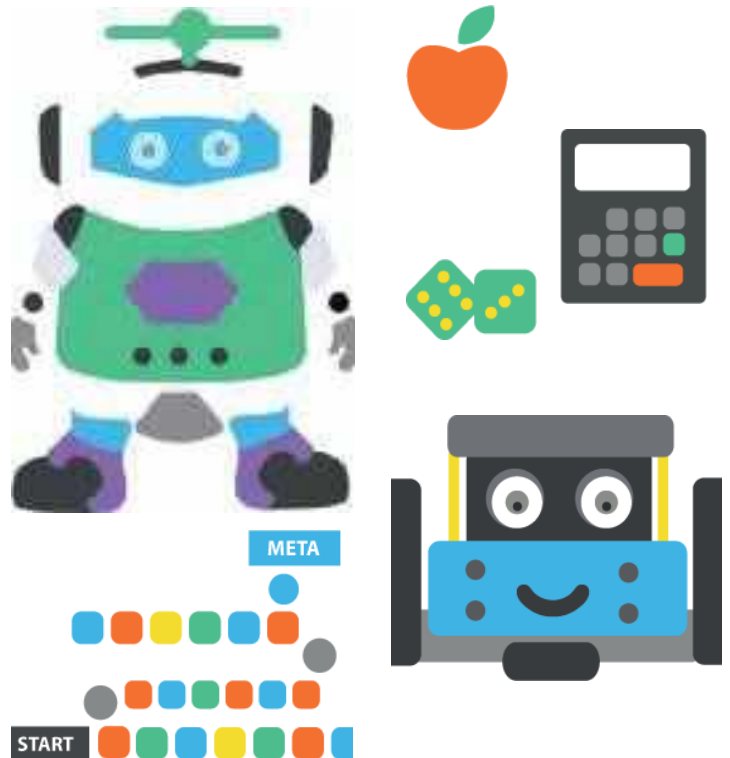


Robo -matematycy

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!



Autorzy:

Joanna Płatkowska-Nęcka, Karolina Czerwińska

Spis lekcji:

- Lekcja 1:** Hello World! Pierwsza linijka kodu dla mBota.
- Lekcja 2:** Silniki, brzęczyki i diody, czyli poznajemy się bliżej z mBotem.
- Lekcja 3:** Programowanie - sposób na łatwe obliczenia.
- Lekcja 4:** Ułamki dziesiętne, procenty oraz... mBot-matematyka.
- 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

Autorzy: Joanna Płatkowska-Nęcka, Karolina Czerwińska

Lekcja 1:

Hello World! Pierwsze linijki kodu dla mBota

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

Cele lekcji:

Uczeń powinien:

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

Materiały pomocnicze:

- roboty mBot V1.1
- 4 baterie AA dla każdego z robotów lub 4 akumulatory AA z ładowarką (wersja bardziej ekonomiczna) - zasilanie robota,
- 1 bateria CR2025 dla każdego z robotów - zasilanie pilota,
- program mBlock (do kodowania robota, scenariusze są zgodne z wersją v5.2.0): <http://www.mblock.cc/download/>,
- komputery stacjonarne lub laptopy (ze sprawnym portem USB).

Pojęcia kluczowe:

→ robot → silnik → czujnik → mikrokontroler mCore → mBlock

Czas realizacji: 45 min.

Metody pracy:

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

Treści programowe:

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

- 1) Liczby naturalne w dziesiętkowym układzie pozycyjnym.

Uczeń powinien:

- 1.1. zapisywać i odczytywać liczby naturalne wielocyfrowe;
- 1.3. porównywać liczby naturalne;

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

I. Rozumienie, analizowanie i rozwiązywanie problemów. Uczeń:

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

- a) rozwiązywanie problemów z życia codziennego i z różnych przedmiotów, np. liczenie średniej, pisemne wykonanie działań arytmetycznych, takich jak dodawanie i odejmowanie,
- b) osiągnięcie postawionego celu, w tym znalezienie elementu w zbiorze nieuporządkowanym lub uporządkowanym, znalezienie elementu najmniejszego i największego,
- c) sterowanie robotem lub obiektem na ekranie.

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

- 1) Projektuje, tworzy i zapisuje w wizualnym języku programowania:

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

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

IV. Rozwijanie kompetencji społecznych. Uczeń:

- 2) Identyfikuje i docenia korzyści płynące ze współpracy nad wspólnym rozwiązywaniem problemów

Przygotowania przed zajęciami:

- 1. Złożenie robota.** W scenariuszach nie przewidujemy osobnej lekcji poświęconej na składanie robotów. Nauczyciel może sam je przygotować 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** oraz w książeczce dołączonej do zestawu mBot.
- 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 3- lub 4-osobowych. Do każdej z grup przypisujemy jednego mBota - podział na grupy wynika więc z liczby robotów, którymi dysponujemy.
- 4. Edukacja matematyczna.** Naszym założeniem jest, by uczniowie w praktyczny sposób uczyli się i utrwalali materiał realizowany na lekcjach matematyki. Program zakłada łączenie programowania i treści z podstawy programowej do przedmiotu matematyka.

Wprowadzenie w tematykę i integracja grupy

Nauczyciel omawia z uczniami plan na kolejne #SuperKoderskie lekcje. Ich celem będzie stworzenie gry planszowej, w której "pionkiem" i "sędzią" będzie mBot i która będzie wymagała od uczniów zastosowania wiedzy zdobytej na lekcjach matematyki. Przygotowanie gry będzie składało się z kilku etapów:

1. Lekcje 1 i 2 - poznanie mBota i możliwości jego zaprogramowania,
2. Lekcje 3 i 4 - omówienie, jak w środowisku mBlock (podobnym do Scratcha) zapisywać i rozwiązywać zadania matematyczne,
3. Lekcje 5, 6, 7 - zaprojektowanie planszy, po której będzie jeździł robot oraz zadań dla przeciwnej drużyny,
4. Lekcja 8 - wielki turniej i wspólna gra w matematyczne gry planszowe.

Nauczyciel zachęca uczniów do zadawania pytań dotyczących projektu. Następnie dzieli klasę na drużyny.

Część zasadnicza

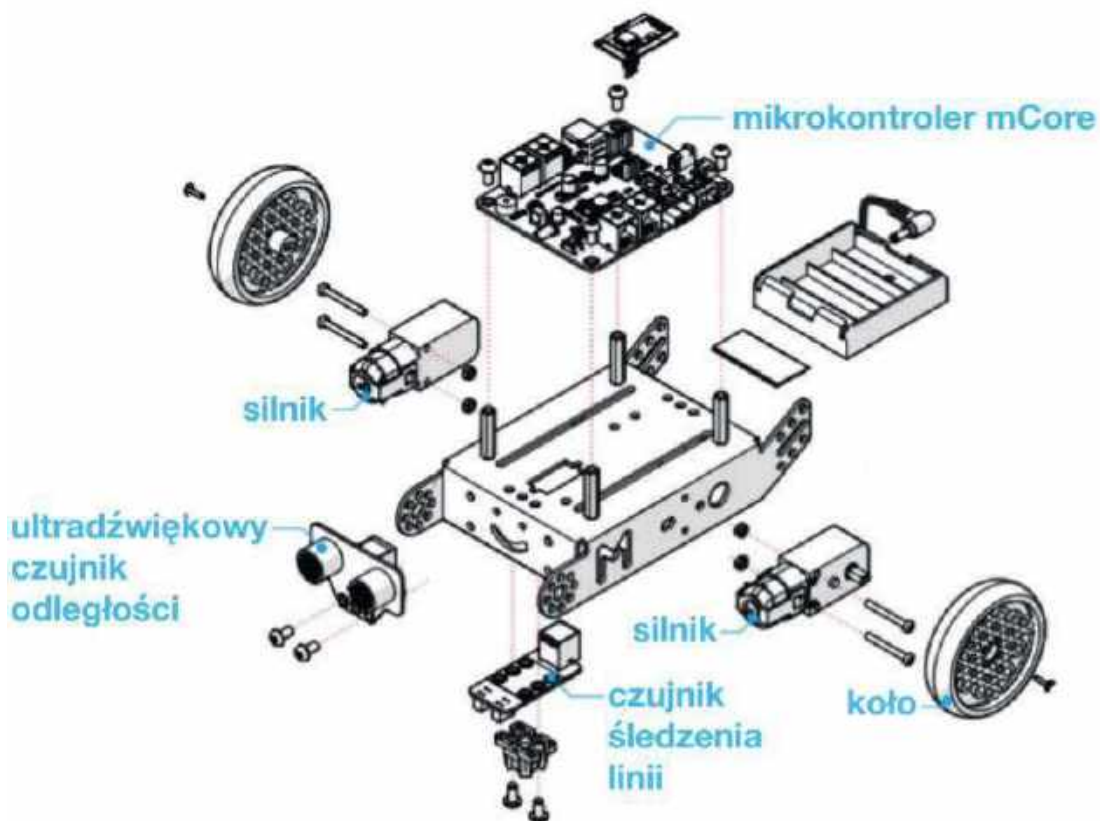
Zadanie 1: budowa robota

Nauczyciel prezentuje mBota 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!

Gdzie znajduje się?...	Uwagi
silnik?	mBot posiada dwa silniki - prawy i lewy. Programując robota, możemy korzystać z gotowych funkcji jazdy naprzód, do tyłu, w prawo i w lewo lub sterować mocą silnika prawego i lewego osobno.
ultradźwiękowy czujnik odległości?	Czujnik odległości działa podobnie jak echolokacja u nietoperzy: czujnik wysyła sygnał ultradźwiękowy przed siebie, a gdy sygnał napotka przeszkodę (np. ścianę), wraca z powrotem do czujnika. Jest to dla robota sygnał, by zmienić kierunek ruchu i uniknąć przeszkody.

czujnik śledzenia linii?	Czujnik śledzenia linii szczytuje dane na temat narysowanej linii. Aby dobrze działał, konieczne jest narysowanie lub wyklejenie bardzo kontrastowej i wyraźnej linii (nieżle spisyjesię w tym celu czarna taśma izolacyjna).
mikrokontroler mCore?	Po złożeniu mBota mikrokontroler jest ukryty pod osłoną z pleksi - można jednak pokusić się o częściowe rozebranie jednego z robotów i pokazanie płytki mCore. Jest to najważniejszy element robota, który umożliwia jego programowanie. Płytkę mCore ma 3 ważne dla nas rzeczy: <ul style="list-style-type: none">· moduł bluetooth, dzięki któremu robot komunikuje się z komputerem (po podłączeniu odbiornika do portu USB w komputerze),· diody - robot potrafi świecić w różnych kolorach,· brzęczyk (buzzer) - robot potrafi wydawać dźwięki. Warto dodać, że płytkę mCore powstała na bazie mikrokontrolera Arduino, który daje niemal nieograniczone możliwości zastosowania w projektowaniu własnych robotów i elektronicznych urządzeń.
koło?	Odpowiedzi są oczywiste :) Warto jednak zapytać w tym miejscu uczniów, po jakiej powierzchni robot będzie jeździł szybciej: po drewnianej podłodze czy po dywanie? Nauczyciel wyjaśnia, że mniejsze tarcie będą miały powierzchnie gładkie i twarde, i po nich robot będzie jeździł szybciej.



