realizację zajęć: 45 minut (1 godzina lekcyjna)

Metody pracy:

- grupowa – praca w grupach z robotem,
- ćwiczenia praktyczne,
- pogadanka.

Podstawa programowa kształcenia ogólnego dla szkół

podstawowych – II etap edukacyjny – klasy IV-VI, informatyka:

I. Rozumienie, analizowanie i rozwiązywanie problemów. Uczeń:

2. formułuje i zapisuje w postaci algorytmów polecenia składające się na:

3) sterowanie robotem lub obiektem na ekranie;

3. w algorytmicznym rozwiązywaniu problemu wyróżnia podstawowe kroki: określenie problemu i celu do osiągnięcia, analiza sytuacji problemowej, opracowanie rozwiązania, sprawdzenie rozwiązania problemu dla przykładowych danych, zapisanie rozwiązania w postaci schematu lub programu.

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych. Uczeń:

1. projektuje, tworzy i zapisuje w wizualnym języku programowania:

1) pomysły historyjek i rozwiązania problemów, w tym proste algorytmy z wykorzystaniem poleceń sekwencyjnych, warunkowych i iteracyjnych oraz zdarzeń,

2) prosty program sterujący robotem lub innym obiektem na ekranie komputera;

2. testuje na komputerze swoje programy pod względem zgodności z przyjętymi założeniami i ewentualnie je poprawia, objaśnia przebieg działania programów;

IV. Rozwijanie kompetencji społecznych. Uczeń:

1. uczestniczy w zespołowym rozwiązaniu problemu posługując się technologią taką jak: poczta elektroniczna, forum, wirtualne środowisko kształcenia, dedykowany portal edukacyjny;

2. identyfikuje i docenia korzyści płynące ze współpracy nad wspólnym rozwiązywaniem problemów;

V. Przestrzeganie prawa i zasad bezpieczeństwa. Uczeń:

1. posługuje się technologią zgodnie z przyjętymi zasadami i prawem; przestrzega zasad bezpieczeństwa i higieny pracy;

Przygotowania przed zajęciami:

1. **Złożenie robota.** W scenariuszach nie przewidujemy osobnej lekcji poświęconej na składanie robotów. Nauczyciel może sam przygotować roboty przed rozpoczęciem realizacji ścieżki przedmiotowej, może także poprosić o pomoc uczniów lub zaproponować przygotowanie robotów np. w ramach lekcji techniki lub informatyki. Obrazkowa instrukcja złożenia robota dostępna jest w materiałach producentach **TUTAJ**.
2. **Oprogramowanie.** Przed zajęciami należy także zainstalować na komputerach aplikację mBlock, stworzoną na bazie środowiska Scratch, która będzie nam służyła do programowania robota. Link do pobrania programu: <http://www.mblock.cc/download/>.
3. **Organizacja zajęć.** Realizacja projektu wymaga pracy w grupach, optymalnie 2- lub 3-osobowych. Do każdej z grup przypisujemy jednego mBot2 - podział na grupy wynika więc z liczby robotów, którymi dysponujemy.
4. **Edukacja językowa.** Naszym założeniem jest, by uczniowie w praktyczny sposób uczyli się języka angielskiego: wykorzystując go w środowisku mBlock (nie programujemy w języku polskim!), nagrywając dialogi i je zapisując.

Przebieg zajęć:

1. Wprowadzenie w tematykę i integracja grupy

Nauczyciel omawia z uczniami plan na kolejne #SuperKoderskie lekcje. Ich celem będzie przygotowanie gry, w której zawodnicy będą rozwiązywać zadania matematyczne, a robot będzie poruszał się po planszy między kolejnymi polami oznaczającymi kolejne zadania (robot może również poruszać się między stacjami poukładanymi na podłodze). Robot będzie również mierzył czas, w jakim zawodnik lub zawodnicy ukończą daną rozgrywkę.

Nauczyciel z uczniami zastanawia się jakiego rodzaju zadania matematyczne mogłyby pojawić się w takiej grze.

2. Część zasadnicza

Zadanie 1: budowa robota

Nauczyciel prezentuje mBota2 i rozdaje roboty uczniom. Nim je włączymy i podłączymy do komputerów, konieczne jest omówienie jego budowy.

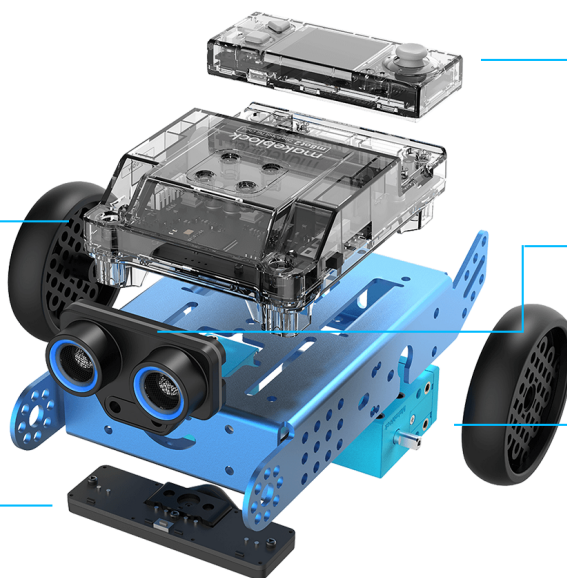
Nauczyciel wymienia kolejne nazwy części robotów i zachęca uczniów do wskazania, gdzie się one znajdują. Jeśli uczniowie nie potrafią od razu wskazać wymienionego elementu, można opisać zasadę działania danej części robota - ważne, by uczniowie sami ją "odkryli"!

mBot2 sterownik

Zawiera porty do podłączania modułów zewnętrznych i silników oraz zintegrowany akumulator ładowany przez USB

4 czujniki linii RGB

Nowy, poczwórny czujnik z rozpoznawaniem kolorów wspiera zaawansowane techniki śledzenia linii.



CyberPi

Mikroprocesor ESP32 z komunikacją bezprzewodową WiFi i Bluetooth, Zintegrowane czujniki i wyświetlacz. Programowany blokowo i w Python. Może być używany samodzielnie po odłączeniu od robota.

Nowy czujnik odległości

Pomiar odległości, wykrywanie przeszkód i sygnalizacja przy pomocy 8 diod LED

Silniki z enkoderami

Nowe silniki z czujnikami obrotu pozwalającymi na pomiar przejechanego dystansu, kontrolę prędkości do 200RPM i położenia z dokładnością do 1 stopnia.

Warto porównać mBot2 do człowieka. “Mózgiem” mBot2 jest mikrokontroler CyberPi, “oczami” jest czujnik ultrasoniczny, “ciałem” sterownik Mbot2, a “nogami” podwozie z kołami. Dodatkowo mBot2 może komunikować się odtwarzając dźwięki, wyświetlając kolory na listwie LED oraz komunikaty czy grafiki na wyświetlaczu. Podobnie jak u człowieka, “mózg” robota może działać tylko, jeśli połączony jest z resztą “ciała” (podwozie) lub bezpośrednio ze źródłem zasilania (podłączony kablem do komputera lub bezpośrednio do kontaktu). “Ciało i nogi” robota (sterownik mBot2 i podwozie) nie są w stanie się poruszać bez “mózgu” (mikrokontrolera).

Jeśli w szkole są dostępne zarówno mBot2, jak i mBoty w starszej wersji konieczne będzie pokazanie uczniom różnic pomiędzy tymi wersjami. Porównanie tych wersji można znaleźć w Załączniku nr 2.

Załącznik nr 2

	mBot 1.1, mBot-S Explorer Kit	mBot2
Sterownik	mCore	CyberPi
Liczba programów w pamięci	1	8
Komunikacja bezprzewodowa	Bluetooth lub 2,4G, IR	Bluetooth, Wi-Fi oraz Wi-Fi LAN
Pilot	tak	nie
Urządzenia wejścia i czujniki zintegrowane	Przycisk, czujnik światła	Joystick, przycisk x2, czujnik światła, mikrofon, żyroskop – akcelerometr
Urządzenia wyjścia	Buzzer (odtwarzający pojedyncze dźwięki/nuty o różnej wysokości), RGB-LED x2, wyświetlacz LED 16x8px mono (Explorer Kit)	Głośnik, RGB LED x5, wyświetlacz kolorowy IPS 128x128px
Języki programowania	mBlock/Scratch, ArduinoC	mBlock/Scratch, microPython, Python3
Moc silnika	zależna od poziomu naładowania baterii = im mniej naładowana bateria, tylko wolniej jedzie robot przy tym samym programie	niezależna od poziomu baterii = robot jedzie w takim samym tempie i pokonuje taką samą trasę niezależnie od poziomu naładowania baterii

Pełną instrukcję i grafiki pokazujące poszczególne elementy mBota2 można znaleźć [TUTAJ](#) oraz [TUTAJ](#).

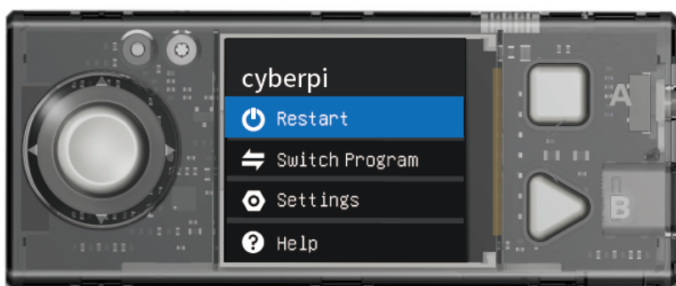
Zadanie 2: co mBot potrafi?

Na początek nauczyciel wyjaśnia uczniom, że mBot2 może pracować w dwóch trybach: live lub upload.

Tryb live polega na ciągłym i bezpośrednim połączeniu robota z komputerem (przy użyciu kabla lub Bluetooth Dongle). Robot wykonuje wszystkie polecenia pisane i na bieżąco zmieniane z poziomu komputera. W tym trybie robot może komunikować się z postaciami na ekranie komputera. Ten tryb będzie wykorzystywany w naszych scenariuszach.

Tryb upload polega na napisaniu na komputerze, a następnie przesłaniu (wgraniu) programu robotowi. Jeśli chcemy wprowadzić zmiany do programu, musimy po zrobieniu niezbędnych poprawek, ponownie wgrać go robotowi. Takie programy działają zdecydowanie szybciej i tryb ten jest idealny do bardzo skomplikowanych, wymagających programów, ale równocześnie znacznie trudniej jest sprawdzać wprowadzane na bieżąco zmiany. Każdy robot w momencie pierwszego uruchomienia ma standardowo wgrane kilka programów, które warto pokazać uczniom. Nie są one wykorzystywane w późniejszych scenariuszach, jednak mogą one pobudzić ciekawość uczniów i chęć włączenia się w programowanie.

Po włączeniu robota suwakiem znajdującym się z boku obudowy pojawia się ekran startowy:



Po wybraniu joystickiem opcji Switch Program możemy wybrać spośród ośmiu wgranych programów. Warto spróbować z uczniami nagrywania i odtwarzania (Magical Recorder), unikania przeszkód (mBot2_demo 2) czy programowania ścieżki jazdy robota (mBot2_demo 3). Instrukcje do każdego programu pojawiają się na wyświetlaczu. Żeby wrócić do głównego menu i zmienić program, należy nacisnąć znajdujący się po prawej stronie guzik Home.

Zadanie 3: łączenie mBota2 z komputerem

Po przeprowadzonych testach nauczyciel zaprasza uczniów do napisania pierwszych programów komputerowych z wykorzystaniem mBota2. Uczniowie siadają do komputerów i łączą roboty z komputerami.

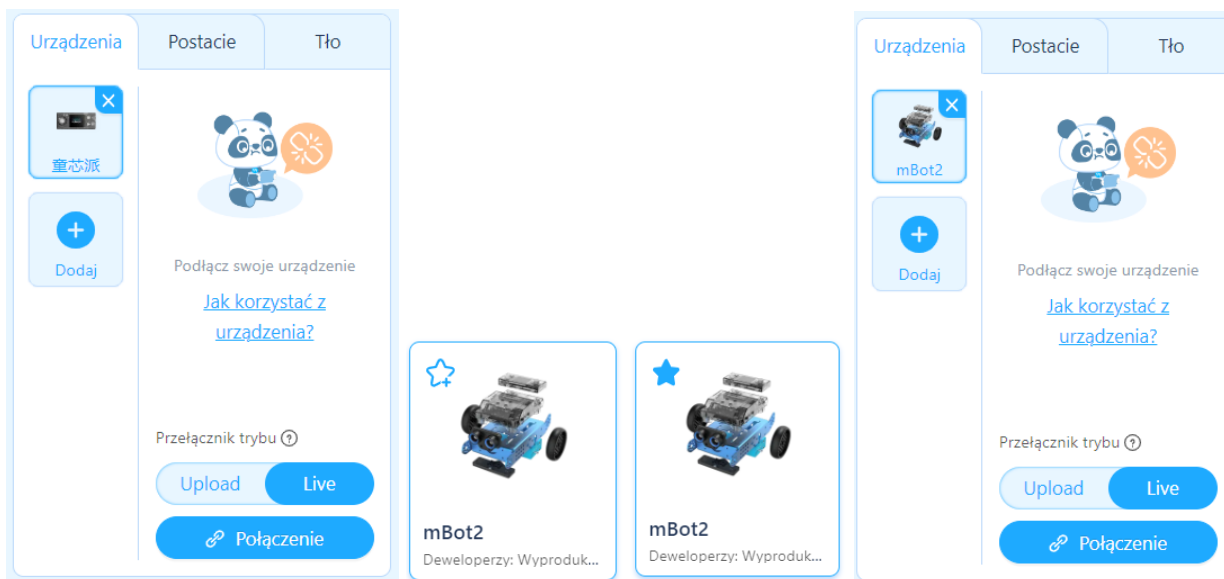
Warto tu zauważyć, że można korzystać z programu mBlock do pisania programu dla robota w wersji online lub offline. Nauczyciel wybiera oczywiście wersję bardziej pasującą do kontekstu szkoły i wydajności łącza internetowego. Przewagą wersji online jest to, że jest ona aktualizowana szybciej i sprawniej niż wersja offline, a także mBot2 jest "gotowy do użycia" w tej wersji. Przewagą wersji offline jest to, że nie jest ona zależna od stabilności łącza internetowego, a także, że pozwala bezpośrednio sprawdzić czas trwania nagrań wykorzystywanych w programie, co może w znaczący sposób przyspieszyć pracę nad projektem. Równocześnie w tej wersji mBota2 trzeba w dwóch prostych, szybkich krokach przygotować do programowania (kroki te pokazane są poniżej w punkcie 4). W przypadku wyboru wersji online konieczne jest jednorazowe pobranie i zainstalowanie programu mLink:

<https://mblock.makeblock.com/en-us/download/mlink/>

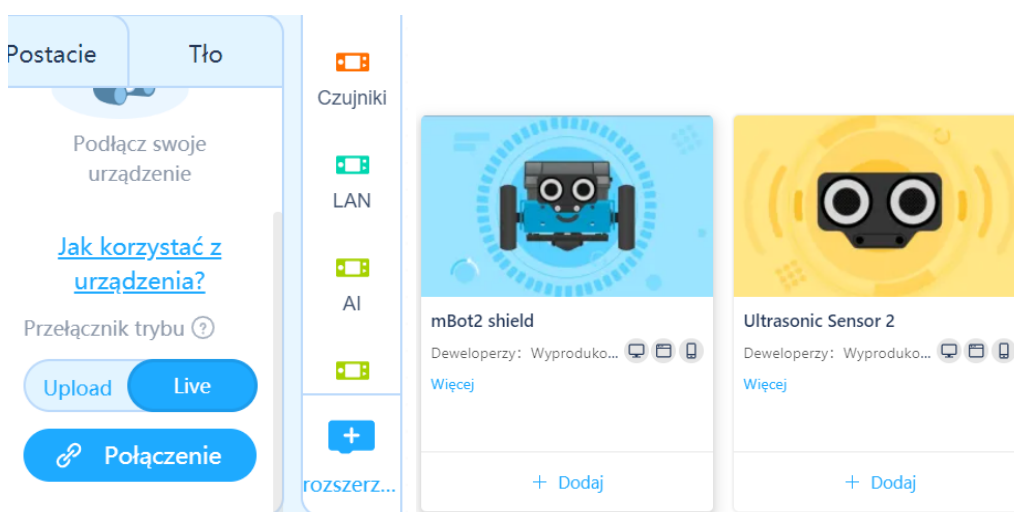
Kolejne kroki (kolejność wykonywania kroków jest znacząca dla prawidłowego działania programu):

1. Podłączenie modułu Bluetooth Dongle; moduł powinien zacząć powoli migać niebieskim światłem.

2. Naciśnięcie na module ikony Bluetooth; moduł powinien zacząć szybko migać niebieskim światłem.
3. Włączenie mBota2; moduł powinien zacząć świecić ciągłym światłem (to znak, że moduł sparował się z robotem).
4. Włączenie mBlocka:
 - a. w wersji online: Przy pierwszym uruchomieniu na danym komputerze w lewym dolnym rogu usunięcie CyberPi krzyżykiem i kliknięcie Dodaj w celu wybrania mBota2. Po otwarciu biblioteki urządzeń warto najechać na lewy górny róg pola z mBotem2, gdzie pojawi się biała gwiazdka. Po kliknięciu, biała gwiazdka zmieni się na niebieską, a tym samym mBot2 zostanie ustawiony jako urządzenie domyślne i będzie pojawiał się zawsze po uruchomieniu mBlocka. Przy pierwszym uruchomieniu system będzie informował o uruchomieniu wtyczki mLink2 i “przeładowuje stronę”.

