

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) 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 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 przesuwamy 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).

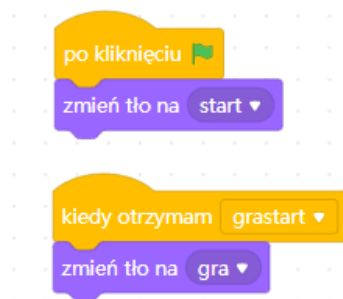
Dla zobrazowania tych wyjaśnień można wyświetlić zdjęcia z [Załącznika nr 1](#) oraz filmik <https://youtu.be/Zsn77kh5jqc>

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 bloczków 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).

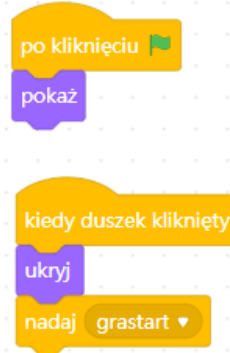
Część 1 - ekran startowy

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 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 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

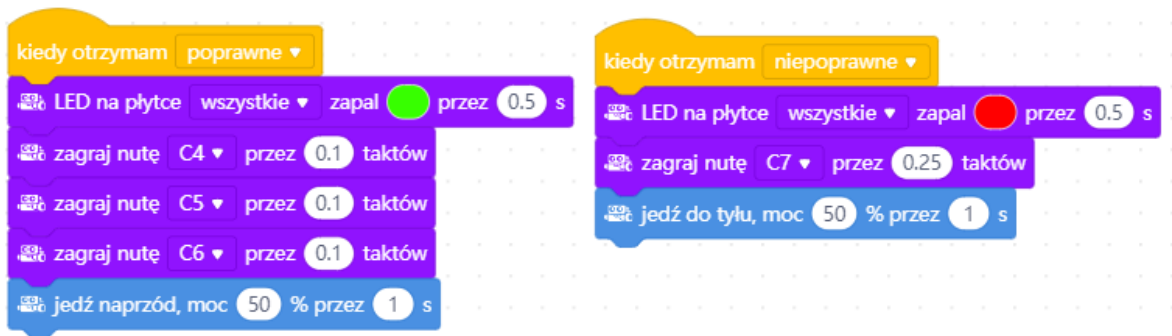
Każde z zadań musi składać się z następujących elementów:

- bloczek «pokaż»
- pętla «jeżeli to – w przeciwnym razie»

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

- określenie, co ma dzieć się po prawidłowej odpowiedzi (nasza propozycja: zielone światło robota, dźwięk i podjechanie do przodu),
- określenie, co ma dzieć 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:

```
when green flag clicked
  show
  change costume to Bug
  loop
    ask "Ile wynosi suma liczb 6.208 i 3.05?" and wait
    if answer is 9.258
      say "Brawo! Możesz przejść do następnego pytania." for 2 seconds
      change costume to Bug2
      loop
        ask "Ile wynosi różnica liczb 3.05 i 6.208?" and wait
        if answer is 3.158
          say "Doskonała odpowiedź." for 2 seconds
          give "4koniec"
          hide
          stop this script
        else
          give "niepoprawne"
          wait 3 seconds
          say "Spróbuj policzyć jeszcze raz." for 2 seconds
          wait 1 second
          loop
    else
      give "niepoprawne"
      wait 3 seconds
      say "Spróbuj policzyć jeszcze raz." for 2 seconds
      wait 1 second
      loop
```

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

Wersja rozszerzona tego samego zadania:

```
when key pressed (spacebar) clicked
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
forever loop
  ask (1 liczba 1 + 1 liczba 2) i wait
  if (answer = 1 liczba 1 + 1 liczba 2) then
    give correct
    wait 5 seconds
    say (Brawo! Możesz przejść do następnego pytania.) through 2 seconds
    change costume to Bug2
  else
    give incorrect
    wait 3 seconds
    say (Spróbuj policzyć jeszcze raz.) through 2 seconds
    wait 1 seconds
  end if
end loop
say (Spróbuj policzyć jeszcze raz.) through 2 seconds
wait 1 seconds
stop script
```

#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.



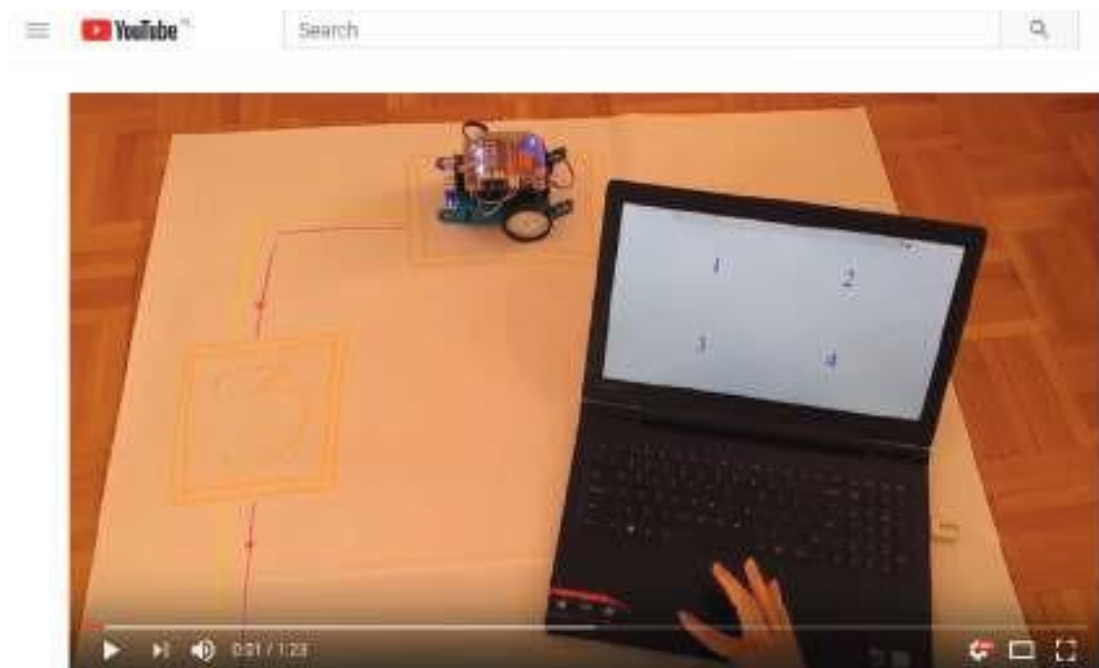
Uczniowie pracują w grupach, żeby stworzyć skrypt dla dwóch pierwszych zadań. W zależności od możliwości czasowych mogą dodać efekty zarówno na ekranie startowym, jak i w działaniu robota.

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>