Zadanie 2: co mBot potrafi?

Po omówieniu budowy mBota, nauczyciel zaprasza uczniów do zabawy robotami (ok. 5-10 minut). Uczniowie po kolei testują trzy tryby pracy robota:

1. Robot zdalnie sterowany,
2. Robot unikający przeszkód,
3. Robot śledzący linię. W zestawie mBot znajduje się plansza z linią, którą można wykorzystać do tego ćwiczenia.



Zadanie 3: łączenie mBota z komputerem

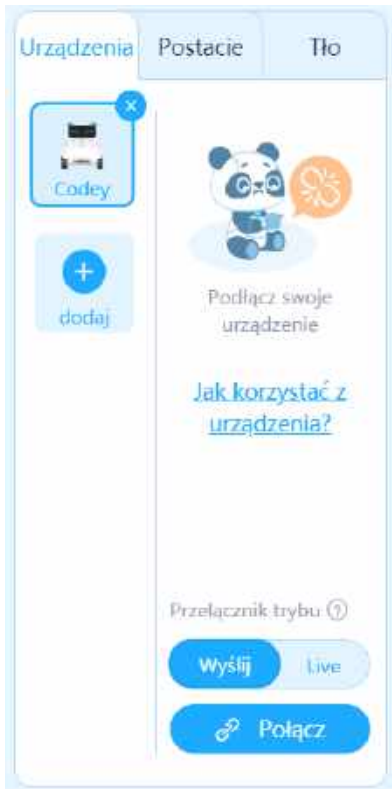
Po przeprowadzonych testach nauczyciel zaprasza uczniów do napisania pierwszych programów komputerowych z wykorzystaniem mBota. Uczniowie siadają do komputerów i podłączają do komputerów. Kolejne kroki:

1. Podłączenie modułu Bluetooth Dongle do portu USB komputera (dioda na module powinna błyskać niebieskim światłem),
2. Wciśnięcie guzika z ikoną bluetooth na module (dioda miga szybko – urządzenie jest w trybie parowania),*
3. Włączenie robota.



#SuperKoderzy/Robo-matematycy/Hello World! Pierwsze linijki kodu dla mBota

4. Uruchomienie aplikacji mBlock,
5. Usunięcie robota Codey (domyślnie ustawiony przy pierwszym uruchomieniu aplikacji mBlock) – należy w lewym dolnym ekranie odszukać Devices (Urządzenia) i obok obrazka przedstawiającego robota Codey kliknąć krzyżyk,*
6. Dodanie mBota poprzez kliknięcie plusa z napisem add (dodaj),*
7. Wybranie robota mBot,*
8. Kliknięcie ikony gwiazdki obok mBota, żeby ustawić go, jako urządzenie startowe,*
9. Kliknięcie przycisku Connect (Połącz).



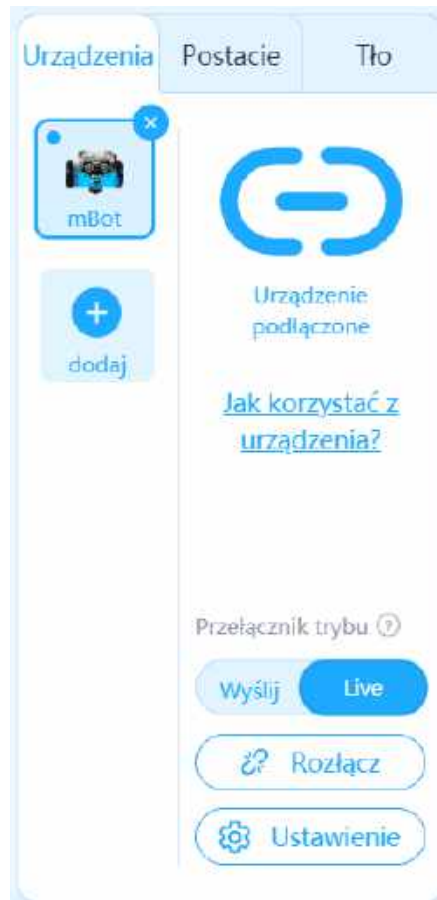
* te punkty trzeba wykonać tylko przy pierwszym uruchomieniu aplikacji mBlock



10. Kliknięcie Connect (Połącz)

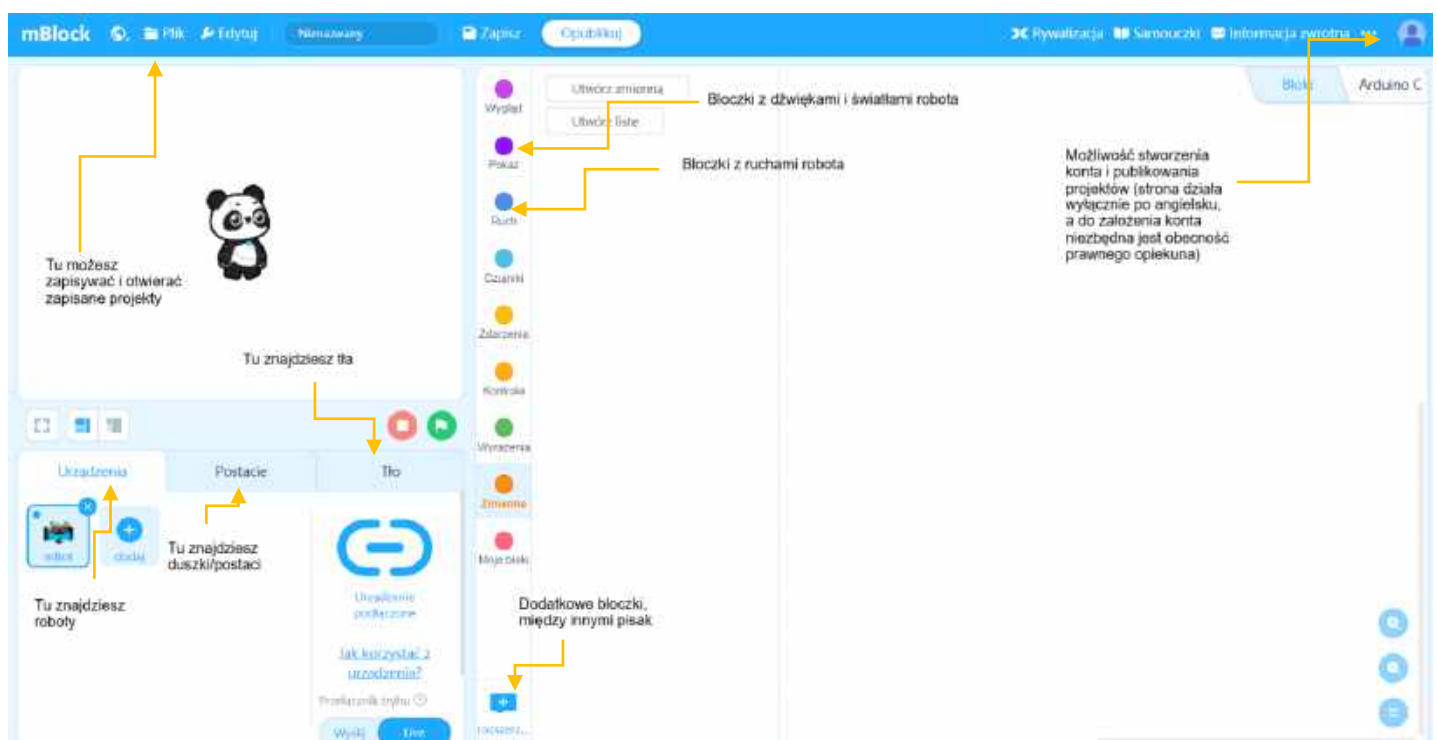


11. Sprawdzenie połączenia w części skryptów dedykowanej mBotowi (niebieska kropka wskazuje na pomyślne połączenie z robotem/brak kropki wskazuje na brak połączenia):



Zadanie 4: pierwszy program dla mBota

Uczniowie zapoznają się z mBlokiem i tym, jak różni się od Scratcha



#SuperKoderzy/Robo-matematycy/Hello World! Pierwsze linijki kodu dla mBota

Po zapoznaniu się z układem mBlocka, uczniowie poznają funkcje robota. Znajdują bloczki, które pozwalają na ruch, świecenie różnymi kolorami światła i wydawanie dźwięków. Odkrywają, że część bloczków robota jest taka sama jak duszków, na przykład kontrola, zdarzenia czy wyrażenia. Oglądają, jakie bloczki ma dodatkowo robot w grupie czujniki. Wraz z nauczycielem zastanawiają się, w jaki sposób robot może komunikować się z duszkami. Nauczyciel pokazuje, że może to robić poprzez zmienne lub komunikaty (message). Można tu posłużyć się porównaniem do sztafety i przekazywania pałeczki.

Zadaniem uczniów jest napisać pierwszy prosty program, w którym duszek będzie mówił, jak daleko trzymamy rękę od czujnika odległości robota. Warto tu zwrócić uwagę, że bloczków z czujnikami robota nie możemy bezpośrednio wkleić do skryptu robota, ale możemy to sprytnie obejść tworząc zmienną i ustawiając na starcie jej wartość na czujnik odległości.

Skrypt dla robota będzie wyglądał tak:



Skrypt dla duszka będzie wyglądał tak:



Podsumowanie i ewaluacja

Zadanie 1: budowa robota

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 napiszemy kolejne programy dla mBotów.

Zadanie domowe: proszę obejrzeć 2 klipy na YouTube:

- <https://www.youtube.com/watch?v=-hQtdYd5i-Y>
- <https://www.youtube.com/watch?v=yx6JtQVpcUw&t=55s>

Pytanie kontrolne do filmu nr 2: jakie czujniki zostały wykorzystane do zaprogramowania robota rozwiązującego labirynt?

Robo-matematycy

Autorzy: Joanna Płatkowska-Nęcka, Karolina Czerwińska

Lekcja 2:

Silniki, brzęczyki i diody, czyli poznajemy się bliżej z mBotem

Lekcja, podczas której uczniowie oswoją się z programowaniem mBota, wykorzystując jego diody, brzęczyki i silniki. Poznają także model RGB barw i będą testować, jakie kolory światła może wyświetlać robot. Podejmą także próbę muzykowania przy użyciu mBota.