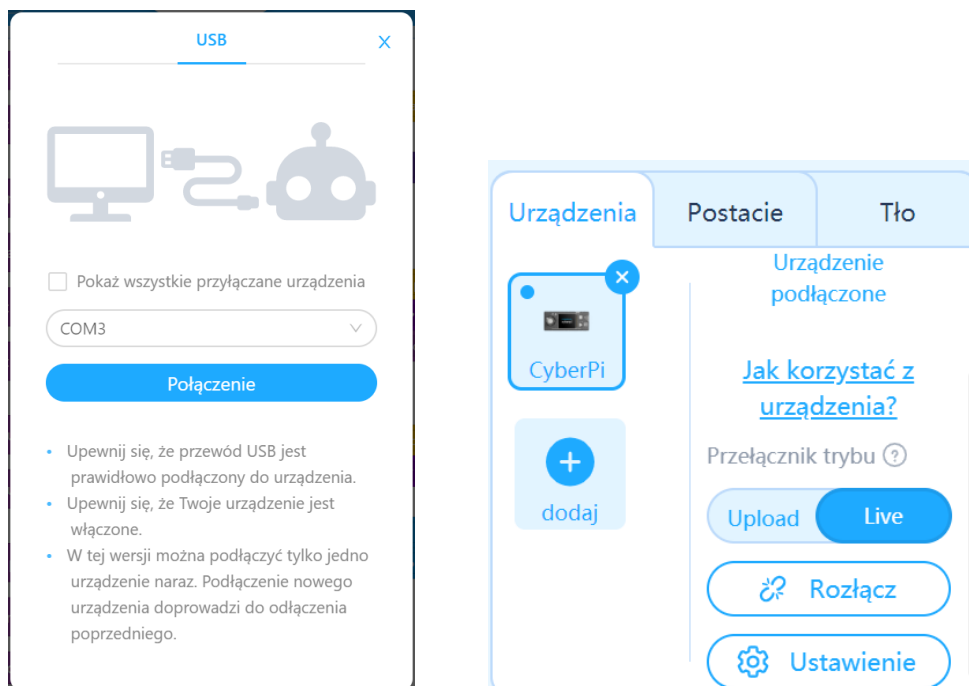


- b. w wersji offline: Przy pierwszych uruchomieniu na danym komputerze dodanie rozszerzeń: podwozie (shield) oraz czujnik ultrasoniczny (ultrasonic sensor 2) poprzez kliknięcie +Dodaj. Dodatkowo jeśli obok rozszerzenia pojawia się ikona zielonej strzałki, należy ją kliknąć, żeby zaktualizować dane rozszerzenie.



5. Połączenie robota z komputerem poprzez kliknięcie guzika Połączenie. Pojawi się komunikat z nazwą portu (standardowo będzie to COM3 lub COM4), gdzie należy kliknąć ponownie Połączenie. Można

sprawdzić czy robot na pewno połączył się z komputerem szukając niebieskiej kropki w lewym górnym rogu jego ikony i zmianie niebieskiego przycisku Połączenie na biały przycisk Rozłącz.



Zadanie 4: pierwszy program dla mBota

Nauczyciel pokazuje uczniom, gdzie znaleźć skrypty mBota. Uczniowie przeglądają różne podgrupy tych bloczków i zastanawiają się, z których bloczków mógł korzystać robot wykonując wcześniej odtwarzane programy (znajdujące się automatycznie w jego pamięci).

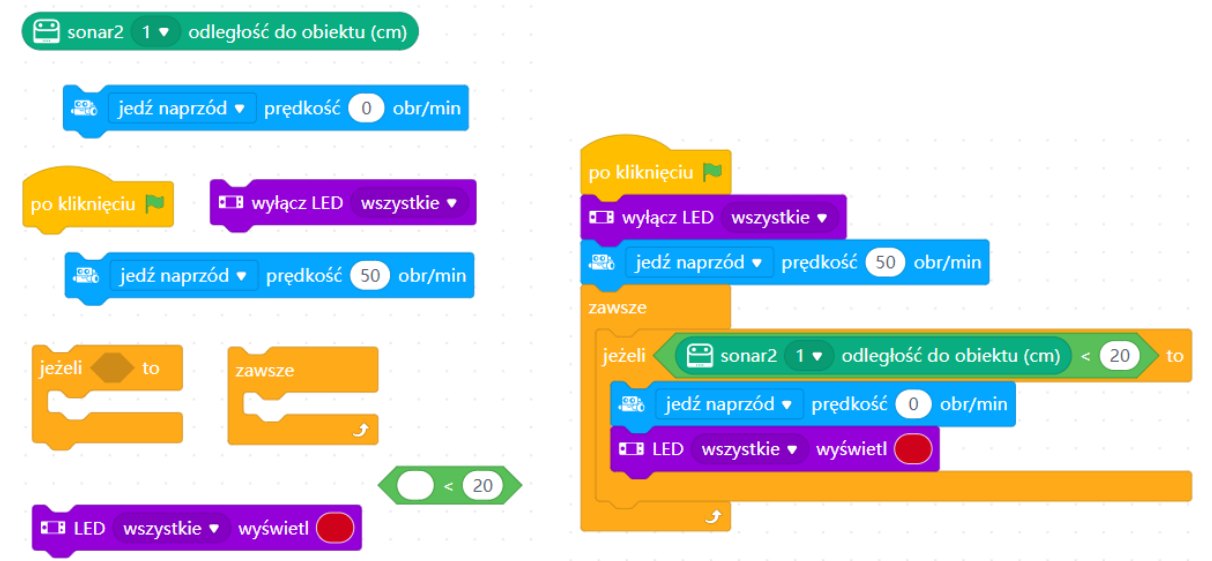
W naszym pierwszym programie użyjemy ultrasonicznego czujnika odległości. Chcemy,

by robot po napotkaniu przeszkody zatrzymał się i wyświetlił na pasku kolor czerwony.

Konieczne jest tu ustalenie, w jakiej odległości od przeszkody ma zatrzymać się robot. W podanych poniżej przykładzie ustalamy, że będzie to wartość 20.

Wskazówka: Warto już tu zwrócić uczniom uwagę na to, że roboty programujemy w zakładce Urządzenia, a postacie na ekranie w zakładce Postacie.

Napisanie takiego programu, może być na początku trudne. Jeśli tak jest, nauczyciel podpowiada uczniom np. układając na polu roboczym rozsypankę z wszystkich potrzebnych bloczków i zachęca uczniów do poprawnego ułożenia skryptu. Poprawnie napisany skrypt znajduje się po prawej stronie.



3. Podsumowanie i ewaluacja – 5 minut

Nauczyciel pyta uczniów, co najbardziej podobało się im w zajęciach, a co najmniej. Pyta także o atmosferę panującą w klasie oraz o tempo pracy. Zapowiada, że na kolejnej lekcji uczniowie napiszą kolejne programy dla mBotów2.

Zadanie domowe: proszę obejrzeć filmik na YouTube <https://www.youtube.com/watch?v=uxpoP175mOU> i przyjrzeć się podstawowym funkcjonalnościom mBota2.

Uwagi/alternatywy:

Zachęcamy do utrwalania materiałów z przebiegu zajęć. Mogą to być zdjęcia, filmy, notatki, zrzuty ekranu, ciekawe lub zabawne teksty/informacje, które pojawiły się podczas lekcji. Posłużą one do udokumentowania przebiegu programu #SuperKoderzy.

Robo-matematycy

Autorki: Joanna Płatkowska-Nęcka, Karolina Czerwińska

Lekcja 2:

Silniki, głośniki i LEDy, czyli poznajemy się bliżej z mBotem2

Lekcja, podczas której uczniowie oswoją się z programowaniem mBota2, wykorzystując jego LEDy, głośnik i silniki. Dowiedzą się, ile różnych kolorów światła może wyświetlić robot. Podejmą także próbę muzykowania przy użyciu mBota2.

Cele zajęć:

Uczeń powinien:

- wykorzystać informacje dotyczące części robota do zaprogramowania ruchu,
- wykorzystać światła robota do pokazania flag wybranych państw lub "opowieści kolorami",
- stworzyć krótką melodię graną przez robota,
- stworzyć krótką grę, która wykorzystuje wcześniej poznane funkcjonalności Scratcha i mBota2.

Materiały pomocnicze:

- roboty mBot2
- program mBlock (do kodowania robota) w wersji online <https://ide.mblock.cc/> lub offline <http://www.mblock.cc/download/>
- komputery stacjonarne lub laptopy (ze sprawnym portem USB)
- laptop lub komputer nauczyciela podłączony do tablicy multimedialnej lub projektora

Pojęcia kluczowe:

→ robot → silnik → RGB → LED → głośnik → mikrokontroler CyberPi → mBlock

Czas na realizację zajęć: 45 minut (1 godzina lekcyjna)

Metody pracy:

- grupowa - praca w grupach z robotem,
- ćwiczenia praktyczne.

Treści programowe (związek z podstawą programową)

Podstawa programowa kształcenia ogólnego dla szkół podstawowych – II etap edukacyjny – klasy IV-VI, informatyka:

- I. Rozumienie, analizowanie i rozwiązywanie problemów. Uczeń:
 2. formułuje i zapisuje w postaci algorytmów polecenia składające się na:
 - 3) sterowanie robotem lub obiektem na ekranie;
 3. w algorytmicznym rozwiązywaniu problemu wyróżnia podstawowe kroki: określenie problemu i celu do osiągnięcia, analiza sytuacji

problemowej, opracowanie rozwiązania, sprawdzenie rozwiązania problemu dla przykładowych danych, zapisanie rozwiązania w postaci schematu lub programu.

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych. Uczeń:

1. projektuje, tworzy i zapisuje w wizualnym języku programowania:

- 1) pomysły historyjek i rozwiązania problemów, w tym proste algorytmy z wykorzystaniem poleceń sekwencyjnych, warunkowych i iteracyjnych oraz zdarzeń,
- 2) prosty program sterujący robotem lub innym obiektem na ekranie komputera;

2. testuje na komputerze swoje programy pod względem zgodności z przyjętymi założeniami i ewentualnie je poprawia, objaśnia przebieg działania programów;

IV. Rozwijanie kompetencji społecznych. Uczeń:

1. uczestniczy w zespołowym rozwiązaniu problemu posługując się technologią taką jak: poczta elektroniczna, forum, wirtualne środowisko kształcenia, dedykowany portal edukacyjny;
2. identyfikuje i docenia korzyści płynące ze współpracy nad wspólnym rozwiązywaniem problemów;

V. Przestrzeganie prawa i zasad bezpieczeństwa. Uczeń:

1. posługuje się technologią zgodnie z przyjętymi zasadami i prawem; przestrzega zasad bezpieczeństwa i higieny pracy;

Przebieg zajęć:

1. Wprowadzenie w tematykę i integracja grupy

Nauczyciel tłumaczy uczniom, że na dzisiejszych zajęciach napiszą wspólnie kilka programów, w których wykorzystane będą różne funkcjonalności mBota2. Prosi uczniów o uruchomienie programu mBlock, podłączenie modemu Bluetooth Dongle komputera i o sparowanie robota z komputerem.

2. Część zasadnicza

Zadanie 1: programowanie ruchu robota

Nauczyciel poleca uczniom zaprogramować robota tak, by przejechał z punktu A do punktu B, np. od komputera, przy którym pracuje uczeń do biurka nauczyciela, po drodze obracając się kilka razy wokół własnej osi i skręcając kilka razy w prawo i w lewo.

Sytuacją idealną jest, gdy uczniowie samodzielnie znajdą sposób na rozwiązanie tego zadania. Jeśli okaże się, że mają z nim problem, nauczyciel podpowiada, jakie bloczki należy wykorzystać do rozwiązania tego zadania (np. wyświetla je na projektorze). Uczniowie mogą wykonując to zadanie odkryć, że można korzystać z bloczków typu jedź naprzód czy skręcaj lub bloczków sterujących osobno każdym z silników. Równocześnie mogą zauważyć, że przy bloczkach skręcaj w prawo lub skręcaj w lewo obrót zawsze będzie odbywał się po tym samym łuku. Żeby robot skręcał po łagodniejszym, większym łuku trzeba skorzystać z programowania obu silników osobno.

Dobrze, żeby w trakcie tego zadania uczniowie również zbadali jaki odcinek przejdzie robot w określonym czasie przy różnych prędkościach, a także zastanowili się, w jakich sytuacjach lepiej użyć bloczków jedź przez x sekund, a kiedy lepiej wybrać bloczek określający w centymetrach odległość do przejechania. Skrypt mógłby wyglądać następująco (wykorzystano w nim wszystkie opisane wyżej opcje ruchu):



Jeśli uczniowie szybko wykonają zadanie, można dodawać kolejne sekwencje ruch robota porównując przy okazji działanie różnych bloczków.

Poprawnie zaprogramowany robot będzie wykonywał polecenia jazdy z dużą dokładnością, ale ze względu na zmianę podłoża czy niewielką przeszkodę po drodze, trasa może delikatnie ulec zmianie. Warto na taką okoliczność zaprogramować sobie sterowanie z poziomu klawiatury, które pozwoli skorygować końcową pozycję robota czy zatrzymać go, jeśli przez przypadek zacznie odjeżdżać daleko od punktu startowego.