Cele lekcji:

Uczeń powinien:

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

Materiały pomocnicze:

- roboty mBot V1.1,
- 4 baterie AA dla każdego z robotów lub 4 akumulatory AA z ładowarką (wersja bardziej ekonomiczna) - zasilanie robota,
- 1 bateria CR2025 dla każdego z robotów - zasilanie pilota,
- program mBlock (do kodowania robota), scenariusze są zgodne z wersją v5.2.0): <http://www.mblock.cc/download/>,
- komputery stacjonarne lub laptopy (ze sprawnym portem USB),
- projektor, ekran lub tablica interaktywna - podłączone do komputera nauczyciela.

Pojęcia kluczowe:

→ robot → silnik → dioda RGB → brzęczyk (buzzer) → mikrokontroler mCore → mBlock

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-VIII, matematyka: Liczby naturalne w dziesiętkowym układzie pozycyjnym. Uczeń powinien:

- 1.1. zapisywać i odczytywać liczby naturalne wielocyfrowe;
- 1.3. porównywać liczby naturalne;

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

I. Rozumienie, analizowanie i rozwiązywanie problemów. Uczeń:

- 2) formułuje i zapisuje w postaci algorytmów polecenia składające się na:
 12. rozwiązanie problemów z życia codziennego i z różnych przedmiotów, np. liczenie średniej, pisemne wykonanie działań arytmetycznych, takich jak dodawanie i odejmowanie,
 13. osiągnięcie postawionego celu, w tym znalezienie elementu w zbiorze nieuporządkowanym lub uporządkowanym, znalezienie elementu najmniejszego i największego.
 14. sterowanie robotem lub obiektem na ekranie.

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

- 1) Projektuje, tworzy i zapisuje w wizualnym języku programowania:
 - a) pomysły historyjek i rozwiązania problemów, w tym proste algorytmy z wykorzystaniem poleceń sekwencyjnych, warunkowych i iteracyjnych oraz zdarzeń jednoczesnych;
 - b) prosty program sterujący robotem lub innym obiektem na ekranie komputera.

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

IV. Rozwijanie kompetencji społecznych. Uczeń:

- 2) Identyfikuje i docenia korzyści płynące ze współpracy nad wspólnym rozwiązywaniem problemów

Przebieg zajęć:

Wprowadzenie w tematykę i integracja grupy

Nauczyciel nawiązuje do zadania domowego zadanego po pierwszej lekcji, sprawdza, kto z uczniów obejrzał filmiki. Zachęca do udzielenia odpowiedzi na pytanie kontrolne: "jakie czujniki zostały wykorzystane do zaprogramowania robota rozwiązującego labirynt?". Pyta także uczniów, jakie części robota potrafią wymienić.

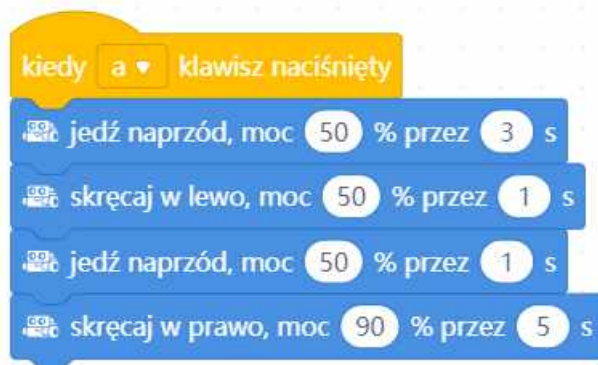
Część zasadnicza

Nauczyciel tłumaczy uczniom, że na dzisiejszych zajęciach napiszą wspólnie kilka programów, w których wykorzystają różne funkcjonalności mBota. Prosi uczniów o uruchomienie programu mBlock, podłączenie modułu bluetooth do portu USB komputera, włączenie robota i o sparowanie go z komputerem.

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, skręcił i by po dotarciu do celu obrócił się kilka razy wokół własnej osi (odtańczył taniec radości).

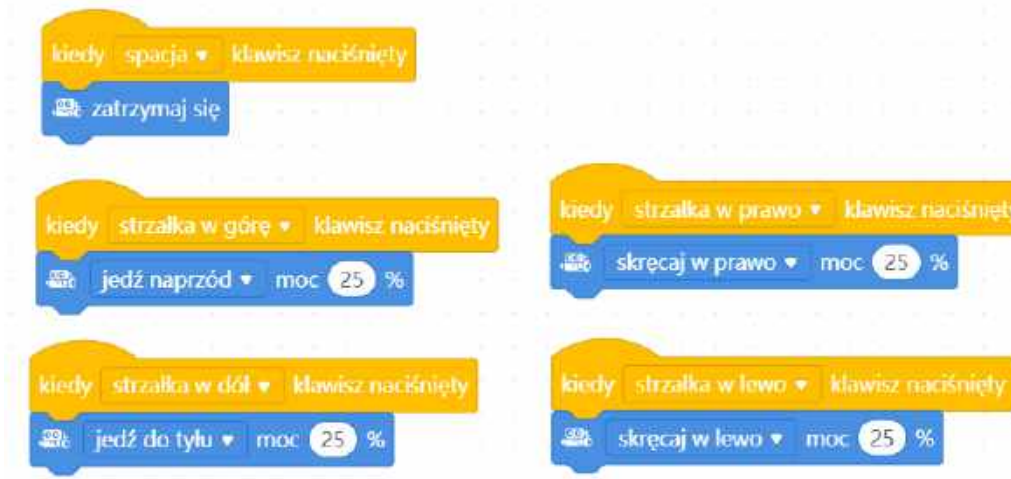
Sytuacją idealną jest, gdy uczniowie samodzielnie znajdą sposób na rozwiązanie tego zadania. Jeśli okaże się, że mają z nim problem, nauczyciel podpowiada, jakie bloczki należy wykorzystać do rozwiązania tego zadania (np. wyświetla je na projektorze).



Ważne, żeby pamiętać, że prędkość robota będzie uzależniona od naładowania baterii (im bardziej naładowane, tym szybciej będzie jechał przy takich samych wartościach procentowych).

Warto zapytać uczniów, co zrobić, jeśli robot nie chce się zatrzymać lub nie dojechał dokładnie do założonego punktu. Uczniowie mogą tu rozpatrzyć zmianę mocy silników i/lub czasu jazdy. Warto zaprogramować tutaj „hamulec bezpieczeństwa” i dodatkowe sterowanie klawiaturą (przyda się szczególnie, kiedy kod jest już napisany i po jakimś czasie przy mocniej rozładowanych bateriach, ruchy robota nie są tak precyzyjne, jak wcześniej). Sterowanie pilotem jest również możliwe, ale jeśli roboty znajdują się blisko (czyli w jednej sali), piloty będą „przejmować” różne roboty; w związku z tym lepiej nie korzystać z pilotów.

Można zaproponować następujących pięć sekwencji i przesunąć ją na bok strony ze skryptami:



Jeśli uczniowie szybko wykonają zadanie, można dodawać kolejne “stacje”, które robot odwiedzi.

Zadanie 2: programowanie diod robota

Nauczyciel zaprasza uczniów do wykonania kolejnego zadania, podczas którego nauczą się wyświetlać na robocie różne kolory. Mówi uczniom, że robot jest wyposażony w trzy diody o trzech różnych kolorach, z połączenia których może powstać ogromna liczba barw. Jest to model RGB kolorów. Podpowiada uczniom, że skrót “RGB” pochodzi z angielskiego i oznacza trzy barwy - zachęca uczniów, by zgadli, o jakie barwy chodzi. Poprawna odpowiedź: red - czerwony; green - zielony; blue - niebieski.

Nauczyciel wyświetla stronę internetową: <http://www.colorsfire.com/rgb-color-wheel/>. Przy przesuwaniu suwakiem widać, jak zmieniają się wartości trzech modelowych kolorów (RGB) i jakie kolory wówczas powstają.

mBlock 5.2 pozwala na wybranie koloru światła bezpośrednio z palety, bez konieczności ustawiania wartości RGB. Pokazujemy ją uczniom, żeby zrozumieli mechanizm powstawania różnych barw.

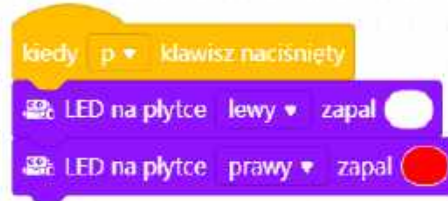
Nauczyciel proponuje uczniom, żeby spróbowali, jakie barwy światła można uzyskać, manipulując kolorem, nasyceniem i jasnością:



Następnie nauczyciel zaprasza uczniów do krótkiego ćwiczenia-quizu. Prosi uczniów o wyszukanie flagi wybranego państwa w internecie lub rozdaje im załącznik nr 1 (po jednym wydruku dla każdej z grup). Prosi uczniów, by każda z grup wyświetliła na swoim robocie kolory flagi wybranego europejskiego państwa. Pozostałe grupy zgadują, o flagę jakiego państwa chodzi.

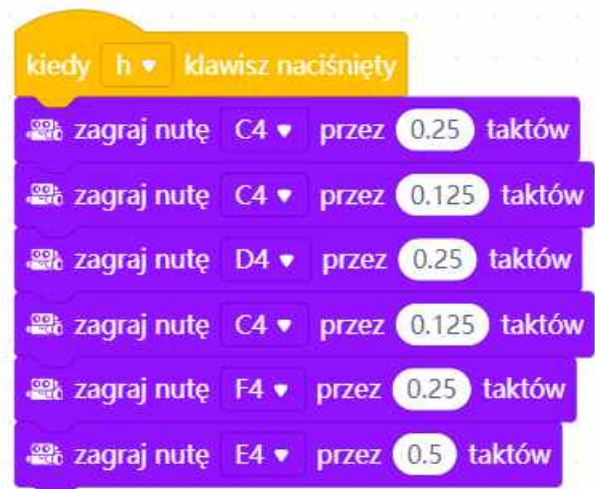
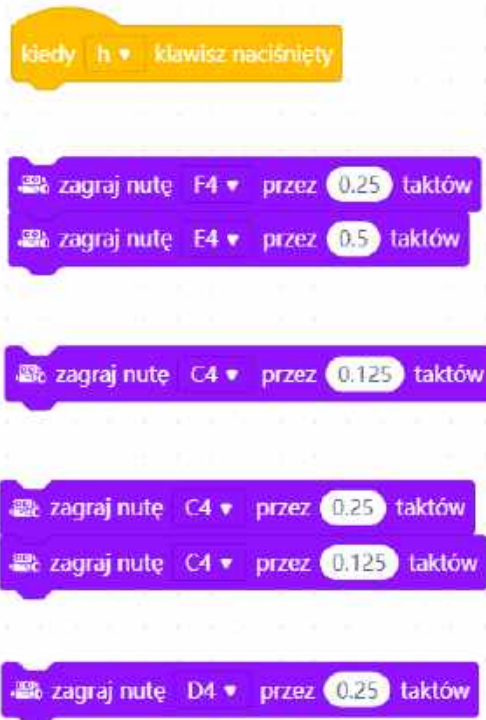


Np. flaga Polski:



Zadanie 3: programowanie buzzera

Nauczyciel przypomina uczniom, że płytka mCore wyposażona jest również w brzęczyk (tzw. buzzer) i tłumaczy, że można na nim grać różne proste melodie. Zachęca uczniów do testów. Może też umieścić na tablicy rozsypankę, z której uczniowie mogą ułożyć znaną melodię:



Zadanie 4: włączanie robota do gier - zadanie dodatkowe

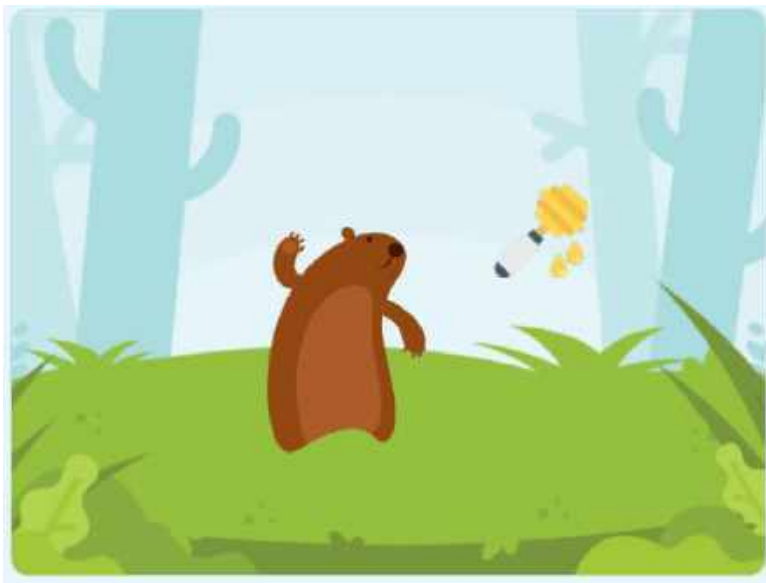
Jeśli grupa sprawnie pracuje, nauczyciel proponuje wykonanie dodatkowego zadania, w którym uczniowie napiszą krótką grę komputerową, w którą włączony będzie robot.

Gra polega na tym, że niedźwiedź próbuje złapać pojawiający się w losowo wybranych miejscach miód. Za każdym razem, gdy go złapie, robot świeci czerwonym światłem i wydaje dźwięk. Niedźwiedź porusza się dzięki czujnikowi odległości robota – ręka bliżej lub dalej od robota będzie odpowiednio poruszać duszkiem w lewo lub w prawo (tu trzeba zwrócić uwagę, w którą stronę zwrócony jest robot).

#SuperKoderzy / Robo-matematycy / Silniki, brzęczyki i diody, czyli poznajemy się bliżej z mBotem

Oczywiście bohaterami gry mogą być zupełnie inne postaci czy przedmioty.

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



Skrypt dla niedźwiedzia:



Skrypt dla miodu:



Skrypt dla mBota:



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 udokumentowania 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

Autorzy: Joanna Płatkowska-Nęcka, Karolina Czerwińska

Lekcja 3:

Programowanie - sposób na łatwe obliczenia

Zajęcia, na których uczniowie odkrywają, w jaki sposób stosować bloczki w 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:

- roboty mBot V1.1,
- 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 całko- witych, 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-VIII, matematyka:

- 1) Liczby naturalne w dziesiętkowym układzie pozycyjnym. Uczeń powinien:
 - 1.1. zapisywać i odczytywać liczby naturalne wielocyfrowe;
 - 1.3. porównywać liczby naturalne;
- 2) Działania na liczbach naturalnych. Uczeń powinien:

- 2.1. dodawać i odejmować w pamięci liczby naturalne dwucyfrowe lub większe (...);
- 2.2. dodawać i odejmować liczby naturalne wielocyfrowe (...);
- 2.3. mnożyć i dzielić liczbę naturalną przez liczbę naturalną jednocyfrową, dwucyfrową lub trzycyfrową (...);
14. Zadania tekstowe. Uczeń powinien:
 - 14.1. umieć przeczytać ze zrozumieniem tekst zawierający informacje liczbowe;
 - 14.3. dostrzegać zależności między podanymi informacjami;
 - 14.6. weryfikować wynik zadania tekstowego, oceniając sensowność rozwiązania.

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

- I. Rozumienie, analizowanie i rozwiązywanie problemów. Uczeń:
 - 2) formułuje i zapisuje w postaci algorytmów polecenia składające się na:
 - a) rozwiązanie problemów z życia codziennego i z różnych przedmiotów, np. liczenie średniej, pisemne wykonanie działań arytmetycznych, takich jak dodawanie i odejmowanie,
 - b) osiągnięcie postawionego celu, w tym znalezienie elementu w zbiorze nieuporządkowanym lub uporządkowanym, znalezienie elementu najmniejszego i największego.
 - c) sterowanie robotem lub obiektem na ekranie.
 - II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych. Uczeń:
 - 1) Projektuje, tworzy i zapisuje w wizualnym języku programowania:
 - a) pomysły historyjek i rozwiązania problemów, w tym proste algorytmy z wykorzystaniem poleceń sekwencyjnych, warunkowych i iteracyjnych oraz zdarzeń jednoczesnych;
 - b) prosty program sterujący robotem lub innym obiektem na ekranie komputera.
- Testuje na komputerze swoje programy pod względem zgodności z przyjętymi założeniami i ewentualnie je poprawia, objaśnia przebieg działania programów.
- 3) Gromadzi, porządkuje i selekcjonuje efekty swojej pracy oraz potrzebne zasoby w komputerze lub w innych urządzeniach oraz środowiskach wirtualnych (w chmurze).
- IV. Rozwijanie kompetencji społecznych. Uczeń:
 - 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 gry matematyczne ze studia #SuperKoderzy/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 i zapisują je na tablicy lub r odejmowanie, mnożenie, dzi

ch grach. Zbierają pomysły ce propozycje: dodawanie,

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 liczb. Nauczyciel wraz z uczniami wybierają te liczby, które będą stanowić podstawę tego zadania i będą spełniały następujący warunek: pierwsza z nich będzie 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. Uczniowie tworzą dwie zmienne ("liczba 1" i "liczba 2"), a następnie ustawiają je jako wartości losowe (przedział ustalony w zależności od poziomu zaawansowania uczniów).

Wskazówka: Aby ułatwić przeanalizowanie działania tego programu, warto w blockach zmienne zaznaczyć te zmienne ptaszkiem, dzięki czemu będą wyświetlać się na ekranie, a także o zanotowanie wyników na kartce. Dla ułatwienia można zmienić czas ich wyświetlania np. na 5



Nauczyciel prosi uczniów o kilkukrotne uruchomienie programu. Pyta, co zauważają w przypadku odejmowania. Okazuje się, że co jakiś czas wynikiem są liczby ujemne. Nauczyciel wyjaśnia, że w wielu przypadkach wynik ujemny nie będzie pasował jako rozwiązanie zadania tekstowego.

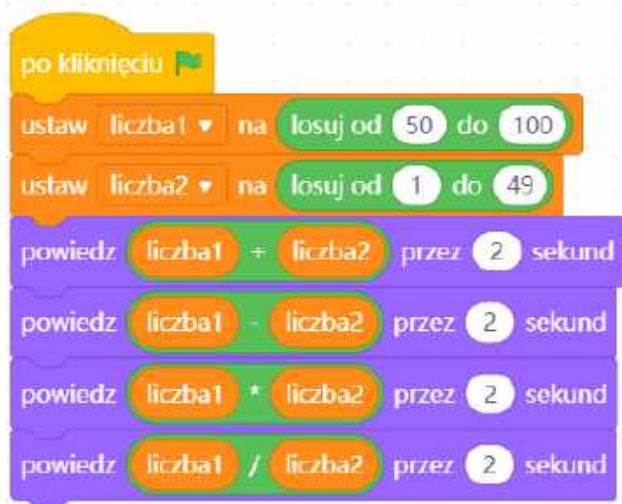
Jako przykład nauczyciel podaje zadanie: "Zuzia ma [liczba 1] książek, a Janek ma [liczba 2] książek. O ile więcej książek ma Zuzia od Janka?" i prosi uczniów o podanie odpowiedzi dla następujących danych:

- 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 książek" 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:

a) Ustawienie początku zakresu "liczby 1" o większej wartości niż koniec zakresu "liczby 2"

b) Zastosowanie bloczka «powtarzaj aż» (szczególnie w przypadku, kiedy nie chcemy zawęzić zakresów)



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

Zadanie 3

Następnie nauczyciel pyta o wynik dzielenia. Nie zawsze jest on liczbą całkowitą, co w zależności od rodzaju zadania tekstowego, może dać odpowiedź pozbawioną sensu.

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

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:

- liczba 1 = 120, liczba 2 = 12
- liczba 1 = 114, 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 "dziewięć i pół zawodnika").

Uczniowie wspólnie z nauczycielem zastanawiają się, jak rozwiązać tę trudność. Żeby liczba dzieliła się bez reszty, musi być 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).



Zadanie 4

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

Nauczyciel pokazuje uczniom proste zadanie i omawia z uczniami działanie bloczka "zadaj pytanie i czekaj" oraz wykorzystanie pętli warunkowej z odpowiedzią.



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.

Załącznik nr 1

Robo-matematycy

Autorzy: Joanna Płatkowska-Nęcka, Karolina Czerwińska

Lekcja 4:

Ułamki dziesiętne, procenty oraz... mBot-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,
- 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 mBot V1.1,
- 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-VIII, matematyka:

- 4) Ułamki zwykłe i dziesiętne. Uczeń powinien:
 - 4.2. przedstawiać ułamek jako iloraz liczb naturalnych, a iloraz liczb naturalnych jako ułamek zwykły;
- 5) Działania na ułamkach zwykłych i dziesiętnych. Uczeń powinien:
 - 5.2. dodawać, odejmować, mnożyć i dzielić ułamki dziesiętne (...);
 - 5.7. obliczać wartości prostych wyrażeń arytmetycznych, stosując

reguły dotyczące kolejności wykonywania działań;

5.8. wykonywać działania na ułamkach dziesiętnych (...);

12) Obliczenia praktyczne. Uczeń powinien:

12.1. interpretować 1% jako jedną setną część danej wielkości liczbowej;

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

14) Zadania tekstowe. Uczeń powinien:

14.1. umieć przeczytać ze zrozumieniem tekst zawierający informacje liczbowe;

14.3. dostrzegać zależności między podanymi informacjami;

14.6. weryfikować wynik zadania tekstowego, oceniając sensowność rozwiązania.