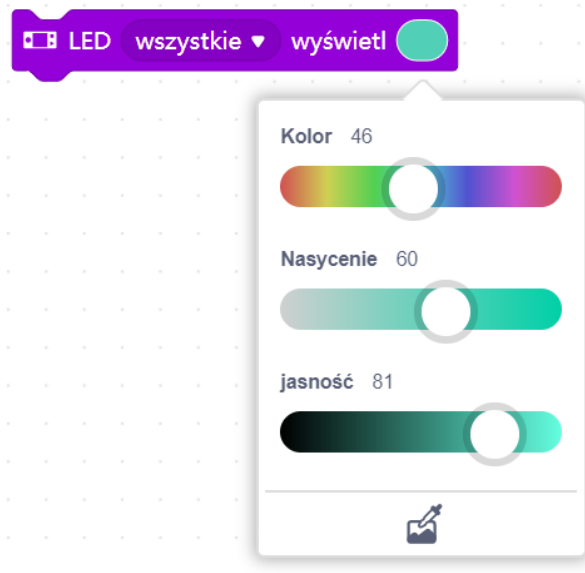
Zadanie 2: programowanie LEDów robota

Nauczyciel zaprasza uczniów do wykonania kolejnego zadania, podczas którego nauczą się wyświetlać na robocie różne kolory. Mówi uczniom, że robot jest wyposażony w pięć diod LED, na których można łatwo wskazać dowolny kolor do wyświetlenia. Warto tu opowiedzieć uczniom, że każdy z wyświetlanych kolorów powstaje z połączenia trzech różnych kolorów. Jest to model RGB. Podpowiada uczniom, że skrót "RGB" pochodzi z angielskiego i oznacza trzy barwy - zachęca uczniów, by zgadli o jakie barwy chodzi (red - czerwony; green - zielony; blue - niebieski)

Nauczyciel wyświetla stronę internetową: <http://www.colorsfire.com/rgb-color-wheel/>. Przesuwając suwakiem, widać, jak zmieniają się wartości trzech modelowych kolorów (RGB) i jakie kolory wówczas powstają. mBlock pozwala oczywiście na wybranie koloru światła bezpośrednio z palety, bez konieczności ustawiania wartości RGB. Pokazanie palety RGB uczniom może jednak pokazać im, w jaki sposób te różne kolory powstają. Uczniowie mogą poeksperymentować i odkryć jakie kolory mogą stworzyć:



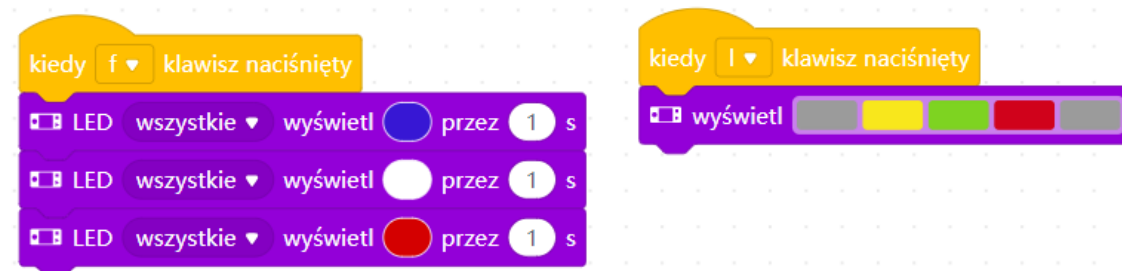
Następnie nauczyciel proponuje uczniom, żeby spróbowali, jakie barwy światła można uzyskać manipulując kolorem, nasyceniem i jasnością:



W programie mBlock można zaprogramować LEDy mBota2 tak, że każda z pięciu diod będzie świecić innym kolorem, a nawet można zaprogramować zmieniające się kolory. Nauczyciel może zaproponować uczniom mniej i bardziej złożoną wersję wykorzystania LEDów w tym zadaniu.

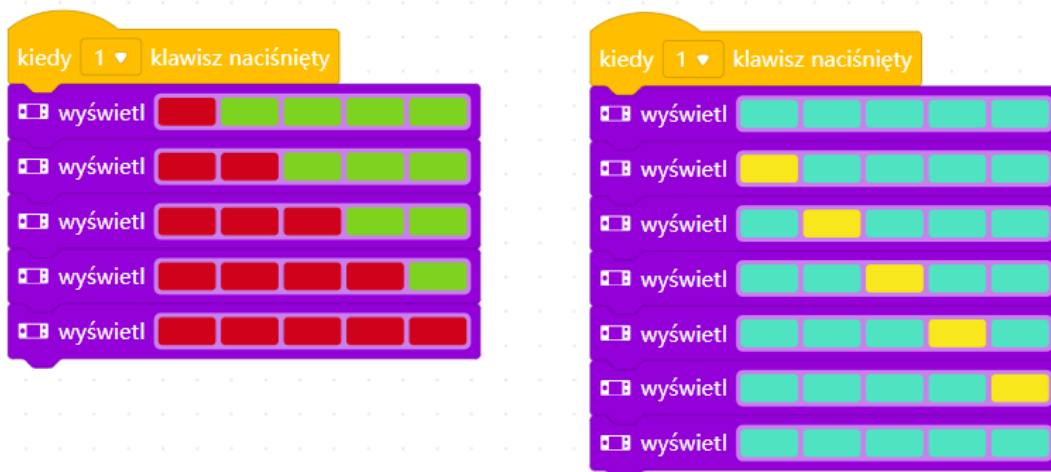
W wersji mniej złożonej nauczyciel zaprasza uczniów do wyszukania flagi dowolnego kraju lub rozdaje uczniom załącznik nr 1 (po jednym wydruku dla każdej z grup). Prosi uczniów, by każda z grup wyświetliła na swoim robocie kolory flagi wybranego państwa. Pozostałe grupy zgadują, o flagę jakiego państwa chodzi. Uczniowie mogą to zrobić na dwa sposoby. W pierwszym wszystkie diody świecą na wybrany kolor przez krótką chwilę, następnie zmieniają się, aż wyświetlą się dwa lub trzy kolory znajdujące się na wybranej fladze. Mogą też wyświetlić te kolory równocześnie wybierając bloczek wyświetl i obok domyślnie kolory tęczy. Po kliknięciu w ten bloczek otworzy się okno, w którym można wybierać kolory podobnie jak w programie Paint. Najpierw klikamy w jeden z dziewięciu kolorów, a następnie w jedną z pięciu diod. Możemy skorzystać z gumki, żeby daną diodę wyłączyć. Po kliknięciu w ikonę dyskietki możemy dany układ zapisać. Po kliknięciu w ikonę trójkąta (play) możemy wyświetlić układy kolorystyczne po kolei.

Poniżej przykład flagi Francji wyświetlony pierwszym sposobem i flagi Litwy wyświetlony drugim sposobem:



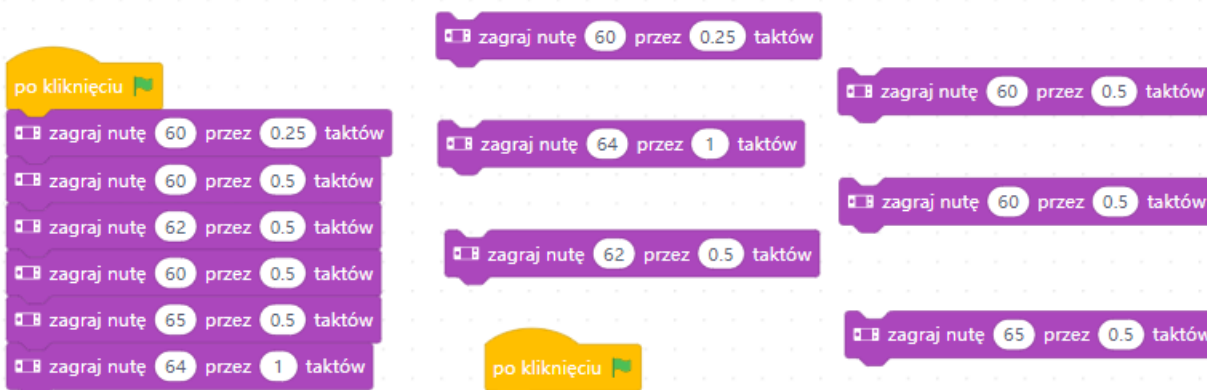
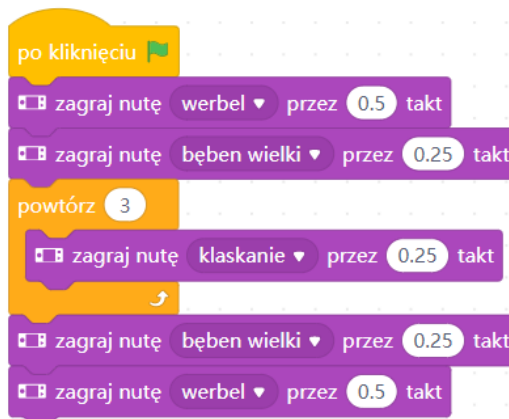
W wersji bardziej złożonej uczniowie mogą spróbować ułożyć "kolorowe historyjki". W tym celu korzystając z bloczka display projektują i zapisują kilka układów kolorystycznych, które kiedy pojawią się jeden po drugim staną się animacją. Uczniowie mogą odgadywać jaką historyjkę chciała zaimować każda z grup.

Poniżej przykład animacji pokazującej pożar lasu oraz rybki przepływającej przez rzekę.



Zadanie 3: programowanie dźwięków

Nauczyciel przypomina uczniom, że mBot2 wyposażony jest w głośnik, a także mikrofon. Wyjaśnia jak można na nim grać różne proste melodie. Zachęca uczniów do testów i wypróbowania dźwięków nut, wbudowanych nagrań robota czy nagrywania własnych dźwięków (mBot2 może zapamiętać i odtworzyć jedno nagranie). Nauczyciel może też umieścić na tablicy rozsypankę, z której uczniowie mogą ułożyć znaną melodię:



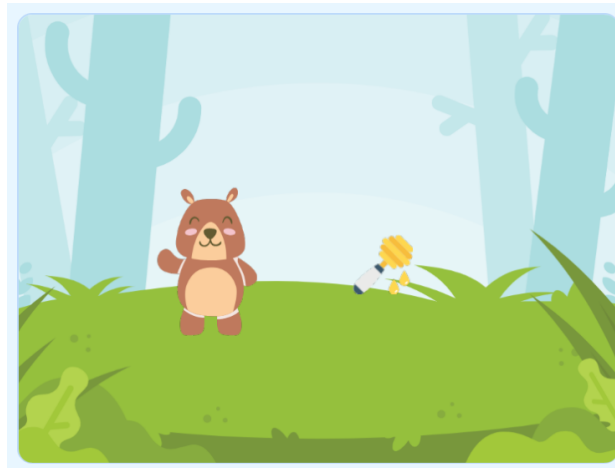
Zadanie 4: włączanie robota do gier – zadanie dodatkowe

Jeśli grupa sprawnie pracuje, nauczyciel proponuje wykonanie dodatkowego zadania, w którym uczniowie napiszą krótką grę komputerową, w którą włączony będzie robot.

Gra polega na tym, że niedźwiedź próbuje złapać pojawiające się w losowo wybranych miejscach łyżeczki z miodem. Za każdym razem, gdy taką łyżeczkę złapie, robot świeci niebieskim światłem. Niedźwiedź porusza się dzięki czujnikowi odległości robota – ręka przyłożona bliżej lub dalej od robota będzie odpowiednio poruszać duszkiem w lewo lub w prawo (tu trzeba zwrócić uwagę, w którą stronę zwrócony jest robot). W przypadku uczniów, którzy dobrze radzą sobie z układem współrzędnych, można zaproponować, żeby miód powoli spadał w różnych miejscach zamiast pojawiać się w losowo wybranych miejscach. Poniżej podano obie wersje skryptu dla miodu.

Oczywiście bohaterami gry mogą być zupełnie inne postaci czy przedmioty i znajdować się w zupełnie innym miejscu.

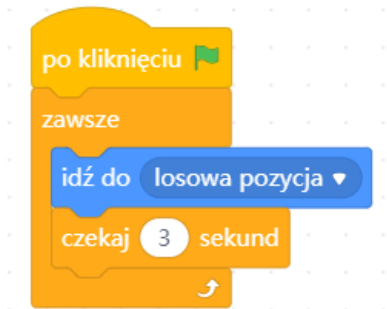
Nauczyciel moderuje dyskusję i zachęca uczniów do zaproponowania, jak może być napisany skrypt gry.



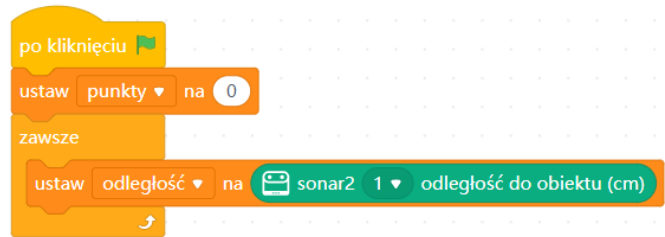
Skrypt dla niedźwiedzia:



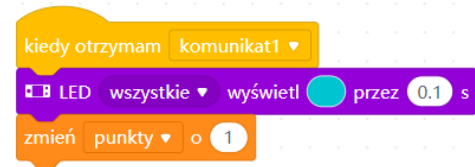
Skrypt dla miotu w wersji podstawowej:



Skrypt dla mBota:



Skrypt dla miotu w wersji rozszerzonej:



3. Podsumowanie i ewaluacja

Nauczyciel pyta uczniów, co najbardziej podobało się im w zajęciach, a co najmniej. Pyta także, czy tempo pracy nie było zbyt szybkie i czy jakieś kwestie zostały niewystarczająco wyjaśnione.

Uwagi/alternatywy:

Zachęcamy do utrwalania materiałów z przebiegu zajęć. Mogą to być zdjęcia, filmy, notatki, zrzuty ekranu, ciekawe lub zabawne teksty/informacje, które pojawiły się podczas lekcji. Posłużą one do dokumentowania przebiegu programu #SuperKoderzy.

Załącznik nr 1



- | | | |
|--------------|---------------------|--------------------------|
| 1. Turcja | 17. Serbia | 33. Białoruś |
| 2. Włochy | 18. Monako | 34. Liechtenstein |
| 3. Słowenia | 19. Luksemburg | 35. Bułgaria |
| 4. Estonia | 20. Chorwacja | 36. Bośnia i Hercegowina |
| 5. Litwa | 21. Finlandia | 37. Malta |
| 6. Belgia | 22. Kosowo | 38. Praga |
| 7. Szwecja | 23. Czarnogóra | 39. Holandia |
| 8. Dania | 24. Macedonia | 40. Niemcy |
| 9. Francja | 25. Hiszpania | 41. San Marino |
| 10. Łotwa | 26. Ukraina | 42. Watykan |
| 11. Albania | 27. Austria | 43. Rosja |
| 12. Andora | 28. Wielka Brytania | 44. Szwajcaria |
| 13. Węgry | 29. Polska | 45. Islandia |
| 14. Irlandia | 30. Norwegia | 46. Mołdawia |
| 15. Lizbona | 31. Słowacja | 47. Grecja |
| 16. Cypr | 32. Rumunia | |

Robo-matematycy

Autorki: Joanna Płatkowska-Nęcka, Karolina Czerwińska

Lekcja 3:

Programowanie – sposób na łatwe obliczenia

Zajęcia, na których uczniowie odkrywają, w jaki sposób stosować bloczki w Scratchu lub mBlocku, żeby zapisywać liczby i działania matematyczne. Poznają komendę zapytaj i czekaj → odpowiedź i ćwiczą korzystanie ze zmiennych.

Cele lekcji:

Uczeń powinien:

- zapisywać cztery podstawowe działania w Scratchu,
- korzystać ze zmiennych i wyrażeń do zapisywania liczb i działań,
- tworzyć proste pytania i formułować odpowiedzi, korzystając z pętli warunkowych i komendy zapytaj i czekaj → odpowiedź.

Materiały pomocnicze:

- program mBlock,
- komputery stacjonarne lub laptopy,
- projektor, ekran lub tablica interaktywna – podłączone do komputera nauczyciela.
- wybrane cztery zadania z <https://scratch.mit.edu/studios/4907086/> (po jednym z mnożenia i dzielenia liczb całkowitych, dodawania i odejmowania ułamków, pól i obwodów figur i zadań tekstowych),

Pojęcia kluczowe:

→ działanie → zmienna → wybór losowy → pytanie i odpowiedź

Czas realizacji: 45 min.

Metody pracy:

- grupowa – praca w grupach z robotem,
- ćwiczenia praktyczne.

Treści programowe:

Podstawa programowa kształcenia ogólnego dla szkół podstawowych – II etap edukacyjny – klasy IV-VI, matematyka:

I. Liczby naturalne w dziesiętkowym układzie pozycyjnym. Uczeń:

1. zapisuje i odczytuje liczby naturalne wielocyfrowe;
3. porównuje liczby naturalne;

II. Działania na liczbach naturalnych. Uczeń:

1. dodaje i odejmuje w pamięci liczby naturalne dwucyfrowe lub większe, liczbę jednocyfrową dodaje do dowolnej liczby naturalnej i odejmuje od dowolnej liczby naturalnej;
2. dodaje i odejmuje liczby naturalne wielocyfrowe sposobem pisemnym i za pomocą kalkulatora;
3. mnoży i dzieli liczbę naturalną przez liczbę naturalną jednocyfrową, dwucyfrową lub trzycyfrową sposobem pisemnym, w pamięci (w najprostszych przykładach) i za pomocą kalkulatora (w trudniejszych przykładach);

VI. Elementy algebry. Uczeń:

1. korzysta z nieskomplikowanych wzorów, w których występują oznaczenia literowe, opisuje wzór słowami;

XIV. Zadania tekstowe. Uczeń:

1. czyta ze zrozumieniem tekst zawierający informacje liczbowe;
5. do rozwiązywania zadań osadzonych w kontekście praktycznym stosuje poznaną wiedzę z zakresu arytmetyki i geometrii oraz nabyte umiejętności rachunkowe, a także własne poprawne metody;

Podstawa programowa kształcenia ogólnego dla szkół podstawowych – II etap edukacyjny – klasy IV-VI, informatyka:

I. Rozumienie, analizowanie i rozwiązywanie problemów. Uczeń:

2. formułuje i zapisuje w postaci algorytmów polecenia składające się na:

1. rozwiązanie problemów z życia codziennego i z różnych przedmiotów, np. liczenie średniej, pisemne wykonanie działań arytmetycznych, takich jak dodawanie i odejmowanie,
3. w algorytmicznym rozwiązywaniu problemu wyróżnia podstawowe kroki: określenie problemu i celu do osiągnięcia, analiza sytuacji problemowej, opracowanie rozwiązania, sprawdzenie rozwiązania problemu dla przykładowych danych, zapisanie rozwiązania w postaci schematu lub programu.