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

I. Rozumienie, analizowanie i rozwiązywanie problemów. Uczeń:

1) formułuje i zapisuje w postaci algorytmów polecenia składające się na:

- a) rozwiązanie problemów z życia codziennego i z różnych przedmiotów, np. liczenie średniej, pisemne wykonanie działań arytmetycznych, takich jak dodawanie i odejmowanie,
- c) sterowanie robotem lub obiektem na ekranie.

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

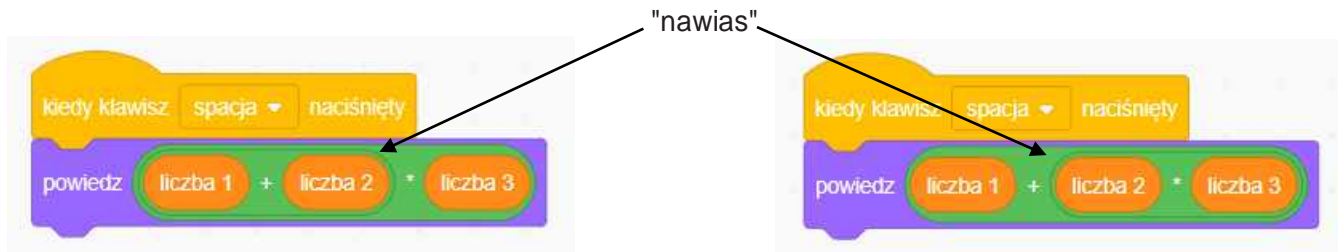
- 1) Projektuje, tworzy i zapisuje w wizualnym języku programowania:
 - a) pomysły historyjek i rozwiązania problemów, w tym proste algorytmy z wykorzystaniem poleceń sekwencyjnych, warunkowych i iteracyjnych oraz zdarzeń jednoczesnych;
 - b) prosty program sterujący robotem lub innym obiektem na ekranie komputera.

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

- 3) Gromadzi, porządkuje i selekcjonuje efekty swojej pracy oraz potrzebne zasoby w komputerze lub w innych urządzeniach oraz środowiskach wirtualnych (w chmurze).

Wprowadzenie w tematykę i integracja grupy

Uczniowie 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 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

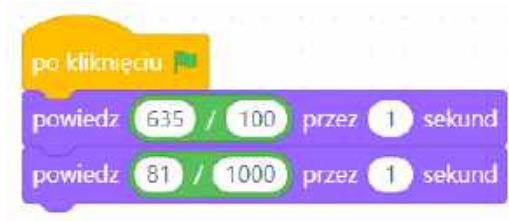
Część 1

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.

Uczniowie 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 $12*4-16/4=$. Uczniowie wymieniają się zadaniami, a następnie piszą program, w którym duszek poda rozwiązanie. Muszą szczególnie zwrócić tu uwagę na 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 lub dzielenie). Warto powtórzyć takie zapisywanie, korzystając z ułamków z różną liczbą cyfr po przecinku. Następnie uczniowie zapisują trzy różne ułamki (najlepiej z różną liczbą cyfr po przecinku), wymieniają się nimi i piszą program, dzięki któremu duszek "powie" podane ułamki.



Po zakończeniu tego zadania uczniowie 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

Uczniowie wykorzystują zadanie z poprzedniej lekcji ("Ile godzin ma doba?") i dodają do skryptu działanie robota tak, aby po poprawnej odpowiedzi świecił na zielono i grał trzy dźwięki, a przy błędnej świecił na czerwono i grał jeden 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» z grupy «wyrażenia». Warto zwrócić uczniom uwagę na to, że trzeba zastosować na końcu i na początku tekstu przed lub po zmiennej, oprócz sytuacji, kiedy występują znaki przestankowe, które nie wymagają spacji (inaczej zmienna i tekst połączą się w jedną nieczytelną całość).

Uczniowie wraz z nauczycielem piszą 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

 <p>Scratch script 1: A sequence of blocks starting with a green flag icon labeled '1000'. This is followed by a purple 'say Witaj! przez 2 sekund' block, a yellow 'when clicked' block, a green 'pick random number from 1 to 100' block, a green 'pick random number from 1 to 1000' block, another purple 'say Witaj! przez 2 sekund' block, and finally a green flag icon labeled '100'.</p>	 <p>Scratch script 2: A sequence of blocks starting with a green flag icon labeled '1000'. This is followed by a purple 'say Witaj! przez 2 sekund' block, a yellow 'when clicked' block, a green 'pick random number from 1 to 100' block, a green 'pick random number from 1 to 1000' block, another purple 'say Witaj! przez 2 sekund' block, and finally a green flag icon labeled '100'.</p>
 <p>Scratch script 3: A sequence of blocks starting with a green flag icon labeled '1000'. This is followed by a purple 'say Witaj! przez 2 sekund' block, a yellow 'when clicked' block, a green 'pick random number from 1 to 100' block, a green 'pick random number from 1 to 1000' block, another purple 'say Witaj! przez 2 sekund' block, and finally a green flag icon labeled '100'.</p>	 <p>Scratch script 4: A sequence of blocks starting with a green flag icon labeled '1000'. This is followed by a purple 'say Witaj! przez 2 sekund' block, a yellow 'when clicked' block, a green 'pick random number from 1 to 100' block, a green 'pick random number from 1 to 1000' block, another purple 'say Witaj! przez 2 sekund' block, and finally a green flag icon labeled '100'.</p>

Robo-matematycy

Autorzy: 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 mBota, 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 mBot v 1.1,
- program mBlock,
- komputery stacjonarne lub laptopy,
- przygotowana wcześniej plansza dla mBota (inspiracja do obejrzenia w [Załączniku Nr 1](#) oraz tutaj: <https://youtu.be/Zsn77kh5jqc>), program "robo mat" pobrany ze strony: http://superkoderzy.pl/wp-content/themes/superkoderzy/include/programy/robo_mat.mblock

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-VIII, matematyka:

- 4) Ułamki zwykłe i dziesiętne. Uczeń powinien:
 - 4.2. przedstawiać ułamek jako iloraz liczb naturalnych, a iloraz liczb naturalnych jako ułamek zwykły;
- 5) Działania na ułamkach zwykłych i dziesiętnych. Uczeń powinien:
 - 5.2. dodawać, odejmować, mnożyć i dzielić ułamki dziesiętne (...);
 - 5.7. obliczać wartość prostych wyrażeń arytmetycznych, stosując reguły dotyczące kolejności wykonywania działań;
 - 5.8. wykonywać działania na ułamkach dziesiętnych (...);
- 12) Obliczenia praktyczne. Uczeń powinien:
 - 12.1. interpretować 1% jako jedną setną część danej wielkości

liczbowej;

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

14) Zadania tekstowe. Uczeń powinien:

- 14.1. umieć przeczytać ze zrozumieniem tekst zawierający informacje liczbowe;
- 14.3. dostrzegać zależności między podanymi informacjami;
- 14.6. weryfikować wynik zadania tekstowego, oceniając sensowność rozwiązania.

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

I. Rozumienie, analizowanie i rozwiązywanie problemów. Uczeń:

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

- a) rozwiązywanie problemów z życia codziennego i z różnych przedmiotów, np. liczenie średniej, pisemne wykonanie działań arytmetycznych, takich jak dodawanie i odejmowanie, c) sterowanie robotem lub obiektem na ekranie.

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

- 1) Projektuje, tworzy i zapisuje w wizualnym języku programowania:
 - a) pomysły historyjek i rozwiązania problemów, w tym proste algorytmy z wykorzystaniem poleceń sekwencyjnych, warunkowych i iteracyjnych oraz zdarzeń jednoczesnych;
 - b) prosty program sterujący robotem lub innym obiektem na ekranie komputera.

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

- 3) Gromadzi, porządkuje i selekcjonuje efekty swojej pracy oraz potrzebne zasoby w komputerze lub w innych urządzeniach oraz środowiskach wirtualnych (w chmurze).

IV. Rozwijanie kompetencji społecznych. Uczeń:

- 2) Identyfikuje i docenia korzyści płynące ze współpracy nad wspólnym rozwiązywaniem problemów.

Wprowadzenie w tematykę i integracja grupy

Nauczyciel rozkłada planszę dla mBota (przygotowaną na szkoleniu lub przed tymi zajęciami), ustawia robota na starcie i uruchamia przygotowaną wcześniej grę matematyczną lub tę, którą można pobrać ze strony http://superkoderzy.pl/wp-content/themes/superkoderzy/include/programy/robo_mat.mblock.

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). Istotne jest, żeby pokazać wszystkie cztery zadania i wyjaśnić, co dzieje się przy odpowiedzi błędnej.

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 gra drużyna przeciwna. Im trudniejsza gra, tym lepiej, ponieważ wygrywa ta drużyna, która szybciej przejdzie grę przygotowaną przez przeciwnika.
- Na grę składa się:
 - plansza, po której jeździ robot. Wykonujemy ją np. z połączonych flipchartów. Rysujemy na niej tor z polami, a kilka z nich to "pola specjalne" lub same "pola specjalne" połączone liniami, po których porusza się robot, na których kryją się zadania matematyczne do rozwiązania. Jeśli stanie na nich robot, uczniowie przystępują do rozwiązywania zadań matematycznych stworzonych w programie mBlock. Liczbę "pól specjalnych" ustala nauczyciel w zależności od możliwości i chęci grupy - może to być stała liczba (np. 4 - dla takiej liczby jest przygotowany wzór planszy i rubryka do oceny). Można też przyjąć zasadę "kto da więcej" - im więcej zadań napiszą uczniowie, tym większa szansa, że przeciwna drużyna dłużej będzie grała w grę.
 - robot mBot. Poruszamy nim za pomocą strzałek (napiszemy do tego oddzielny skrypt). Robot kontroluje, czy zadania matematyczne z "pól specjalnych" zostały poprawnie wykonane - jeśli tak, świeci się na zielono i gra 3 dźwięki. Jeśli nie - świeci się na czerwono i gra 1 dźwięk. 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).
 - kostka do gry - w taki analogowy sposób określamy, o ile pól przesuwa się nasz robot; jeśli plansza nie ma pól pomiędzy polami specjalnymi, a jedynie trasę dla robota, kostka nie będzie potrzebna.
 - stoper do pomiaru czasu. Jeśli nie mamy stoperów, można skorzystać z tego programu: <https://scratch.mit.edu/projects/167824599/> (trudność tego rozwiązania polega na tym, że w żadnym momencie nie można zatrzymać stopera w przypadku nagłej przerwy w grze).

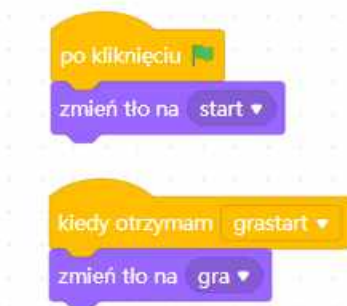
Dla zobrazowania tych wyjaśnień można wyświetlić zdjęcia z [Załącznika nr 1](#) oraz filmik <https://youtu.be/Zsn77kh5jqc>

Nauczyciel informuje uczniów, że od tego momentu muszą dbać, by na jednym z komputerów przypisanych do drużyny była tworzona wersja "master" programu (gry), czyli ta najbardziej poprawna. Oznacza to, że poszczególne zadania mogą być tworzone indywidualnie na różnych komputerach, ale potem ich poprawne wersje muszą zostać przepisane do programu "master". Niestety mBlock nie dopuszcza opcji kopiowania skryptów między różnymi programami, dlatego jedyną metodą na dodawanie kolejnych bloków w wersji "master" jest ułożenie ich "na piechotę" według wcześniej opracowanego wzoru. Warto zachęcić uczniów, by w każdej drużynie była osoba, której dodatkowym zadaniem będzie opiekowanie się wersją "master" - będzie ona dbała o zapisywanie zmian, integrowanie jej z programami pisanymi indywidualnie przez członków drużyny, robienie back-upu po każdym zajęciu (poprzez wysłanie maila lub zapisanie na pendrive'ie).

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 duży napis START. Po jego kliknięciu na ekranie mogą pojawić się duże numery od 1 do 4. Po kliknięciu każdego z nich, wszystkie znikają i pojawia się odpowiednie zadanie.

Skrypt tła startowego



Skrypt tła z numerami



Skrypt dla wybranego numeru (w tym przypadku zadania)



Część 2 - sterowanie robotem

Ponieważ między polami z pytaniami robot musi jechać kierowany strzałkami, należy pamiętać o napisaniu dla niego odpowiednich skryptów.



Uczniowie pracują w grupach, żeby stworzyć stronę startową i skrypt sterowania robota. Jeśli uczniowie pracują w grupach czteroosobowych, można ustalić, że każdy uczeń jest liderem przy tworzeniu jednego z zadań (można spróbować przydzielić role liderów w tworzeniu zadań również przy innej liczbie uczniów w grupach).

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 dziać się po prawidłowej odpowiedzi (nasza propozycja: zielone światło robota, dźwięk i podjechanie do przodu),
- określenie, co ma dziać się po nieprawidłowej odpowiedzi (nasza propozycja: czerwone światło robota, dźwięk i odjechanie do tyłu).

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:



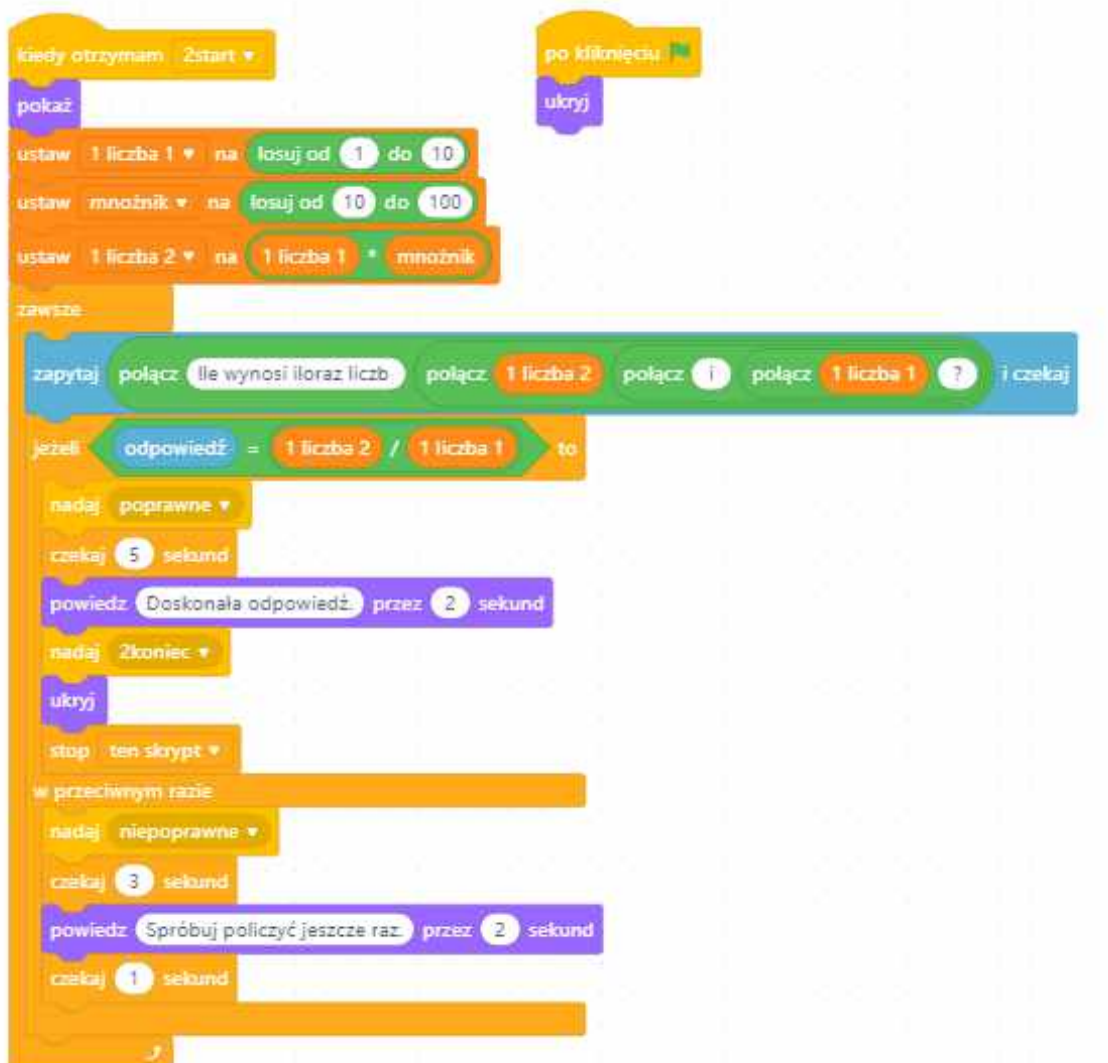
Wersja rozszerzona tego samego zadania:

```
when key pressed:
  show
  hide
  change costume to Bug
  set 1 liczba 1 to random number from 1 to 10
  set 1 liczba 2 to random number from 10 to 100 / 100
  always:
    ask "Ile wynosi suma liczb" of 1 liczba 1 and 1 liczba 2 and wait
    if answer = 1 liczba 1 + 1 liczba 2 then
      give correct
      wait 5 seconds
      say "Brawo! Możesz przejść do następnego pytania." for 2 seconds
      change costume to Bug2
    always:
      ask "Ile wynosi różnica liczb" of 1 liczba 1 and 1 liczba 2 and wait
      if answer = 1 liczba 1 - 1 liczba 2 then
        give correct
        wait 5 seconds
        say "Doskonała odpowiedź." for 2 seconds
        stop this script
      else
        give incorrect
        wait 3 seconds
        say "Spróbuj policzyć jeszcze raz." for 2 seconds
        wait 1 second
      else
        give incorrect
        wait 3 seconds
        say "Spróbuj policzyć jeszcze raz." for 2 seconds
        wait 1 second
```

#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.



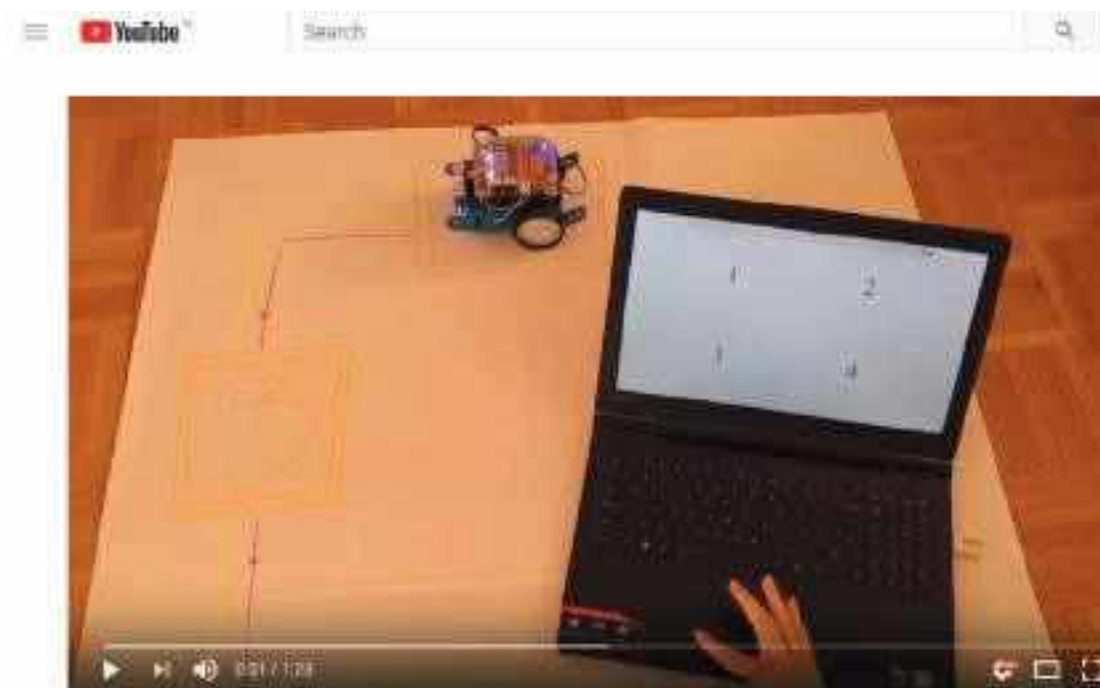
Uczniowie pracują w grupach, żeby stworzyć skrypt dla dwóch pierwszych zadań. W zależności od możliwości czasowych mogą dodać efekty zarówno na ekranie startowym, jak i w działaniu robota.

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.