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych. Uczeń:

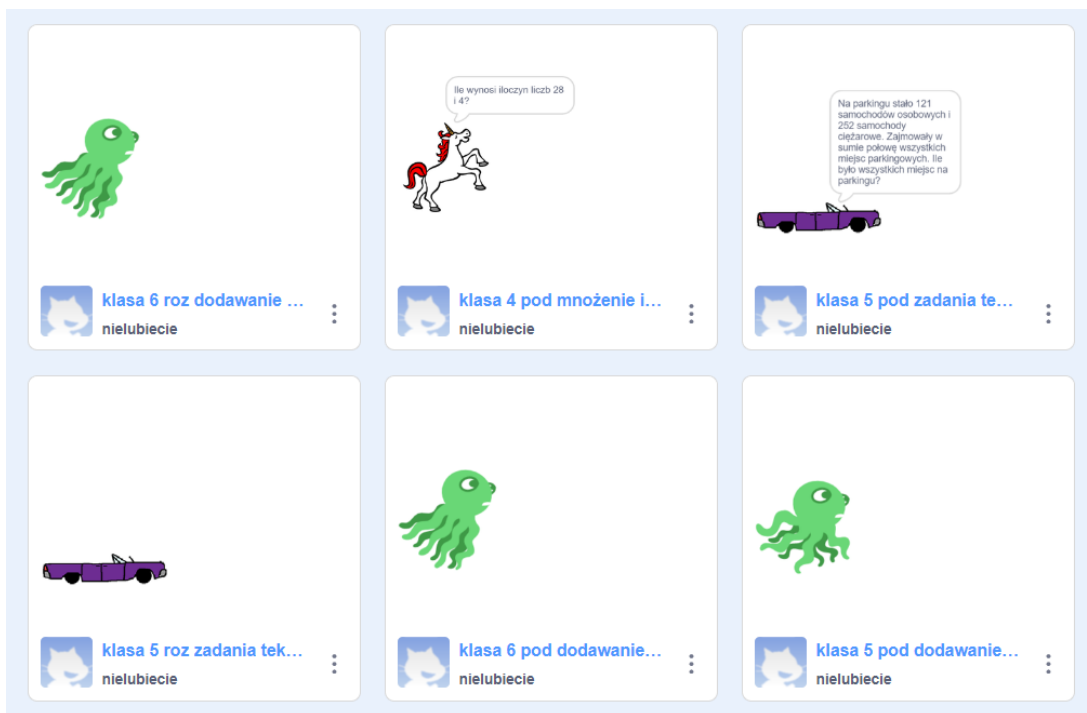
2. testuje na komputerze swoje programy pod względem zgodności z przyjętymi założeniami i ewentualnie je poprawia, objaśnia przebieg działania programów;

IV. Rozwijanie kompetencji społecznych. Uczeń:

1. uczestniczy w zespołowym rozwiązaniu problemu posługując się technologią taką jak: poczta elektroniczna, forum, wirtualne środowisko kształcenia, dedykowany portal edukacyjny;
2. identyfikuje i docenia korzyści płynące ze współpracy nad wspólnym rozwiązywaniem problemów;

Wprowadzenie w tematykę i integracja grupy

Uczniowie mają za zadanie zagrać w cztery z szesnastu gier matematycznych ze studia Robo-matematycy, znajdującego się pod adresem <https://scratch.mit.edu/studios/4907086/> (po jednym z mnożenia i dzielenia liczb całkowitych, dodawania i odejmowania ułamków, pól i obwodów figur i zadań tekstowych).



Po zagraniu w gry nauczyciel pyta o to, jakie liczby i jakie działania pojawiają się w tych grach. Zabierają pomysły i zapisują je na tablicy lub na dużym arkuszu papieru. Powinny pojawić się tu następujące propozycje: dodawanie, odejmowanie, mnożenie, dzielenie, ułamki dziesiętne i procenty (te ostatnie tylko dla klasy 6).

Część zasadnicza

Zadanie 1

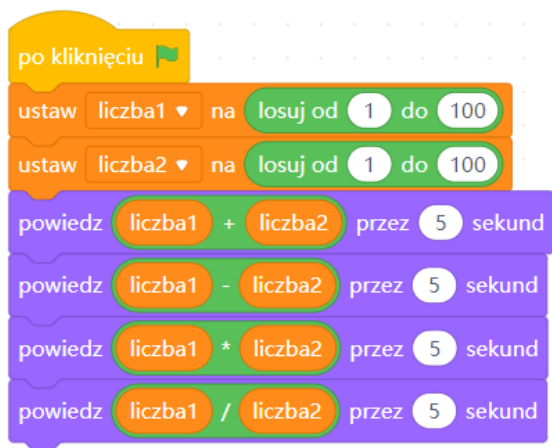
Uczniowie tworzą nowy projekt i przyglądają się bloczkom w grupie “wyrażenia”. Mają za zadanie napisać prosty program, w którym duszek mówi kolejno, jakie są: suma, różnica, iloczyn i iloraz dwóch konkretnych liczb. Nauczyciel wybiera te dwie liczby, przy czym pierwsza musi być wielokrotnością drugiej.

Uczniowie w parach sprawdzają poprawność napisanego skryptu.



Zadanie 2

Kolejnym zadaniem uczniów jest napisanie podobnego skryptu, jak w poprzednim zadaniu, jednak tym razem liczby mają być za każdym razem inne (losowo wybrana para liczb do wszystkich czterech działań przy każdym uruchomieniu programu). Uczniowie tworzą dwie zmienne (“liczba 1” i “liczba 2”), a następnie ustawiają je jako wartości losowe (przedział ustalony w zależności od możliwości uczniów).



Wskazówka: Aby ułatwić przeanalizowanie działania tego programu, warto w bloczkach zostawić zmienne zaznaczone ptaszkiem, dzięki czemu będą wyświetlać się na ekranie, a także o zanotowanie wyników na kartce.

Nauczyciel prosi uczniów o kilkukrotne uruchomienie programu. Pyta, czy zauważają jaką zasadę w przypadku różnych działań. Okazuje się, że w przypadku dodawania i mnożenia wynik zawsze jest liczbą całkowitą dodatnią. W przypadku odejmowania co jakiś czas wynikiem są liczby ujemne, a w przypadku dzielenia - ułamki. Niezbędne jest dokładne przyjrzenie się obu tym przypadkom.

Nauczyciel wyjaśnia, że w wielu przypadkach wynik ujemny nie będzie pasował jako rozwiązanie zadania tekstowego.

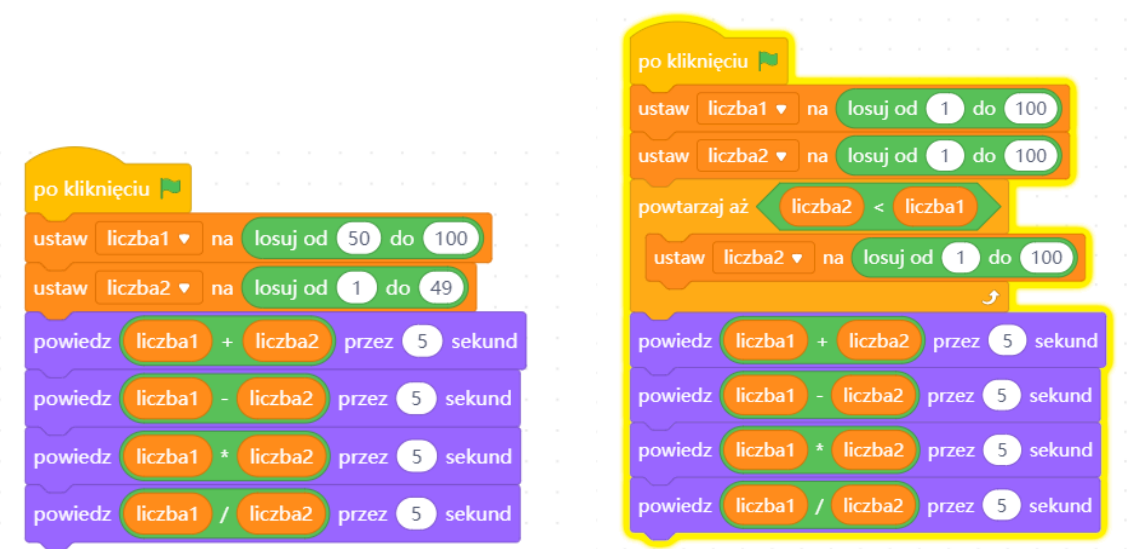
Jako przykład nauczyciel podaje zadanie: "Zuzia ma [liczba 1] książek, a Janek ma [liczba 2] książek. O ile więcej książek ma Zuzia od Janka?" i prosi uczniów o podanie odpowiedzi dla następujących danych:

liczba 1 = 10, liczba 2 = 7

liczba 1 = 7, liczba 2 = 10

Nauczyciel prosi uczniów o komentarz, czy przy drugim zestawie danych odpowiedź będzie miała sens (odpowiedź "-3 książki" nie będzie miała sensu). Uczniowie wraz z nauczycielem zastanawiają się, w jaki sposób zapewnić, żeby odjemna wybierana losowo była zawsze większa od odjemnika. Rozwiązania są dwa (można uczniom pokazać oba lub to, które jest dopasowane do ich możliwości):

- Ustawienie niższego zakresu losowania "liczby1" niż "liczby2"
- Zastosowanie bloczka «powtarzaj aż» (szczególnie, jeśli nie chcemy zawęzić zakresu losowania)



Uczniowie wprowadzają wybrane przez siebie (lub nauczyciela) rozwiązanie do usprawnienia działania skryptu.

Zadanie 3

W przypadku dzielenia wynik będący ułamkiem może nie być sensowną odpowiedzią w przypadku wielu zadań tekstowych.

Jako przykład nauczyciel podaje zadanie: “W szkole podstawowej uczy się [liczba 1] uczniów. Z okazji dnia sportu podzielono ich na [liczba 2] zespołów. Ilu uczniów było w każdym zespole?” i prosi uczniów o podanie odpowiedzi dla następujących danych:

$$\text{liczba 1} = 120, \text{liczba 2} = 12$$

$$\text{liczba 1} = 114, \text{liczba 2} = 12$$

Nauczyciel prosi uczniów o komentarz, czy przy drugim zestawie danych odpowiedź będzie miała sens (uczniów nie dzielimy na ułamki 😊, drużyna nie może liczyć "dziewięciu i pół zawodnika").

Uczniowie wspólnie z nauczycielem zastanawiają się, jak rozwiązać tę trudność. Jednym z pomysłów jest to, że liczba dzieli się bez reszty, jeśli jest iloczynem dwóch liczb całkowitych. W związku z tym propozycja jest następująca: tworzymy dodatkową zmienną, nazwaną roboczo “mnożnikiem”, a “liczbę 1” definiujemy jako iloczyn “liczby 2” i mnożnika (gdzie obie wartości są ustalane losowo w zakresie liczbowym odpowiednim do grupy). Uczniowie zmieniają swój program tak, żeby nie pojawiała się w nim reszta z dzielenia. Warto podpowiedzieć tu uczniom, żeby przy zmiennej “mnożnik” usunęli zaznaczenie ptaszkiem, ponieważ ta zmienna jest równa wynikowi. W scenariuszach nie przewidziano takich zadań, w których równocześnie wynik dzielenia i odejmowania musi sprawdzać się dobrze w różnych zadaniach tekstowych.



Zadanie 4

Nauczyciel ustala z uczniami, z jakich elementów składa się zadanie matematyczne. Ważne jest, by ustalić, że istotne są dane, pytanie i odpowiedź.

#SuperKoderzy / Robo-matematycy / Programowanie – sposób na łatwe obliczenia

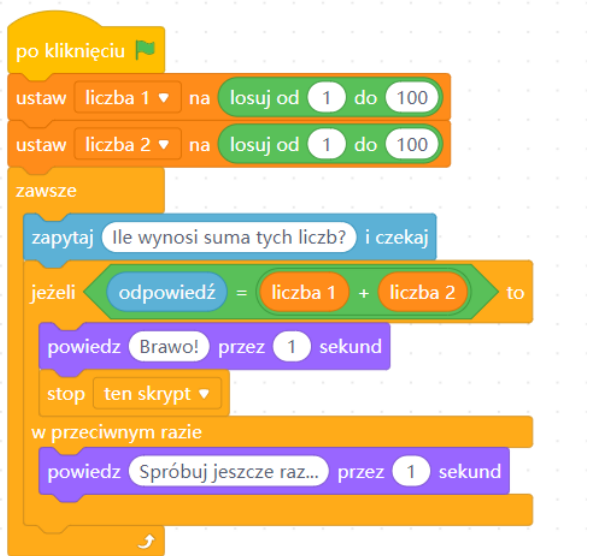
Nauczyciel pokazuje uczniom proste zadanie i omawia z uczniami działanie bloczka “zadaj pytanie i czekaj” oraz wykorzystanie pętli warunkowej z odpowiedzią. Warto zapisać ten skrypt, ponieważ będzie wykorzystany na kolejnych zajęciach.



Podsumowanie i ewaluacja

Uczniowie z rozsypanki bloczków układają program pytający gracza o sumę dwóch losowo wybranych liczb. Nauczyciel może skorzystać z wydrukowanego [Załącznika nr 1](#) lub wyświetlić rozsypankę na tablicy. Następnie uczniowie w parach testują napisane przez siebie programy.

Skrypt ułożony prawidłowo wygląda następująco:



Załącznik nr 1



Robo-matematycy

Autorki: Joanna Płatkowska-Nęcka, Karolina Czerwińska

Lekcja 4:

Ułamki dziesiętne, procenty oraz... mBot2 matematyk

Zajęcia, na których uczniowie odkrywają, w jaki sposób zapisywać ustalone i losowo wybrane ułamki dziesiętne oraz procenty. Uczą się używać bloczka “połącz”, żeby zapisywać poprawnie treść zadań matematycznych, a także zagnieżdżania bloczków w taki sposób, aby zachować poprawną kolejność wykonywania działań. Łączą także zadania obliczeniowe ze skryptem robota.

Cele lekcji:

Uczeń powinien:

- zapisywać cztery podstawowe działania w Scratchu lub mBlocku,
- korzystać ze zmiennych i wyrażeń do zapisywania liczb i działań, w tym ułamków i procentów, pamiętając o kolejności wykonywania działań,
- zapisywać dane i pytania w zadaniach tekstowych w jasny i poprawny sposób,
- łączyć skrypt z funkcjonalnościami robota.

Materiały pomocnicze:

- roboty mBot2,
- program mBlock,
- komputery stacjonarne lub laptopy,
- projektor, ekran lub tablica interaktywna – podłączone do komputera nauczyciela,
- gra “Kto ma rację? A czy Z?” - <https://scratch.mit.edu/projects/210935897/>
- gra “Kto powiedział to lepiej?” - <https://scratch.mit.edu/projects/211060770/>

Pojęcia kluczowe:

→ działanie → zmienna → pytanie → kolejność

Czas realizacji: 45 min.

Metody pracy:

- pogadanka,
- ćwiczenia praktyczne.