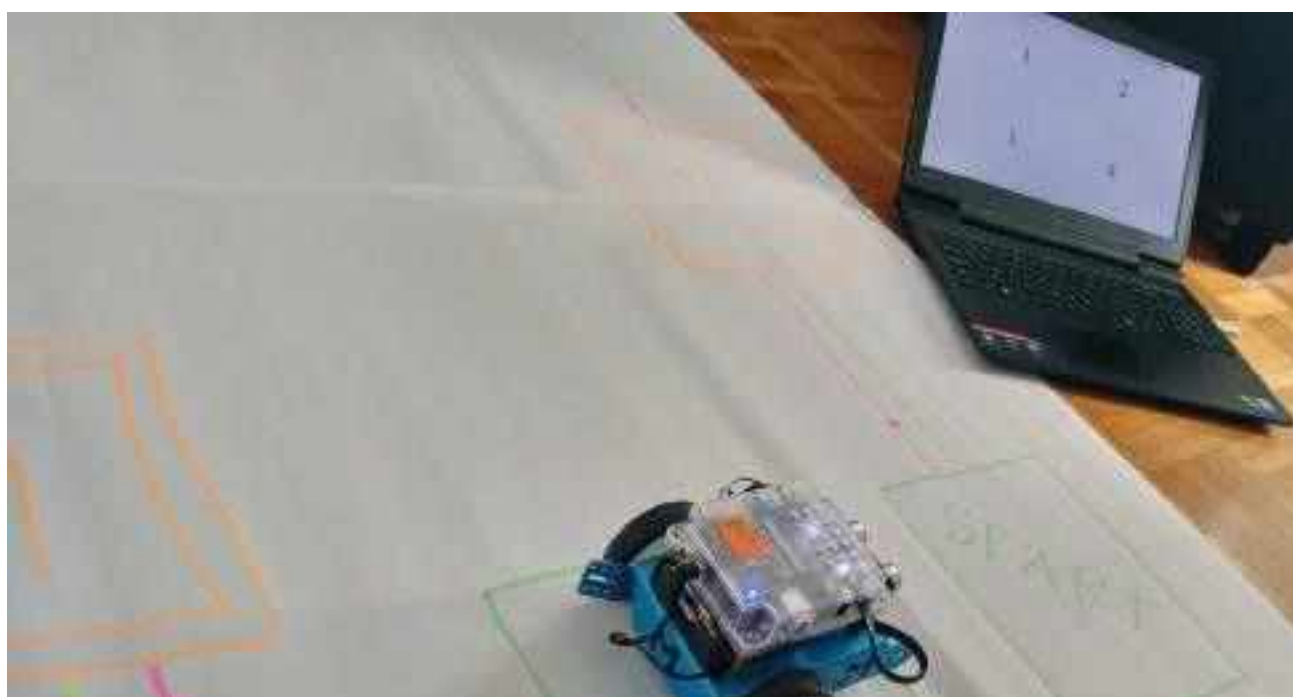
Nauczyciel przypomina uczniom o zapisaniu plików tak, by były one dostępne na następnych zajęciach.

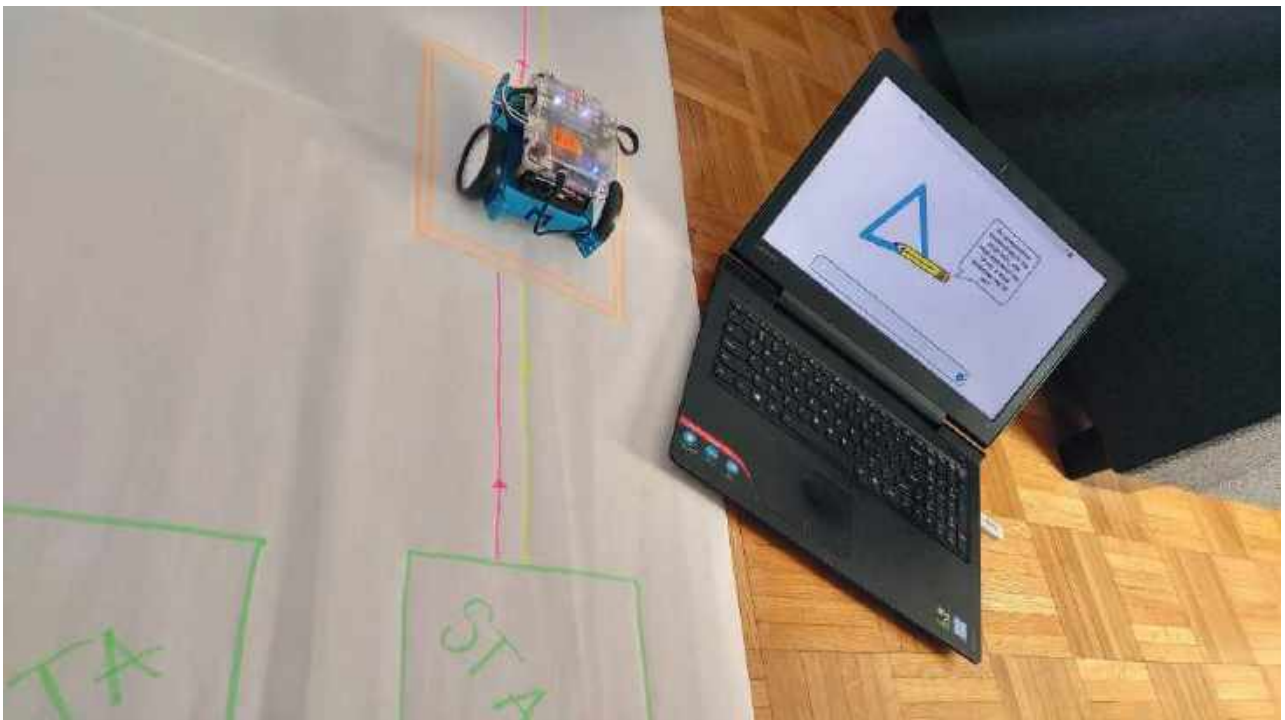
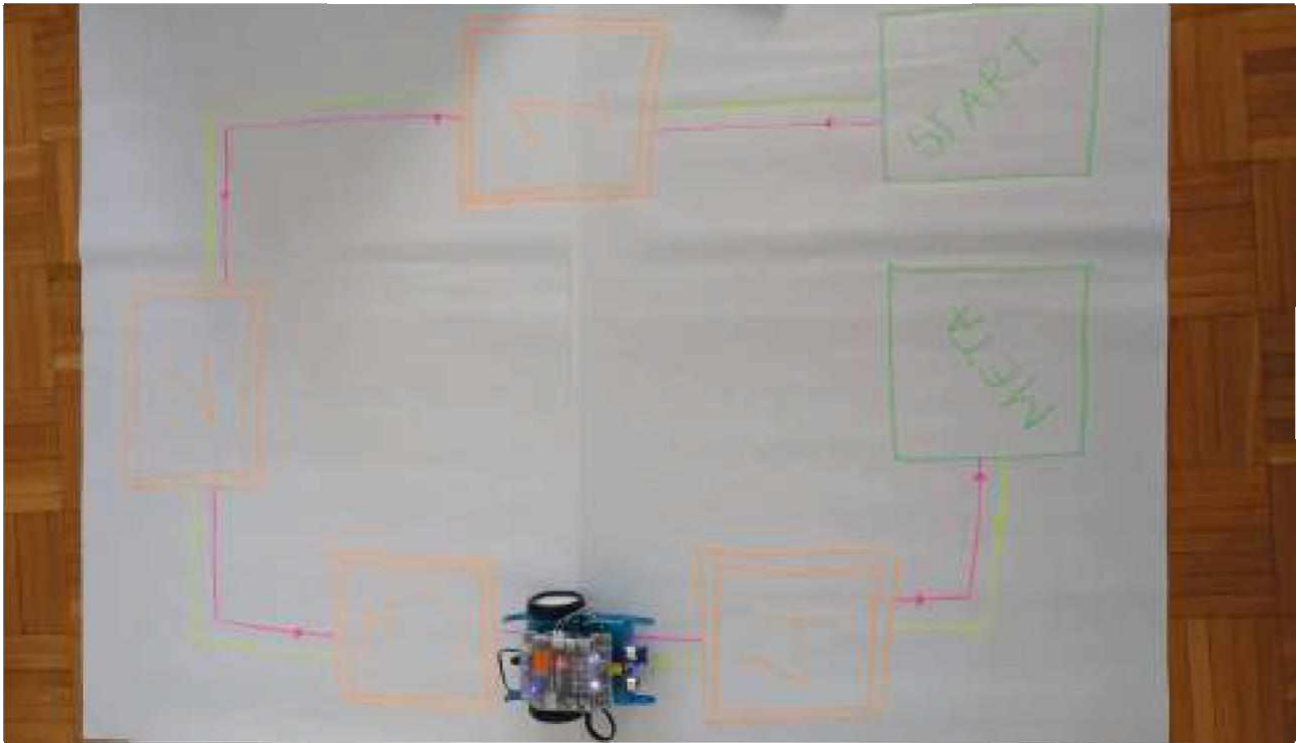
Załącznik nr 1



#SuperKoderzy - Ścieżka matematyczna

filmik: <https://youtu.be/Zsn77kh5jqc>





Robo-matematycy

Autorzy: 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 mBot v 1.1,
- program mBlock,
- komputery stacjonarne lub laptopy,
- program "lekcja5.sb2" pobrany ze strony: http://superkoderzy.pl/wp-content/themes/superkoderzy/include/programy/robo_mat.mblock

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-VIII, matematyka:

9) Wielokąty, koła i okręgi. Uczeń powinien:

- 9.3. stosować twierdzenie o sumie kątów trójkąta;
- 9.4. rozpoznawać i nazywać: kwadrat, prostokąt, romb, równoległobok i trapez;
- 9.5. znać najważniejsze własności kwadratu, prostokąta, rombu, równoległoboku i trapezu;

11) Obliczenia w geometrii. Uczeń powinien:

- 11.1. umieć obliczyć obwód wielokąta o danych długościach boków;
- 11.2. obliczać pola: kwadratu, prostokąta, rombu, równoległoboku, trójkąta, trapezu przedstawionych na rysunku oraz w sytuacjach praktycznych;
- 11.3. stosować jednostki pola: mm², cm², dm², m², km², ar, hektar (bez zamiany jednostek w trakcie obliczeń);

12) Obliczenia praktyczne. Uczeń powinien:

- 12.1. interpretować 1% jako jedną setną część danej wielkości liczbowej;
- 12.2. w przypadkach osadzonych w kontekście praktycznym obliczać procent danej wielkości w stopniu trudności typu 50%, 20%, 10%;

14) Zadania tekstowe. Uczeń powinien:

- 14.1. umieć przeczytać ze zrozumieniem tekst zawierający informacje liczbowe;
- 14.3. dostrzegać zależności między podanymi informacjami;
- 14.6. weryfikować wynik zadania tekstowego, oceniając sensowność rozwiązania.

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

I. Rozumienie, analizowanie i rozwiązywanie problemów. Uczeń:

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

- a) rozwiązanie problemów z życia codziennego i z różnych przedmiotów, np. liczenie średniej, pisemne wykonanie działań arytmetycznych, takich jak dodawanie i odejmowanie,
- c) sterowanie robotem lub obiektem na ekranie.

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

- 1) Projektuje, tworzy i zapisuje w wizualnym języku programowania:
 - a) pomysły historyjek i rozwiązania problemów, w tym proste algorytmy z wykorzystaniem poleceń sekwencyjnych, warunkowych i iteracyjnych oraz zdarzeń jednoczesnych;
 - b) prosty program sterujący robotem lub innym obiektem na ekranie komputera.

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

- 3) Gromadzi, porządkuje i selekcjonuje efekty swojej pracy oraz potrzebne zasoby w komputerze lub w innych urządzeniach oraz środowiskach wirtualnych (w chmurze).

IV. Rozwijanie kompetencji społecznych. Uczeń:

- 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/>

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 kwadrat o boku 15 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.

```
when green flag clicked
  go to x: 0 y: 0
  delete everything
  move pen to x: 160 y: 0
  move pen to x: 120 y: 100
  move pen to x: 40 y: 100
  move pen 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 "odpowiedź" = 0.5 * wysokość * górna podstawa + dolna podstawa then
      give "poprawne"
      wait 5 seconds
      say "Brawo!" for 2 seconds
      delete everything
      give "koniec"
      stop this script
    otherwise
      give "niepoprawne"
      wait 3 seconds
      say "Sprawdź czy dobrze zastosowałeś wzór na pole powierzchni trapezu." for 2 seconds
      wait 1 second
```

W zaproponowanych przykładach znajdujących się w studio #SuperKoderzy/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

Autorzy: 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 mBot v 1.1 i 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-VIII, matematyka:

- 1) Liczby naturalne w dziesiętkowym układzie pozycyjnym. Uczeń powinien:
 - 1.1. zapisywać i odczytywać liczby naturalne wielocyfrowe;
- 2) Działania na liczbach naturalnych. Uczeń powinien:
 - 2.1. dodawać i odejmować w pamięci liczby naturalne dwucyfrowe lub większe (...)
 - 2.2. dodawać i odejmować liczby naturalne wielocyfrowe (...);
 - 2.3. mnożyć i dzielić liczbę naturalną przez liczbę naturalną jednocyfrową, dwucyfrową lub trzycyfrową (...);
 - 2.11. stosować reguły dotyczące kolejności wykonywania działań;
- 9) Wielokąty, koła i okręgi. Uczeń powinien:
 - 9.3. stosować twierdzenie o sumie kątów trójkąta;
 - 9.4. rozpoznawać i nazywać: kwadrat, prostokąt, romb, równoległobok i trapez;
 - 9.5. znać najważniejsze własności kwadratu, prostokąta, rombu, równoległoboku i trapezu;
- 11) Obliczenia w geometrii. Uczeń powinien:
 - 11.1. umieć obliczyć obwód wielokąta o danych długościach boków;
 - 11.2. obliczać pola: kwadratu, prostokąta, rombu, równoległoboku,

trójkąta, trapezu przedstawionych na rysunku oraz w sytuacjach praktycznych;

11.3. stosować jednostki pola: mm², cm², dm², m², km², ar, hektar (bez zamiany jednostek w trakcie obliczeń);

14) Zadania tekstowe. Uczeń powinien:

14.1. umieć przeczytać ze zrozumieniem tekst zawierający informacje liczbowe;

14.3. dostrzegać zależności między podanymi informacjami;

14.6. weryfikować wynik zadania tekstowego, oceniając sensowność rozwiązania.

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

I. Rozumienie, analizowanie i rozwiązywanie problemów. Uczeń:

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

- a) rozwiązanie problemów z życia codziennego i z różnych przedmiotów, np. liczenie średniej, pisemne wykonanie działań arytmetycznych, takich jak dodawanie i odejmowanie,
- c) 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:
 - a) pomysły historyjek i rozwiązania problemów, w tym proste algorytmy z wykorzystaniem poleceń sekwencyjnych, warunkowych i iteracyjnych oraz zdarzeń jednoczesnych;
 - b) prosty program sterujący robotem lub innym obiektem na ekranie komputera.

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

3) Gromadzi, porządkuje i selekcjonuje efekty swojej pracy oraz potrzebne zasoby w komputerze lub w innych urządzeniach oraz środowiskach wirtualnych (w chmurze).

IV. Rozwijanie kompetencji społecznych. Uczeń:

- 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. 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.
- Konieczne jest napisanie skryptu, dzięki któremu robot będzie poruszał się i zatrzymywał.
- 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 jazda do tyłu 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).

Podsumowanie i ewaluacja

Uczniowie dokonują samooceny korzystając z listy kontrolnej.

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 znajdują się duszki poszczególnych zadań.



Konieczne jest napisanie skryptu, dzięki któremu robot będzie poruszał się i zatrzymywał.



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 jazda do tyłu 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).

Robo-matematycy

Autorzy: Joanna Płatkowska-Nęcka, Karolina Czerwińska

Lekcja 8:

Wielki matematyczny turniej-gramy!

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 mBot v 1.1 i 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-VIII, matematyka:

- 1) Liczby naturalne w dziesiętkowym układzie pozycyjnym. Uczeń powinien:
 - 1.1. zapisywać i odczytywać liczby naturalne wielocyfrowe;
- 2) Działania na liczbach naturalnych. Uczeń powinien:
 - 2.1. dodawać i odejmować w pamięci liczby naturalne dwucyfrowe lub większe (...);
 - 2.2. dodawać i odejmować liczby naturalne wielocyfrowe (...);
 - 2.3. mnożyć i dzielić liczbę naturalną przez liczbę naturalną jednocyfrową, dwucyfrową lub trzycyfrową (...);
 - 2.11. stosować reguły dotyczące kolejności wykonywania działań;
- 4) Ułamki zwykłe i dziesiętne. Uczeń powinien:
 - 4.2. przedstawiać ułamek jako iloraz liczb naturalnych, a iloraz liczb naturalnych jako ułamek zwykły;
- 5) Działania na ułamkach zwykłych i dziesiętnych. Uczeń powinien:
 - 5.2. dodawać, odejmować, mnożyć i dzielić ułamki dziesiętne (...);
 - 5.7. obliczać wartość prostych wyrażeń arytmetycznych, stosując reguły dotyczące kolejności wykonywania działań;
 - 5.8. wykonywać działania na ułamkach dziesiętnych (...);
- 9) Wielokąty, koła i okręgi. Uczeń powinien:

- 9.4. rozpoznawać i nazywać: kwadrat, prostokąt, romb, równoległobok i trapez;
- 9.5. znać najważniejsze własności kwadratu, prostokąta, rombu, równoległoboku i trapezu;
- 11) Obliczenia w geometrii. Uczeń powinien:
 - 11.1. umieć obliczyć obwód wielokąta o danych długościach boków;
 - 11.2. obliczać pola: kwadratu, prostokąta, rombu, równoległoboku, trójkąta, trapezu przedstawionych na rysunku oraz w sytuacjach praktycznych;
 - 11.3. stosować jednostki pola: mm², cm², dm², m², km², ar, hektar (bez zamiany jednostek w trakcie obliczeń);
- 14) Zadania tekstowe. Uczeń powinien:
 - 14.1. umieć przeczytać ze zrozumieniem tekst zawierający informacje liczbowe;
 - 14.3. dostrzegać zależności między podanymi informacjami;
 - 14.6. weryfikować wynik zadania tekstowego, oceniając sensowność rozwiązania.

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

- I. Rozumienie, analizowanie i rozwiązywanie problemów. Uczeń:
 - 2) formułuje i zapisuje w postaci algorytmów polecenia składające się na:
 - a) rozwiązanie problemów z życia codziennego i z różnych przedmiotów, np. liczenie średniej, pisemne wykonanie działań arytmetycznych, takich jak dodawanie i odejmowanie,
 - c) sterowanie robotem lub obiektem na ekranie.
 - II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych. Uczeń:
 - 1) Projektuje, tworzy i zapisuje w wizualnym języku programowania:
 - a) pomysły historyjek i rozwiązania problemów, w tym proste algorytmy z wykorzystaniem poleceń sekwencyjnych, warunkowych i iteracyjnych oraz zdarzeń jednoczesnych;
 - b) prosty program sterujący robotem lub innym obiektem na ekranie komputera.
- Testuje na komputerze swoje programy pod względem zgodności z przyjętymi założeniami i ewentualnie je poprawia, objaśnia przebieg działania programów.
-) Gromadzi, porządkuje i selekcjonuje efekty swojej pracy oraz potrzebne zasoby w komputerze lub w innych urządzeniach oraz w środowiskach wirtualnych (w chmurze).
- IV. Rozwijanie kompetencji społecznych. Uczeń:
 - 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 z uczniami ustala, która z grup będzie grała w którą grę. W zależności od liczby grup można spróbować umożliwić wszystkim zagranie we wszystkie gry. Nauczyciel informuje również uczniów, że po każdej grze ocenia ją, korzystając z kryteriów oceny i formularza z [Załączniku nr 1](#).

Część zasadnicza

Uczniowie grają w gry stworzone przez inne grupy. Zapisują czas przejścia rozgrywki poszczególnych grup, żeby określić najkrótszy możliwy czas dla każdej z gier i wyłonić zwycięzców. Formułują także informację zwrotną dla przeciwnych drużyn oraz zastanawiają się nad oceną ich własnej gry - pomoże im w tym formularz z [Załącznika nr 1](#). Nauczyciel również ocenia pracę grup.

Nauczyciel z uczniami ustala, 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 mBota). 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

			
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 swobodne sterowanie robotem, Wielkość pól nie zawsze odpowiada wielkości robota.	Plansza jest estetyczna i ciekawa, droga robota czytelna, odległości pozwalają na swobodne kierowanie robotem, 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.
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.

Jak oceniacie grę, którą sami stworzyliście?

Gra przygotowana przez drużynę (nazwa drużyny):

Skład drużyny:

	Liczba przyznanych gwiazdek * / ** / ***	Komentarz
Zadania		
Plansza		
Przebieg gry		
Współpraca		

Karta informacji zwrotnej

	★	★ ★	★ ★ ★
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 swobodne sterowanie robotem, Wielkość pól nie zawsze odpowiada wielkości robota.	Plansza jest estetyczna i ciekawa, droga robota czytelna, odległości pozwalają na swobodne kierowanie robotem, 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.

Jak oceniacie grę, w którą graliście?

Gra przygotowana przez drużynę (nazwa drużyny):

Skład drużyny:

Nasz wynik:minut sekund

	Liczba przyznanych gwiazdek * / ** / ***	Komentarz
Zadania		
Plansza		
Przebieg gry		

Załącznik nr 2