Treści programowe:

Podstawa programowa kształcenia ogólnego dla szkół podstawowych – II etap edukacyjny – klasy IV-VI, matematyka:

I. Liczby naturalne w dziesiętkowym układzie pozycyjnym. Uczeń:

1. zapisuje i odczytuje liczby naturalne wielocyfrowe;
3. porównuje liczby naturalne;

II. Działania na liczbach naturalnych. Uczeń:

1. dodaje i odejmuje w pamięci liczby naturalne dwucyfrowe lub większe, liczbę jednocyfrową dodaje do dowolnej liczby naturalnej i odejmuje od dowolnej liczby naturalnej;

2. dodaje i odejmuje liczby naturalne wielocyfrowe sposobem pisemnym i za pomocą kalkulatora;
3. mnoży i dzieli liczbę naturalną przez liczbę naturalną jednocyfrową, dwucyfrową lub trzycyfrową sposobem pisemnym, w pamięci (w najprostszych przykładach) i za pomocą kalkulatora (w trudniejszych przykładach);
5. stosuje wygodne dla siebie sposoby ułatwiające obliczenia, w tym przemienność i łączność dodawania i mnożenia oraz rozdzielność mnożenia względem dodawania;
- IV. Ułamki zwykłe i dziesiętne. Uczeń:
 2. przedstawia ułamek jako iloraz liczb naturalnych;
 2. dodaje, odejmuje, mnoży i dzieli ułamki dziesiętne w pamięci (w przykładach najprostszych), pisemnie i za pomocą kalkulatora (w przykładach trudnych);
- VI. Elementy algebry. Uczeń:
 1. korzysta z nieskomplikowanych wzorów, w których występują oznaczenia literowe, opisuje wzór słowami;
- XIV. Zadania tekstowe. Uczeń:
 1. czyta ze zrozumieniem tekst zawierający informacje liczbowe;
 5. do rozwiązywania zadań osadzonych w kontekście praktycznym stosuje poznaną wiedzę z zakresu arytmetyki i geometrii oraz nabyte umiejętności rachunkowe, a także własne poprawne metody;
 6. układa zadania i łamigłówki, rozwiązuje je;

Podstawa programowa kształcenia ogólnego dla szkół podstawowych – II etap edukacyjny – klasy IV-VI, informatyka:

I. Rozumienie, analizowanie i rozwiązywanie problemów. Uczeń:

2. formułuje i zapisuje w postaci algorytmów polecenia składające się na:
 1. rozwiązywanie problemów z życia codziennego i z różnych przedmiotów, np. liczenie średniej, pisemne wykonanie działań arytmetycznych, takich jak dodawanie i odejmowanie,
 3. w algorytmicznym rozwiązywaniu problemu wyróżnia podstawowe kroki: określenie problemu i celu do osiągnięcia, analiza sytuacji problemowej, opracowanie rozwiązania, sprawdzenie rozwiązania problemu dla przykładowych danych, zapisanie rozwiązania w postaci schematu lub programu.
- II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych. Uczeń:
 2. testuje na komputerze swoje programy pod względem zgodności z przyjętymi założeniami i ewentualnie je poprawia, objaśnia przebieg działania programów;
- IV. Rozwijanie kompetencji społecznych. Uczeń:
 1. uczestniczy w zespołowym rozwiązaniu problemu postępując się technologią taką jak: poczta elektroniczna, forum, wirtualne środowisko kształcenia, dedykowany portal edukacyjny;
 2. identyfikuje i docenia korzyści płynące ze współpracy nad wspólnym rozwiązywaniem problemów;

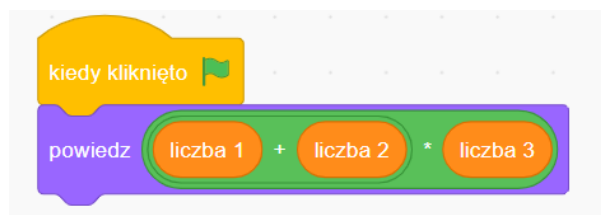
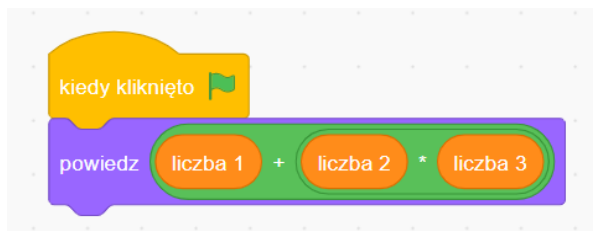
Wprowadzenie w tematykę i integracja grupy

Ucniowie grają w grę "Kto ma rację? A czy Z?" - <https://scratch.mit.edu/projects/210935897/>. Zastanawiają się, dlaczego litery A i Z udzielają różnych odpowiedzi oraz która z nich jest prawdziwa. Następnie sprawdzają, czy ich przypuszczenia są słuszne. Podpowiedź: o różnicy stanowi poprawnie lub błędnie zaprogramowana kolejność wykonywania działań. W Scratchu i mBlocku nie ma bloczków, które samodzielnie pozwalają na wykonanie działania, które ma więcej niż dwa elementy. Żeby takie działanie zapisać, trzeba łączyć takie bloczki w większe całości.

Część zasadnicza

Część 1

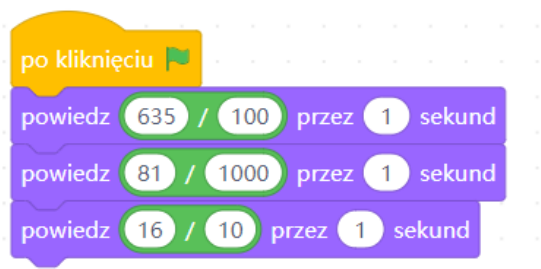
Nauczyciel omawia z uczniami, jak zapisywać złożone działania, zachowując kolejność wykonywania działań. Podpowiedź: kiedy używamy bloczków z grupy «wyrażenia», program zawsze najpierw będzie wykonywał działanie otoczone jedną pętlą.



Ucniowie w parach przygotowują dla siebie po jednym zadaniu, w których występują trzy lub cztery działania (w zależności od możliwości), np. $34-16:4+4=$ lub $12*4-16:4=$. Ucniowie wymieniają się zadaniami, a następnie piszą program, w którym duszek poda rozwiązanie. Muszą szczególnie zwrócić tu uwagę na prawidłowe zagnieżdżenie bloczków z działu wyrażenia. Następnie sprawdzają poprawność wykonania działań.

W kolejnej części lekcji nauczyciel wraca do pomysłów zapisanych na poprzedniej lekcji i ustala z uczniami, jakie elementy muszą jeszcze nauczyć się zapisywać. W przypadku klas 4 i 5 będą to ułamki, dodatkowo w klasie 6 będą to procenty.

Nauczyciel zapisuje ułamek dziesiętny na tablicy i ustala z uczniami, w jaki sposób można go zapisać jako działanie (mnożenie przez 10 lub wielokrotność 10). Warto powtórzyć takie zapisywanie, korzystając z ułamków z różną liczbą cyfr po przecinku. Następnie ucniowie zapisują trzy różne ułamki (najlepiej z różną liczbą cyfr po przecinku), wymieniają się nimi i piszą program, dzięki któremu duszek "powie" podane (np. wypisane wspólnie na tablicy) ułamki.



Po zakończeniu tego zadania ucniowie z rozsypanki ([Załącznik nr 1](#)) układają program podający dwa losowo wybrane ułamki dziesiętne. Dla klasy 6: nauczyciel omawia z uczniami, w jaki sposób można podobnie zapisać procenty (losuj od 1 do 100/100).

Część 2

Ucniowie wykorzystują zadanie z poprzedniej lekcji ("Ile godzin ma doba?") i dodają do skryptu działanie robota tak, aby po poprawnej odpowiedzi wyświetlił tęczę, obrócił się dookoła i odtworzył dźwięk "jej" (lub inny radosny dźwięk), a przy błędnej - odtworzył dźwięk "słaba bateria" (lub inny odpowiedni dźwięk).

Skrypt duszka na ekranie:



Skrypt robota:



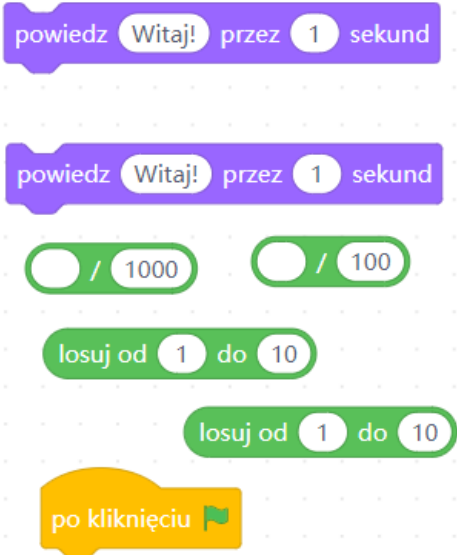

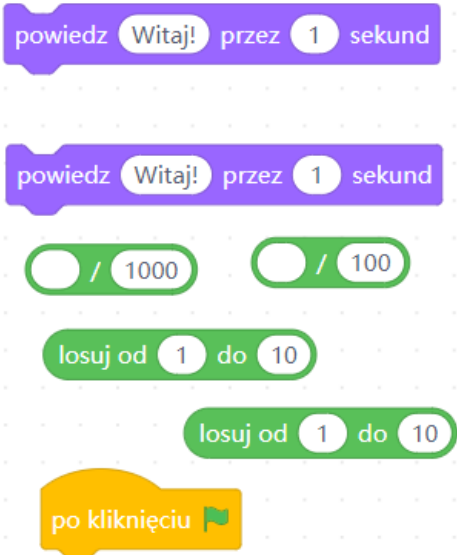
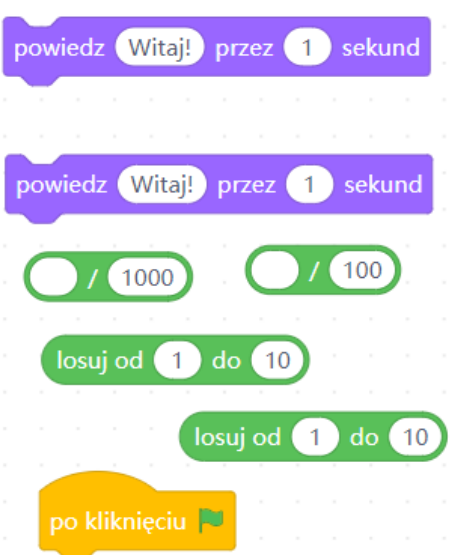
Podsumowanie i ewaluacja

Jako ostatni element uczniowie uruchamiają program "Kto powiedział to lepiej?"

<https://scratch.mit.edu/projects/211060770/> i zastanawiają się, który z duszków zadaje pytanie w czytelniejszy sposób. Oglądają skrypt duszka po prawej stronie i analizują, w jaki sposób zapisać zdanie, w które wplecione są zmienne.

Wskazówka: kiedy chcemy, żeby w wypowiedzianym tekście pojawiły się zmienne, trzeba użyć bloczka "połącz" (domyślnie jest to "połącz jabłko banan") z grupy "wyrażenia". Warto zwrócić uczniom uwagę na to, że na końcu i na początku tekstu znajdującego się przed lub po zmiennej trzeba wstawić spację, inaczej zmienna i tekst połączą się w jedną nieczytelną całość. Uczniowie mogą wraz z nauczycielem napisać proste zadanie, np. "W bibliotece było (losowo wybrane) książek dla dorosłych i (losowo wybrane) książek dla dzieci. Ile książek było razem w bibliotece?". Następnie zastanawiają się, jak napisać do niego skrypt i o jakich detalach należy pamiętać.

Załącznik nr 1

 <pre>po kliknięciu powiedz Witaj! przez 1 sekund losuj od 1 do 10 losuj od 1 do 100 losuj od 1 do 1000</pre>	 <pre>po kliknięciu powiedz Witaj! przez 1 sekund losuj od 1 do 10 losuj od 1 do 100 losuj od 1 do 1000</pre>
 <pre>po kliknięciu powiedz Witaj! przez 1 sekund losuj od 1 do 10 losuj od 1 do 100 losuj od 1 do 1000</pre>	 <pre>po kliknięciu powiedz Witaj! przez 1 sekund losuj od 1 do 10 losuj od 1 do 100 losuj od 1 do 1000</pre>

Robo-matematycy

Autorki: Joanna Płatkowska-Nęcka, Karolina Czerwińska

Lekcja 5:

Tworzymy grę matematyczną: część 1

Zajęcia, na których uczniowie poznają szczegółowo przepis na napisanie gry matematycznej, w którą zagrają przy pomocy mBota2, samodzielnie wykonanej planszy i napisanego programu. Uczniowie stworzą z nauczycielem schemat budowania zadań do gry. W grupach stworzą startowy ekran gry i pierwsze dwa zadania.

Cele lekcji:

Uczeń powinien:

- tworzyć planszę startową do gry matematycznej,
- ustalać kroki potrzebne do napisania skryptu wykonującego proste i złożone zadania matematyczne w tym z wyborem losowym,
- formułować polecenia składające się na wykonanie działań matematycznych,
- tworzyć skrypt sterujący robotem,
- decydować o trudności tworzonych rozwiązań.

Materiały pomocnicze:

- roboty mBot2,
- program mBlock,
- komputery stacjonarne lub laptopy,
- przygotowana wcześniej plansza dla mBota (inspiracja do obejrzenia w [Załączniku Nr 1](https://superkoderzy.pl/wp-content/uploads/2023/04/SK-robomat-2.m.p4) oraz tutaj: <https://superkoderzy.pl/wp-content/uploads/2023/04/SK-robomat-2.m.p4>),
- program "robo mat" pobrany ze strony: https://superkoderzy.pl/wp-content/uploads/2023/04/robo_mat-final.mblock.zip.

Pojęcia kluczowe:

→ działanie → wybór losowy → zasady → kolejność / sekwencja

Czas realizacji: 45 min.

Metody pracy:

- pogadanka,
- ćwiczenia praktyczne.

Treści programowe:

Podstawa programowa kształcenia ogólnego dla szkół podstawowych – II etap edukacyjny – klasy IV-VI, matematyka:

I. Liczby naturalne w dziesiątkowym układzie pozycyjnym. Uczeń:

1. zapisuje i odczytuje liczby naturalne wielocyfrowe;
3. porównuje liczby naturalne;

II. Działania na liczbach naturalnych. Uczeń:

1. dodaje i odejmuje w pamięci liczby naturalne dwucyfrowe lub większe, liczbę jednocyfrówą dodaje do dowolnej liczby naturalnej i odejmuje od dowolnej liczby naturalnej;
2. dodaje i odejmuje liczby naturalne wielocyfrowe sposobem pisemnym i za pomocą kalkulatora;

3. mnoży i dzieli liczbę naturalną przez liczbę naturalną jednocyfrówą, dwucyfrówą lub trzycyfrówą sposobem pisemnym, w pamięci (w najprostszych przykładach) i za pomocą kalkulatora (w trudniejszych przykładach);
5. stosuje wygodne dla siebie sposoby ułatwiające obliczenia, w tym przemienność i łączność dodawania i mnożenia oraz rozdzielność mnożenia względem dodawania;

IV. Ułamki zwykłe i dziesiętne. Uczeń:

2. przedstawia ułamek jako iloraz liczb naturalnych;
- V. Działania na ułamkach zwykłych i dziesiętnych. Uczeń:
 2. dodaje, odejmuje, mnoży i dzieli ułamki dziesiętne w pamięci (w przykładach najprostszych), pisemnie i za pomocą kalkulatora (w przykładach trudnych);

VI. Elementy algebry. Uczeń:

1. korzysta z nieskomplikowanych wzorów, w których występują oznaczenia literowe, opisuje wzór słowami;

XI. Obliczenia w geometrii. Uczeń:

1. oblicza obwód wielokąta o danych długościach boków;
2. oblicza pola: trójkąta, kwadratu, trapezu, przedstawionych na rysunku oraz w sytuacjach praktycznych;

XII. Obliczenia praktyczne. Uczeń:

2. w przypadkach osadzonych w kontekście praktycznym oblicza procent danej wielkości w stopniu trudności typu 50%, 20%, 10%;

XIV. Zadania tekstowe. Uczeń:

1. czyta ze zrozumieniem tekst zawierający informacje liczbowe;
5. do rozwiązywania zadań osadzonych w kontekście praktycznym stosuje poznaną wiedzę z zakresu arytmetyki i geometrii oraz nabyte umiejętności rachunkowe, a także własne poprawne metody;
6. układa zadania i łamigłówki, rozwiązuje je;

Podstawa programowa kształcenia ogólnego dla szkół podstawowych – II etap edukacyjny – klasy IV-VI, informatyka:

I. Rozumienie, analizowanie i rozwiązywanie problemów. Uczeń:

2. formułuje i zapisuje w postaci algorytmów polecenia składające się na:
 1. rozwiązanie problemów z życia codziennego i z różnych przedmiotów, np. liczenie średniej, pisemne wykonanie działań arytmetycznych, takich jak dodawanie i odejmowanie,

3. w algorytmicznym rozwiązywaniu problemu wyróżnia podstawowe kroki: określenie problemu i celu do osiągnięcia, analiza sytuacji problemowej, opracowanie rozwiązania, sprawdzenie rozwiązania problemu dla przykładowych danych, zapisanie rozwiązania w postaci schematu lub programu.

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych. Uczeń:

2. testuje na komputerze swoje programy pod względem zgodności z przyjętymi założeniami i ewentualnie je poprawia, objaśnia przebieg działania programów;
- IV. Rozwijanie kompetencji społecznych. Uczeń:

1. uczestniczy w zespołowym rozwiązaniu problemu posługując się technologią taką jak: poczta elektroniczna, forum, wirtualne środowisko kształcenia, dedykowany portal edukacyjny;
2. identyfikuje i docenia korzyści płynące ze współpracy nad wspólnym rozwiązywaniem problemów;

Wprowadzenie w tematykę i integracja grupy

Nauczyciel rozkłada planszę dla mBota2 (przygotowaną na szkoleniu lub przed tymi zajęciami) lub zaznacza miejsca kolejnych zadań na podłodze (odmierzając wcześniej odpowiednio równe odległości między polami), ustawia robota na starcie i uruchamia przygotowaną wcześniej grę matematyczną lub tę, którą można pobrać ze strony https://superkoderzy.pl/wp-content/uploads/2023/04/robo_mat-final.mblock.zip.

Klasa wraz z nauczycielem gra w grę, żeby zapoznać się ze sposobem jej działania (ekran startowy, cztery zadania, aktywność robota po podaniu odpowiedzi uzależniona od poprawności odpowiedzi, powrót do ekranu startowego po prawidłowej odpowiedzi, powtórzenie pytania po nieprawidłowej odpowiedzi, czas liczony na timerze wyświetlanym na ekranie mBota2). Istotne jest, żeby pokazać wszystkie cztery zadania i wyjaśnić, co dzieje się przy odpowiedziach błędnych i poprawnych oraz ustalić, że grę wygrywa ten, kto skończy grę w najkrótszym czasie.

Część zasadnicza

Na tym etapie rozpoczynamy pisanie analogicznego skryptu, który umożliwi nam stworzenie gry matematycznej. Na czym będzie ona polegała?

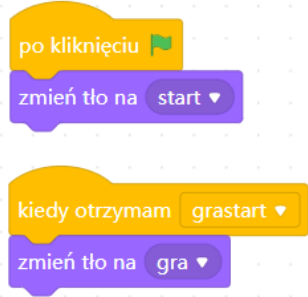
- Uczniowie dzielą się na 4-osobowe drużyny.
- Każda z drużyn przygotowuje grę, w którą zagrają pozostałe drużyny. Im trudniejsza gra, tym lepiej, ponieważ wygrywa ta drużyna, która najszybciej przejdzie grę przygotowaną przez przeciwnika.
- Na grę składa się:
 - plansza, po której jeździ robot. Wykonujemy ją np. z połączonych flipchartów lub umieszczamy oznaczenia pól w ustalonych odległościach na podłodze. Rysujemy na niej tor z polami połączonymi liniami, po których porusza się robot, na których kryją się zadania matematyczne do rozwiązania. Liczbę pól ustala nauczyciel wraz z grupą w zależności od ich możliwości i chęci - może to być stała liczba (np. 4 - dla takiej liczby jest przygotowana rubryka do oceny).
 - robot mBot2. Po pierwsze mBot2 działa jak stoper - odlicza czas od początku do końca gry. Po drugie po udzieleniu prawidłowej odpowiedzi robot przemieszcza się po planszy do następnego zadania (jest pionkiem). Robot kontroluje, czy zadania matematyczne zostały poprawnie wykonane i odpowiednio wydaje dźwięki/świeci światłami. Uczniowie mogą zaproponować inne działania robota, ważne, żeby były spójne i jasne.
 - zadania matematyczne, napisane w środowisku mBlock. Jako inspirację można wykorzystać dotychczas wykonywane zadania oraz zadania udostępnione w Studiu: <https://scratch.mit.edu/studios/4907086/>. Do każdego z zadań dodajemy skrypt stworzony podczas Lekcji nr 4, w którym programowaliśmy reakcje robota po otrzymaniu danego komunikatu (poprawnej lub niepoprawnej odpowiedzi).

Dla zobrazowania tych wyjaśnień można wyświetlić zdjęcia z [Załącznika nr 1](#) oraz filmik <https://superkoderzy.pl/wp-content/uploads/2023/04/SK-robomat-2.mp4>

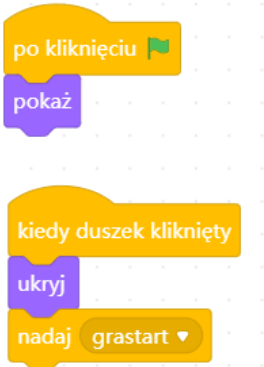
Część 1 - ekran startowy

Nauczyciel wraz z klasą zastanawia się, jakie elementy muszą pojawić się na ekranie startowym, żeby zadziałał i jakie skrypty należy napisać w tym celu. Warto zaplanować kolorowe tło, na którym pojawi się duszek w postaci napisu START. Po jego kliknięciu na ekranie mogą pojawić się duszki w postaci numerów od 1 do 4. Po kliknięciu każdego z nich, wszystkie znikają i pojawia się odpowiednie zadanie.

Skrypt tła startowego



Skrypt duszka Start



Skrypt dla wybranego numeru (w tym przypadku zadania nr 1)



Część 2 - zaprogramowanie stopera (timer)

Żeby robot wyświetlał czas odmierzany od momentu wciśnięcia napisu Start, należy ustawić pętlę warunkową zawsze i bloczek wyświetl z działu Ekran, w który należy wstawić guzik stoper z działu Czujniki. Ważne, żeby po naciśnięciu zielonej flagi stoper się resetował (można porównać to do zadań z pisakiem, które uczniowie wykonują w ramach scenariuszy dotyczących Scratcha).

Można dodatkowo stworzyć listę czasów (żeby nie trzeba było ich zapisywać na kartce). Wtedy należy utworzyć listę (w tym samym dziale co zmienne), nazwać ją Czasy lub wyniki i skorzystać z bloczka dodaj [rzecz] do [nazwa listy]. W ten sposób każdy kolejny czas zostanie dodany do listy po udzieleniu odpowiedzi na ostatnie pytanie.



Część 3 i 4 - dodawanie i odejmowanie ułamków zwykłych i liczb całkowitych oraz mnożenie i dzielenie liczb całkowitych

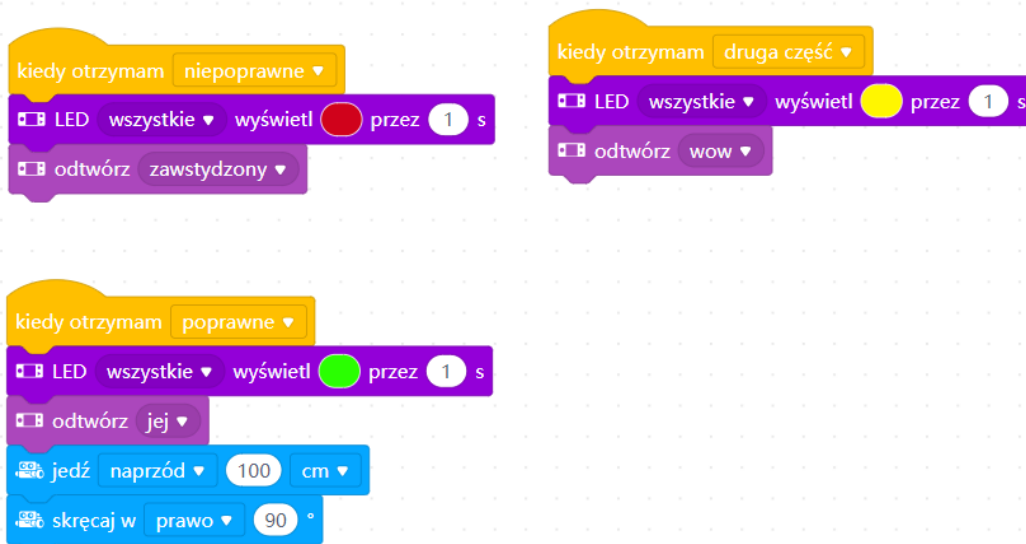
Każde z zadań musi składać się z następujących elementów:

- bloczek pokaż
- pętla jeżeli to – w przeciwnym razie

Dla wszystkich zadań będzie też jeden wspólny skrypt robota, w którym znajdzie się:

- określenie, co ma dzieć się po prawidłowej odpowiedzi (nasza propozycja: zielone światło robota, dźwięk i przejazd do następnego zadania w przypadku zadań pojedynczych; żółte światło i odpowiedni dźwięk w przypadku pierwszej części zadań złożonych z dwóch części),
- określenie, co ma dzieć się po nieprawidłowej odpowiedzi (nasza propozycja: czerwone światło robota i odpowiedni dźwięk),

Skrypty dla robota:



Uczniowie wspólnie z nauczycielem ustalają, jak powinien wyglądać skrypt dla dwóch pierwszych zadań: **dodawanie i odejmowanie ułamków zwykłych i liczb całkowitych (dodawanie i odejmowanie dwóch ułamków niekiedy działa niepoprawnie ze względu na ukryte miejsca dziesiętne) oraz mnożenie i dzielenie liczb całkowitych**. Nauczyciel może zdecydować, czy uczniowie będą pracować na zadaniach w wersji podstawowej (podane wartości), czy rozszerzonej (wartości wybierane losowo). Może również wybór zostawić uczniom lub ustalić z nimi, że część zadań ma być w wersji podstawowej, a część w rozszerzonej.

Wersja podstawowa przykładowego zadania:

```
po kliknięciu
ukryj

kiedy otrzymam 4start
pokaż
zmień kostium na Bug

zawsze
zapytaj Ile wynosi suma liczb 6.208 i 3.05? i czekaj
jeżeli odpowiedź = 9.258 to
  nadaj druga część
  czekaj 5 sekund
  powiedz Brawo! Możesz przejść do następnego pytania. przez 2 sekund
  zmień kostium na Bug2
  zawsze
  zapytaj Ile wynosi różnica liczb 3.05 i 6.208? i czekaj
  jeżeli odpowiedź = 3.158 to
    nadaj poprawne
    czekaj 5 sekund
    powiedz Doskonała odpowiedź. przez 2 sekund
    nadaj 4koniec
    ukryj
    stop ten skrypt
  w przeciwnym razie
    nadaj niepoprawne
    czekaj 3 sekund
    powiedz Spróbuj policzyć jeszcze raz. przez 2 sekund
    czekaj 1 sekund
w przeciwnym razie
  nadaj niepoprawne
  czekaj 3 sekund
  powiedz Spróbuj policzyć jeszcze raz. przez 2 sekund
  czekaj 1 sekund
```

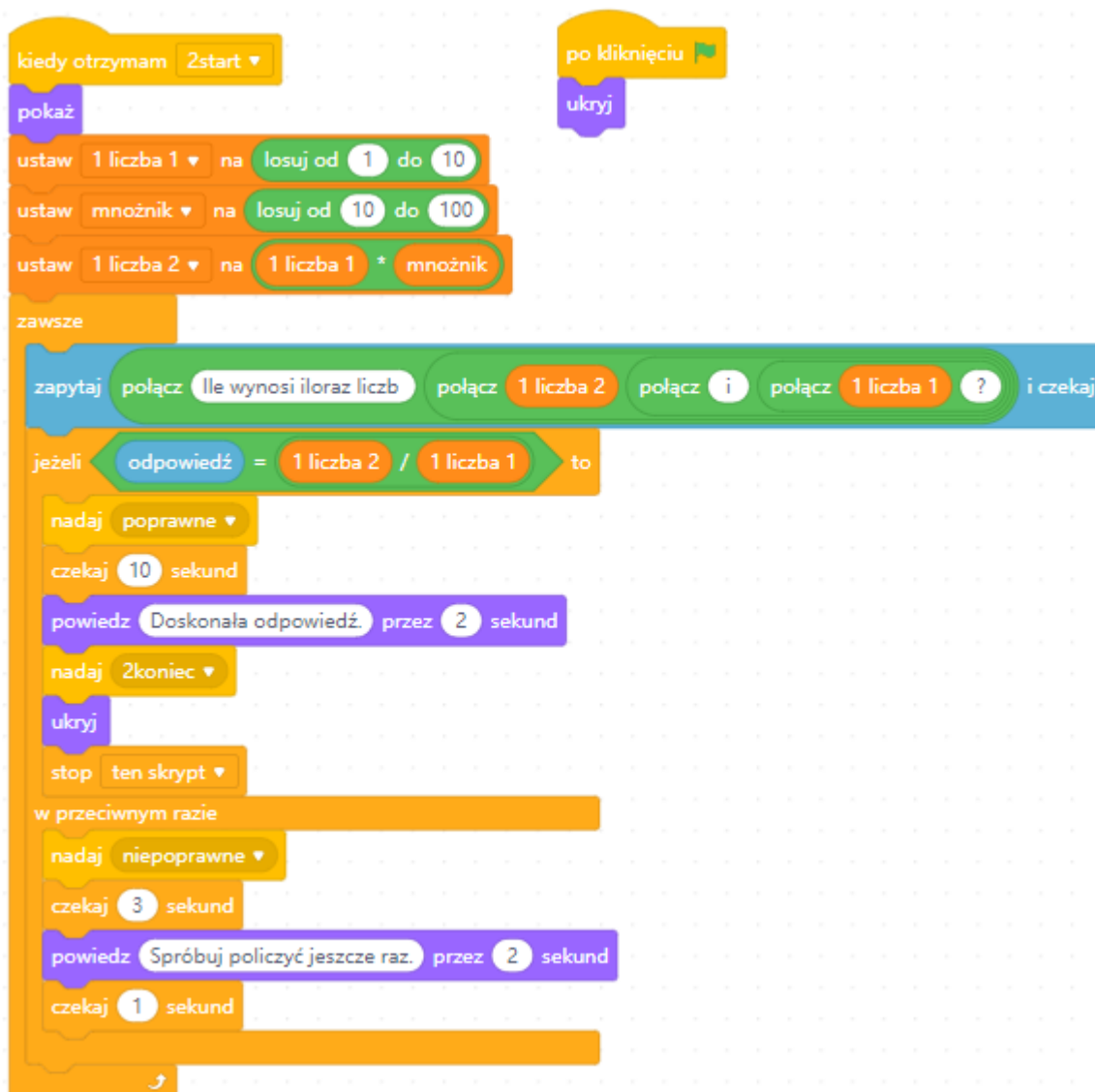

Wersja rozszerzona tego samego zadania:



#SuperKoderzy / Robo-matematycy / Tworzymy grę matematyczną: część 1

W przypadku wersji rozszerzonej przy dodawaniu i odejmowaniu trzeba pamiętać, że liczba 1 musi być większa od liczby 2, żeby uniknąć liczb ujemnych.

W przypadku wersji rozszerzonej mnożenia i dzielenia, trzeba pamiętać, że liczba pierwsza musi być wielokrotnością liczby drugiej, według sposobu pokazanego na lekcji 3.



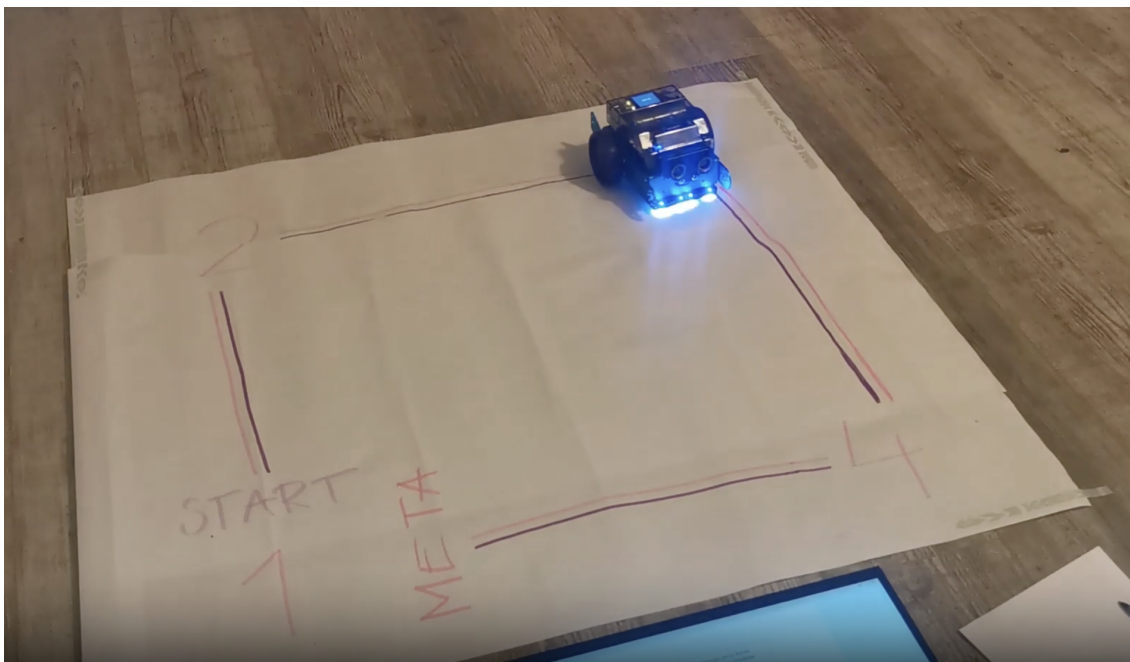
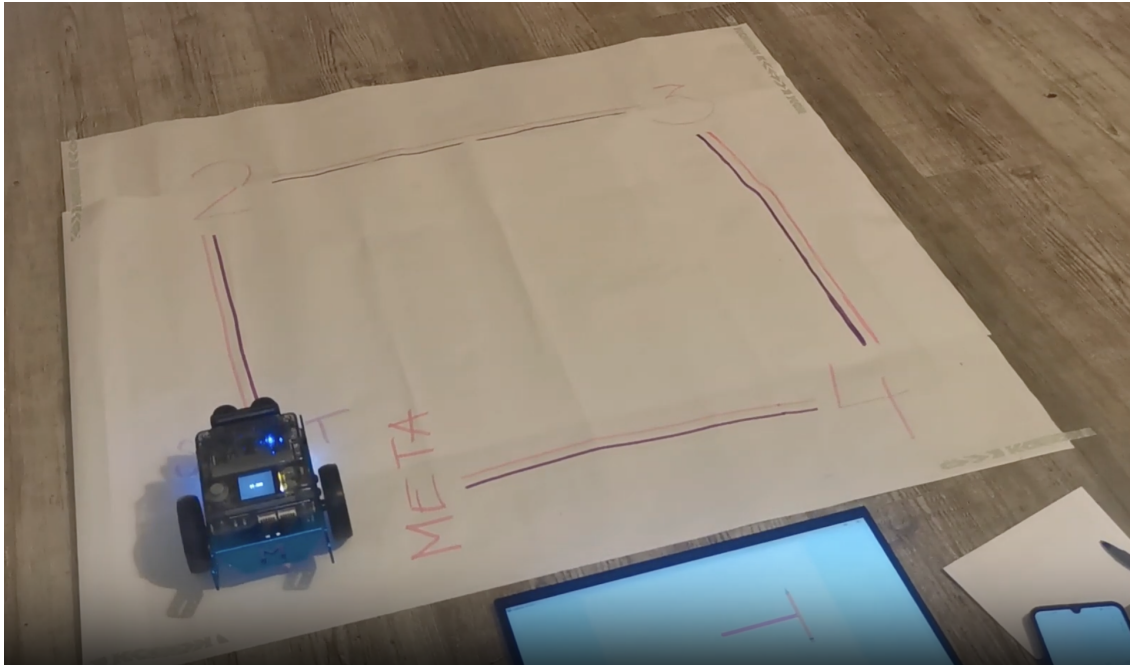
Uczniowie pracują w grupach, żeby stworzyć skrypt dla dwóch pierwszych zadań. W zależności od możliwości czasowych mogą dodać efekty zarówno na ekranie startowym, jak i w działaniu robota.

Podsumowanie i ewaluacja

Grupy zamieniają się miejscami i próbują wykonać zadania drugiej grupy. Następnie przekazują sobie informację zwrotną uwzględniając elementy, które działały poprawnie i takie, nad którymi grupa musi jeszcze popracować.

Nauczyciel przypomina uczniom o zapisaniu plików tak, by były one dostępne na następnych zajęciach.

Załącznik nr 1



Robo-matematycy

Autorki: Joanna Płatkowska-Nęcka, Karolina Czerwińska

Lekcja 6:

Tworzymy grę matematyczną: część 2

Zajęcia, na których uczniowie tworzą kolejne dwa zadania do gry. Korzystają z informacji zwrotnej z poprzednich zajęć, żeby wprowadzić ewentualne poprawki w dwóch pierwszych zadaniach.

Cele lekcji:

Uczeń powinien:

- tworzyć planszę końcową do gry matematycznej,
- ustalać kroki potrzebne do napisania skryptu wykonującego proste i złożone zadania matematyczne w tym z wyborem losowym,
- formułować polecenia składające się na wykonanie działań matematycznych,
- tworzyć skrypt sterujący robotem,
- decydować o trudności tworzonych rozwiązań.

Materiały pomocnicze:

- roboty mBot2,
- program mBlock,
- komputery stacjonarne lub laptopy,
- program "lekcja5.sb2" pobrany ze strony:
https://superkoderzy.pl/wp-content/uploads/2023/04/robo_m at-final.mblock.zip

Pojęcia kluczowe:

→ działanie → wybór losowy → efekt → kolejność / sekwencja

Czas realizacji: 45 min.

Metody pracy:

- pogadanka,
- ćwiczenia praktyczne.

Treści programowe:

Podstawa programowa kształcenia ogólnego dla szkół podstawowych – II etap edukacyjny – klasy IV-VI, matematyka:

I. Liczby naturalne w dziesiętkowym układzie pozycyjnym. Uczeń:

1. zapisuje i odczytuje liczby naturalne wielocyfrowe;
3. porównuje liczby naturalne;

II. Działania na liczbach naturalnych. Uczeń:

1. dodaje i odejmuje w pamięci liczby naturalne dwucyfrowe lub większe, liczbę jednocyfrową dodaje do dowolnej liczby naturalnej i odejmuje od dowolnej liczby naturalnej;
2. dodaje i odejmuje liczby naturalne wielocyfrowe sposobem pisemnym i za pomocą kalkulatora;
3. mnoży i dzieli liczbę naturalną przez liczbę naturalną jednocyfrową, dwucyfrową lub trzycyfrową sposobem pisemnym, w pamięci (w najprostszymi przykładach) i za pomocą kalkulatora (w trudniejszych przykładach);
5. stosuje wygodne dla siebie sposoby ułatwiające obliczenia, w tym przemienność i łączność dodawania i mnożenia oraz rozdzielność mnożenia względem dodawania;

IV. Ułamki zwykłe i dziesiętne. Uczeń:

2. przedstawia ułamek jako iloraz liczb naturalnych;
- V. Działania na ułamkach zwykłych i dziesiętnych. Uczeń:
 2. dodaje, odejmuje, mnoży i dzieli ułamki dziesiętne w pamięci (w przykładach najprostszymi), pisemnie i za pomocą kalkulatora (w przykładach trudnych);
- VI. Elementy algebry. Uczeń:

1. korzysta z nieskomplikowanych wzorów, w których występują oznaczenia literowe, opisuje wzór słowami;

XI. Obliczenia w geometrii. Uczeń:

1. oblicza obwód wielokąta o danych długościach boków;
2. oblicza pola: trójkąta, kwadratu, trapezu, przedstawionych na rysunku oraz w sytuacjach praktycznych;

XII. Obliczenia praktyczne. Uczeń:

2. w przypadkach osadzonych w kontekście praktycznym oblicza procent danej wielkości w stopniu trudności typu 50%, 20%, 10%;
- XIV. Zadania tekstowe. Uczeń:

1. czyta ze zrozumieniem tekst zawierający informacje liczbowe;
5. do rozwiązywania zadań osadzonych w kontekście praktycznym stosuje poznaną wiedzę z zakresu arytmetyki i geometrii oraz nabyte umiejętności rachunkowe, a także własne poprawne metody;
6. układa zadania i łamigłówki, rozwiązuje je;

Podstawa programowa kształcenia ogólnego dla szkół podstawowych – II etap edukacyjny – klasy IV-VI, informatyka:

I. Rozumienie, analizowanie i rozwiązywanie problemów. Uczeń:

2. formułuje i zapisuje w postaci algorytmów polecenia składające się na:

1. rozwiązanie problemów z życia codziennego i z różnych przedmiotów, np. liczenie średniej, pisemne wykonanie działań arytmetycznych, takich jak dodawanie i odejmowanie,
3. w algorytmicznym rozwiązywaniu problemu wyróżnia podstawowe kroki: określenie problemu i celu do osiągnięcia, analiza sytuacji problemowej, opracowanie rozwiązania, sprawdzenie rozwiązania problemu dla przykładowych danych, zapisanie rozwiązania w postaci schematu lub programu.

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych. Uczeń:

2. testuje na komputerze swoje programy pod względem zgodności z przyjętymi założeniami i ewentualnie je poprawia, objaśnia przebieg działania programów;

IV. Rozwijanie kompetencji społecznych. Uczeń:

1. uczestniczy w zespołowym rozwiązaniu problemu posługując się technologią taką jak: poczta elektroniczna, forum, wirtualne środowisko kształcenia, dedykowany portal edukacyjny;
2. identyfikuje i docenia korzyści płynące ze współpracy nad wspólnym rozwiązywaniem problemów;

Wprowadzenie w tematykę i integracja grupy

Nauczyciel informuje, że jedno z zadań, które zostaną stworzone na tej lekcji, będzie **zadaniem tekstowym**. Uczniowie w grupach piszą zadanie, które umieszczają w swojej grze.

Pomocne może być stworzenie z uczniami kilku zadań podobnych do zaproponowanych w przykładach z lekcji 3 - <https://scratch.mit.edu/studios/4907086/>

```
when green flag clicked
  show
  go to x: 0 y: 0
  move pen to tip
  set pen color to pink
  set pen size to 10
  move 1 sec to x: 160 y: 0
  move 1 sec to x: 120 y: 100
  move 1 sec to x: 40 y: 100
  move 1 sec to x: 0 y: 0
  ask "Ile centymetrów długości ma dolna podstawa Twojego trapezu?" and wait
  set "dolna podstawa" to answer
  ask "Ile centymetrów długości ma górna podstawa Twojego trapezu?" and wait
  set "górna podstawa" to answer
  ask "Ile centymetrów długości ma wysokość Twojego trapezu?" and wait
  set "wysokość" to answer
  always
    ask "Ile centymetrów kwadratowych ma pole Twojego trapezu?" and wait
    if "answer" = 0.5 * "wysokość" * "górna podstawa" + "dolna podstawa" then
      say "poprawne"
      wait 10 sec
      say "Brawo!" for 2 sec
      erase everything
      hide
      say "1koniec"
      stop this script
    otherwise
      say "niepoprawne"
      wait 2 sec
      say "Sprawdź czy dobrze zastosowałeś wzór na pole powierzchni trapezu." for 2 sec
      wait 1 sec
```

Część zasadnicza

Podobnie jak na poprzedniej lekcji nauczyciel wraz z uczniami tworzy schemat pisania skryptu kolejnych zadań. Tym razem skupiamy się na **zadaniach z geometrii i zadaniach tekstowych**.

W przypadku zadania z geometrii ważne jest, żeby zaznaczyć, że rysunek jest jedynie ilustracją do zadania, a nie przedstawieniem faktycznej figury z zadania. Dobrze również zwrócić uwagę na to, że wartości liczbowe nie są tu podane w centymetrach, więc bok kwadratu określony jako 15 kroków nie będzie miał 15 cm i będzie mało widoczny na ekranie (wartości około 150 będą bardziej odpowiednie).

W klasie szóstej warto wprowadzić także zadania z obliczaniem pola trapezu (zgodnie z podstawą programową). Należy tu pamiętać o kolejności wykonywania działań.

W zaproponowanych przykładach znajdujących się w studio Robo-matematycy

<https://scratch.mit.edu/studios/4907086/> znalazły się następujące zadania tekstowe:

- Klasa 4: Alicja ma (losowo) zł. Artur ma o (losowo) zł więcej niż Alicja. Ile pieniędzy mają razem dzieci?
- Klasa 5: Na parkingu zaparkowało (losowo) samochodów osobowych i (losowo) samochodów ciężarowych. Razem zajęły one połowę wszystkich miejsc na tym parkingu. Ile miejsc ma ten parking?
- Klasa 6: Cenę gitary obniżono o (losowo) %, a następnie podwyższono o (losowo) %. Ile teraz kosztuje ta gitara?

Oczywiście w zależności od możliwości uczniów można wykorzystać zadania z klasy programowo niższej lub wyższej, czy też stworzyć swoje własne, zupełnie inne zadania.

Uczniom, którzy szczególnie dobrze radzą sobie ze Scratchem, można zaproponować zwiększenie liczby zadań.

Podsumowanie i ewaluacja

Grupy zamieniają się miejscami i próbują wykonać zadania drugiej grupy. Następnie przekazują sobie informację zwrotną uwzględniając elementy, które działały poprawnie i takie, nad którymi grupa musi jeszcze popracować.

Robo-matematycy

Autorki: Joanna Płatkowska-Nęcka, Karolina Czerwińska

Lekcja 7:

Tworzymy grę matematyczną: część 3

Zajęcia, na których uczniowie tworzą planszę swojej gry. Jest to też czas na przeprowadzanie prób działania gry, modyfikowanie, korygowanie błędów i wprowadzanie dodatkowych efektów do gry.

Cele lekcji:

Uczeń powinien:

- tworzyć planszę do gry, po której będzie poruszał się robot,
- łączyć skrypt z funkcjonalnościami robota,
- urozmaicać grę poprzez dodawanie efektów na ekranie i w działaniu robota.

Materiały pomocnicze:

- roboty mBot2
- program mBlock,
- komputery stacjonarne lub laptopy,
- duże arkusze papieru,
- pisaki, kredki, flamastry.

Pojęcia kluczowe:

→ działanie → wybór losowy → efekt → kolejność / sekwencja

Czas realizacji: 45 min.

Metody pracy:

- pogadanka,
- ćwiczenia praktyczne.

Treści programowe:

Podstawa programowa kształcenia ogólnego dla szkół podstawowych – II etap edukacyjny – klasy IV-VI, matematyka:

I. Liczby naturalne w dziesiętkowym układzie pozycyjnym. Uczeń:

1. zapisuje i odczytuje liczby naturalne wielocyfrowe;
3. porównuje liczby naturalne;

II. Działania na liczbach naturalnych. Uczeń:

1. dodaje i odejmuje w pamięci liczby naturalne dwucyfrowe lub większe, liczbę jednocyfrówą dodaje do dowolnej liczby naturalnej i odejmuje od dowolnej liczby naturalnej;
2. dodaje i odejmuje liczby naturalne wielocyfrowe sposobem pisemnym i za pomocą kalkulatora;
3. mnoży i dzieli liczbę naturalną przez liczbę naturalną jednocyfrówą, dwucyfrówą lub trzycyfrówą sposobem pisemnym, w pamięci (w najprostszych przykładach) i za pomocą kalkulatora (w trudniejszych przykładach);
5. stosuje wygodne dla siebie sposoby ułatwiające obliczenia, w tym przemienność i łączność dodawania i mnożenia oraz rozdzielność mnożenia względem dodawania;

IV. Ułamki zwykłe i dziesiętne. Uczeń:

2. przedstawia ułamek jako iloraz liczb naturalnych;

V. Działania na ułamkach zwykłych i dziesiętnych. Uczeń:

2. dodaje, odejmuje, mnoży i dzieli ułamki dziesiętne w pamięci (w przykładach najprostszych), pisemnie i za pomocą kalkulatora (w przykładach trudnych);

VI. Elementy algebry. Uczeń:

1. korzysta z nieskomplikowanych wzorów, w których występują oznaczenia literowe, opisuje wzór słowami;

XI. Obliczenia w geometrii. Uczeń:

1. oblicza obwód wielokąta o danych długościach boków;
2. oblicza pola: trójkąta, kwadratu, trapezu, przedstawionych na rysunku oraz w sytuacjach praktycznych;

XII. Obliczenia praktyczne. Uczeń:

2. w przypadkach osadzonych w kontekście praktycznym oblicza procent danej wielkości w stopniu trudności typu 50%, 20%, 10%;

XIV. Zadania tekstowe. Uczeń:

1. czyta ze zrozumieniem tekst zawierający informacje liczbowe;
5. do rozwiązywania zadań osadzonych w kontekście praktycznym stosuje poznaną wiedzę z zakresu arytmetyki i geometrii oraz nabyte umiejętności rachunkowe, a także własne poprawne metody;
6. układa zadania i łamigłówki, rozwiązuje je;

Podstawa programowa kształcenia ogólnego dla szkół podstawowych – II etap edukacyjny – klasy IV-VI, informatyka:

I. Rozumienie, analizowanie i rozwiązywanie problemów. Uczeń:

2. formułuje i zapisuje w postaci algorytmów polecenia składające się na:

1. rozwiązanie problemów z życia codziennego i z różnych przedmiotów, np. liczenie średniej, pisemne wykonanie działań arytmetycznych, takich jak dodawanie i odejmowanie,
3. w algorytmicznym rozwiązywaniu problemu wyróżnia podstawowe kroki: określenie problemu i celu do osiągnięcia, analiza sytuacji problemowej, opracowanie rozwiązania, sprawdzenie rozwiązania problemu dla przykładowych danych, zapisanie rozwiązania w postaci schematu lub programu.

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych. Uczeń:

2. testuje na komputerze swoje programy pod względem zgodności z przyjętymi założeniami i ewentualnie je poprawia, objaśnia przebieg działania programów;

IV. Rozwijanie kompetencji społecznych. Uczeń:

1. uczestniczy w zespołowym rozwiązaniu problemu posługując się technologią taką jak: poczta elektroniczna, forum, wirtualne środowisko kształcenia, dedykowany portal edukacyjny;
2. identyfikuje i docenia korzyści płynące ze współpracy nad wspólnym rozwiązywaniem problemów;

Wprowadzenie w tematykę i integracja grupy

Nauczyciel ponownie pokazuje uczniom zdjęcia przykładowej planszy do gry lub wcześniej przygotowaną planszę. Uczniowie mogą oczywiście tylko rozkładać pola bez tworzenia papierowej planszy, jednak wtedy trzeba szczególnie pamiętać o odległościach między oznaczeniami. Nauczyciel rozmawia z uczniami o tym, jakie warunki musi spełnić plansza, żeby grało się sprawnie i przyjemnie:

- musi być jasno zaznaczone miejsce startu i mety,
- na planszy muszą znaleźć się jasno oznaczone pola zadań,
- droga między zadaniami musi być odpowiednio szeroka, żeby robot mógł się w niej zmieścić,
- zakręty powinny być zaplanowane w ten sposób, żeby skręcanie nie było zbyt uciążliwe. Nauczyciel omawia z uczniami listę kontrolną ([Załącznik nr 1](#)) lub tworzy podobną z uczniami.

Część zasadnicza

Uczniowie w grupach tworzą gry pamiętając o wszystkich warunkach zawartych w checkliście.

- Na planszy lub na ekranie powinna pojawić się krótka instrukcja.
- Na planszy muszą znajdować się pola startu, mety i poszczególnych zadań.
- Gra na ekranie powinna mieć ekran startowy i ekran, na którym znajdują się duszki poszczególnych zadań.
- Każde z zadań ma pojawiać się po kliknięciu odpowiedniego duszka i znikać po podaniu prawidłowej odpowiedzi.
- Warto ustalić podobny sposób reakcji robota na odpowiedzi poprawne i niepoprawne (np. zielone światło, dźwięk i jazda do przodu w przypadku odpowiedzi poprawnej i czerwone światło i dźwięk w przypadku odpowiedzi niepoprawnej).
- Gra powinna zawierać kilka różnych zadań (dotyczących różnych tematów, np. dodawanie, odejmowanie, mnożenie dzielenie, ułamki dziesiętne, pola lub obwody figur, zadania tekstowe).
- Robot powinien poprawnie liczyć czas przejścia gry. Na ekranie może również pojawiać się lista wszystkich kolejnych czasów.

Podsumowanie i ewaluacja

Uczniowie dokonują samooceny korzystając z listy kontrolnej.

Załącznik nr 1

O czym należy pamiętać, aby stworzyć naprawdę dobrą grę matematyczną?



Na planszy lub na ekranie powinna pojawić się krótka instrukcja.



Na planszy muszą znajdować się pola startu, mety i poszczególnych zadań.



Gra na ekranie powinna mieć planszę startową i taką, na której znajdą się duszki poszczególnych zadań.



Każde z zadań ma pojawiać się po kliknięciu odpowiedniego duszka i znikać po podaniu prawidłowej odpowiedzi.



Warto ustalić podobny sposób reakcji robota na odpowiedzi poprawne i niepoprawne (np. zielone światło, dźwięk i jazda do przodu w przypadku odpowiedzi poprawnej i czerwone światło, dźwięk i brak ruchu w przypadku odpowiedzi niepoprawnej).



Gra powinna zawierać kilka różnych zadań (dotyczących różnych tematów, np. dodawanie, odejmowanie, mnożenie dzielenie, ułamki dziesiętne, pola lub obwody figur, zadania tekstowe).



Robot powinien poprawnie liczyć czas przejścia gry. Na ekranie może również pojawiać się lista wszystkich kolejnych czasów.

Robo-matematycy

Autorki: Joanna Płatkowska-Nęcka, Karolina Czerwińska

Lekcja 8:

Wielki matematyczny turniej – gry!

Zajęcia, na których uczniowie grają w gry stworzone przez przeciwne drużyny. Sprawdzają, komu uda się przejść daną grę w najkrótszym czasie. Dają sobie informację zwrotną i poprawiają ewentualne błędy. Ustalają, w jaki sposób udostępnią większemu gronu uczniów stworzone przez siebie gry.

Cele lekcji:

Uczeń powinien:

- zagrać w gry innych grup,
- rozwiązać zadania znajdujące się w grach innych grup,
- udzielić informacji zwrotnej innym grupom, która określa mocne strony i obszary do poprawy stworzonej gry.

Materiały pomocnicze:

- roboty mBot2
- program mBlock,
- komputery stacjonarne lub laptopy,
- gry stworzone przez uczniów.

Pojęcia kluczowe:

→ płynność → plansza

Czas realizacji: 90 min.

Metody pracy:

- ćwiczenia praktyczne.

Treści programowe:

Podstawa programowa kształcenia ogólnego dla szkół podstawowych – II etap edukacyjny – klasy IV-VI, matematyka:

I. Liczby naturalne w dziesiętkowym układzie pozycyjnym. Uczeń:

1. zapisuje i odczytuje liczby naturalne wielocyfrowe;
3. porównuje liczby naturalne;
- II. Działania na liczbach naturalnych. Uczeń:
 1. dodaje i odejmuje w pamięci liczby naturalne dwucyfrowe lub większe, liczbę jednocyfrówą dodaje do dowolnej liczby naturalnej i odejmuje od dowolnej liczby naturalnej;
 2. dodaje i odejmuje liczby naturalne wielocyfrowe sposobem pisemnym i za pomocą kalkulatora;
 3. mnoży i dzieli liczbę naturalną przez liczbę naturalną jednocyfrówą, dwucyfrową lub trzycyfrową sposobem pisemnym, w pamięci (w najprostszych przykładach) i za pomocą kalkulatora (w trudniejszych przykładach);
 5. stosuje wygodne dla siebie sposoby ułatwiające obliczenia, w tym przemienność i łączność dodawania i mnożenia oraz rozdzielność mnożenia względem dodawania;

IV. Ułamki zwykłe i dziesiętne. Uczeń:

2. przedstawia ułamek jako iloraz liczb naturalnych;
- V. Działania na ułamkach zwykłych i dziesiętnych. Uczeń:
 2. dodaje, odejmuje, mnoży i dzieli ułamki dziesiętne w pamięci (w przykładach najprostszych), pisemnie i za pomocą kalkulatora (w przykładach trudnych);

VI. Elementy algebry. Uczeń:

1. korzysta z nieskomplikowanych wzorów, w których występują oznaczenia literowe, opisuje wzór słowami;

XI. Obliczenia w geometrii. Uczeń:

1. oblicza obwód wielokąta o danych długościach boków;
2. oblicza pola: trójkąta, kwadratu, trapezu, przedstawionych na rysunku oraz w sytuacjach praktycznych;

XII. Obliczenia praktyczne. Uczeń:

2. w przypadkach osadzonych w kontekście praktycznym oblicza procent danej wielkości w stopniu trudności typu 50%, 20%, 10%;

XIV. Zadania tekstowe. Uczeń:

1. czyta ze zrozumieniem tekst zawierający informacje liczbowe;
5. do rozwiązywania zadań osadzonych w kontekście praktycznym stosuje poznaną wiedzę z zakresu arytmetyki i geometrii oraz nabyte umiejętności rachunkowe, a także własne poprawne metody;
6. układa zadania i łamigłówki, rozwiązuje je;

Podstawa programowa kształcenia ogólnego dla szkół podstawowych – II etap edukacyjny – klasy IV-VI, informatyka:

I. Rozumienie, analizowanie i rozwiązywanie problemów. Uczeń:

2. formułuje i zapisuje w postaci algorytmów polecenia składające się na:

1. rozwiązanie problemów z życia codziennego i z różnych przedmiotów, np. liczenie średniej, pisemne wykonanie działań arytmetycznych, takich jak dodawanie i odejmowanie,
3. w algorytmicznym rozwiązywaniu problemu wyróżnia podstawowe kroki: określenie problemu i celu do osiągnięcia, analiza sytuacji problemowej, opracowanie rozwiązania, sprawdzenie rozwiązania problemu dla przykładowych danych, zapisanie rozwiązania w postaci schematu lub programu.

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych. Uczeń:

2. testuje na komputerze swoje programy pod względem zgodności z przyjętymi założeniami i ewentualnie je poprawia, objaśnia przebieg działania programów;

IV. Rozwijanie kompetencji społecznych. Uczeń:

1. uczestniczy w zespołowym rozwiązaniu problemu postugując się technologią taką jak: poczta elektroniczna, forum, wirtualne środowisko kształcenia, dedykowany portal edukacyjny;
2. identyfikuje i docenia korzyści płynące ze współpracy nad wspólnym rozwiązywaniem problemów;

Wprowadzenie w tematykę i integracja grupy

Nauczyciel przypomina uczniom zasady fair play, które obowiązują również w stworzonej przez nich grze. Wspólnie ustalają kto w jakiej kolejności zagra w poszczególne gry. Nauczyciel informuje również uczniów, że po każdej grze ocenią ją, korzystając z kryteriów oceny i formularza z [Załączniku nr 1](#).

Część zasadnicza

Uczniowie grają w gry stworzone przez inne grupy. Jeśli w danej grze nie ma stworzonej listy czasów, zapisują na kartce czas przejścia rozgrywki poszczególnych grup, żeby określić najkrótszy czas i wyłonić zwycięzców. Formułują także informację zwrotną dla przeciwnych drużyn oraz zastanawiają się nad oceną swojej gry - pomoże im w tym formularz z [Załącznika nr 1](#). Nauczyciel również ocenia pracę grup.

Nauczyciel z uczniami ustalają, w jaki sposób mogą udostępnić gry szerszej grupie uczniów – w czasie lekcji informatyki, zajęć świetlicowych?

Na koniec uczniowie mogą indywidualnie stworzyć mini wersję takiej gry w Scratchu (czyli już bez wykorzystania mBota2). Wykorzystają do tego wszystkie umiejętności zdobyte w czasie poprzednich siedmiu lekcji. W zależności od dostępnego czasu może to być dowolna liczba zadań połączonych planszą startową.

Podsumowanie i ewaluacja

Nauczyciel rozmawia z uczniami o ich największych sukcesach w wykonanym zadaniu, największych trudnościach i sposobach poradzenia sobie z nimi. Rozdaje formularze "trzy gwiazdki i życzenie" ([Załącznik nr 2](#)) i prosi o opisanie swoich wrażeń po wykonaniu projektu:

- w polach "gwiazdki" uczniowie wpisują rzeczy, które im się podobały w projekcie,
- w polu "życzenie" uczniowie wpisują rzecz, którą chcieliby zmienić albo zrobić inaczej.


Uwagi

Zachęcamy, by podczas tej lekcji szczególną uwagę zwrócić na dokumentację jej przebiegu. Można zrobić filmiki, zdjęcia - także przydzielając uczniom różne reporterskie zadania.

Karta samooceny

Gra przygotowana przez drużynę (nazwa drużyny):

Skład drużyny:



			
Zadania	W grze pojawia się jedno zadanie lub wszystkie zadania są bardzo podobne.	Gra składa się z dwóch do czterech zadań, zadania są w większości różnorodne i wykorzystują zmienne.	Gra składa się z czterech lub więcej zadań, zadania są różnorodne, część z nich korzysta ze zmiennych.
Plansza	Plansza jest nieestetyczna, trudno zrozumieć drogę robota, droga jest zbyt wąska albo zbyt krótka, żeby robot mógł się swobodnie po niej poruszać.	Plansza jest estetyczna, droga robota jest czytelna, ale odległości nie zawsze pozwalają na swobodną jazdę robota, Wielkość pól nie zawsze odpowiada wielkości robota.	Plansza jest estetyczna i ciekawa, droga robota czytelna, odległości pozwalają na swobodną jazdę robota, wielkość pól jest dopasowana do wielkości robota.
Przebieg gry	Gra nie przebiega płynnie, pojawiają się liczne błędy czy trudności, plansza startowa nie działa poprawnie, zadania nie pojawiają się i nie kończą odpowiednio. Robot nie reaguje odpowiednio na poprawnie i niepoprawnie wykonane zadania.	Gra generalnie przebiega płynnie, mogą pojawić się pojedyncze błędy czy trudności. Zadania odpowiednio pojawiają się i kończą. Plansza startowa działa poprawnie. Robot reaguje odpowiednio na poprawnie i niepoprawnie wykonane zadania.	Gra przebiega płynnie, zadania odpowiednio pojawiają się i kończą, plansza startowa działa poprawnie, gra kończy się planszą końcową. Robot reaguje odpowiednio na poprawnie i niepoprawnie wykonane zadania.
Współpraca	Grę stworzył tylko jeden uczeń z grupy. Pozostali nie brali czynnego udziału w zadaniu.	Gra stworzona została tylko przez część członków grupy.	Wszyscy członkowie grupy mieli wkład w tworzenie gry, odpowiednio do możliwości i zainteresowań.

Karta informacji zwrotnej

Gra przygotowana przez drużynę (nazwa drużyny):

Skład drużyny:

Nasz wynik:minut sekund

			
Zadania	W grze pojawia się jedno zadanie lub wszystkie zadania są bardzo podobne.	Gra składa się z dwóch do czterech zadań, zadania są w większości różnorodne i wykorzystują zmienne.	Gra składa się z czterech lub więcej zadań, zadania są różnorodne, część z nich korzysta ze zmiennych.
Plansza	Plansza jest nieestetyczna, trudno zrozumieć drogę robota, droga jest zbyt wąska albo zbyt krótka, żeby robot mógł się swobodnie po niej poruszać.	Plansza jest estetyczna, droga robota jest czytelna, ale odległości nie zawsze pozwalają na swobodną jazdę robota, Wielkość pól nie zawsze odpowiada wielkości robota.	Plansza jest estetyczna i ciekawa, droga robota czytelna, odległości pozwalają na swobodną jazdę robota, wielkość pól jest dopasowana do wielkości robota.
Przebieg gry	Gra nie przebiega płynnie, pojawiają się liczne błędy czy trudności, plansza startowa nie działa poprawnie, zadania nie pojawiają się i nie kończą odpowiednio.	Gra generalnie przebiega płynnie, mogą pojawić się pojedyncze błędy czy trudności. Zadania odpowiednio pojawiają się i kończą. Plansza startowa działa poprawnie.	Gra przebiega płynnie, zadania odpowiednio pojawiają się i kończą, plansza startowa działa poprawnie, gra kończy się planszą końcową.

Załącznik nr 2

