

Rafał Pietruszka
Szkoła Podstawowa nr 4
im. Jana Pawła II
w Lubartowie

Przykładowe ćwiczenia pomocne przy nauce programowania dla uczniów szkoły podstawowej z wykorzystaniem programu Scratch.

W procesie dydaktycznym nauczyciele stają przed zadaniem nauczania, kształcenia umiejętności oraz wychowania. Sposób przekazywanej wiedzy dostosowują do wieku i możliwości grupy, z którą pracują. Aby nauczyciel był skuteczny, powinien zainspirować uczniów, rozbudzić ich ciekawość, pozwolić na samodzielność oraz motywować do dalszego działania.¹ Proces nauczania może być łatwiejszy, gdy uda się nam zaangażować u ucznia wszystkie zmysły. Skuteczniejszym sposobem niż słuchanie wykładu nauczyciela jest poznawanie przez ucznia informacji poprzez obrazy oraz możliwość działania. Wzmaga to proces zapamiętywania i rozwój kompetencji przedmiotowych.² Uczniowie, których zmysły uda się nam pobudzić, są bardziej chętni do nauki i ciekawsi świata. Może to nastąpić poprzez wykorzystanie filmów, pomocy multimedialnych oraz eksponatów. Im częściej nauczyciel wykorzystuje metody angażujące różne zmysły jednocześnie, tym większą przykuwa uwagę uczniów.³

Nowoczesne technologie, które są wykorzystywane w odpowiedni sposób i z umiarem, potrafią ułatwić i uatrakcyjnić proces dydaktyczny. Za stosowaniem nowoczesnych technologii w edukacji przemawia wiele argumentów. Dzięki ich wykorzystaniu podczas zajęć dziecko rozwija nie tylko wyobraźnię.⁴ Gdy technologia jest zintegrowana z lekcjami, uczniowie stają się bardziej zainteresowani przedmiotami, których się uczą. Wykorzystanie technologii powoduje, iż nauczanie staje się przyjemniejsze, a nauczanie tych samych rzeczy w nowy, inny sposób sprawia uczniom frajdę. Wykorzystanie technologii może zachęcić do bardziej aktywnego udziału uczniów w lekcji. Dzięki odpowiednio dobranym narzędziom każda lekcja może stać się ciekawsza i bardziej

¹ <https://www.wbc.poznan.pl/Content/9794/ch02s04.html>

² <https://www.lscdn.pl/pl/publikacje/cww/dydaktyka-ogolna/5125,Zadania-nauczyciela-we-wspolczesnej-szkole.html>

³ <https://www.zycieszkoly.com.pl/artukul/jak-zainteresowac-uczniow-lekcja>

⁴ Zosiuk D. Programuję i koduję – nowe technologie wykorzystują, czyli jak wykorzystać nowe technologie w edukacji, „Inspirator” 2019, nr 1, s.7

angażująca. Uczniowie podczas takich lekcji lepiej przyswajają wiedzę, gdyż technologia wyzwała chęć do aktywnego uczestnictwa w zajęciach.⁵

Wśród wielu wytycznych, które w szkole podstawowej wskazuje nam do zrealizowania podstawa programowa nauczania informatyki, znajdują się te z dziedziny programowania i rozwiązywania problemów z wykorzystaniem komputera i innych urządzeń cyfrowych. Wymagania te są zrozumiałe, zważywszy na postęp techniczny jaki nastąpił w ostatnich latach. Dziś coraz więcej urządzeń codziennego użytku funkcjonuje z pomocą technologii komputerowej, a co za tym idzie – ktoś musi stworzyć do nich program. I nie chodzi tutaj o tworzenie zwykłych programów komputerowych. Samochód, telefon, ekspres do kawy, nawigacja samochodowa, telewizor, zegarek czy nawet pralka lub lodówka coraz częściej wymagają stworzonego przez człowieka oprogramowania.

Zasadność wprowadzenia nauki programowania w szkołach potwierdza fakt, iż nowe media, usługi i urządzenia będą wymagać od odbiorców nowych kompetencji, w tym podstawowych umiejętności programistycznych, bez których ludzie staną się biernymi konsumentami, nie znającymi cyfrowego świata i zagrożeń z nim związanych. „Programowanie rozwija kreatywność. Programowanie uczy uporządkowania, wymusza precyzję myślenia i zachowania. Programowanie to proces twórczy, którego efektem są własne dzieła. Programowanie pozwala odkrywać i rozwijać uzdolnienia dzieci, a także drzemiące w nich pasje i szczególne talenty”.⁶

Jakie zatem działania powinien podjąć nauczyciel informatyki, aby zaciekawić uczniów programowaniem? Uważam, że skuteczność swoich działań można podnieść poprzez wprowadzenie elementów zabawy, wykorzystując do tego dostępne narzędzia.

Aby wprowadzić uczniów w świat kodowania, nauczyciel może wykorzystać różnego rodzaju plansze, po których porusza się sterowany bohater programu. Na rynku dostępnych jest także wiele gier planszowych, z którymi zabawa wymaga od uczestników rozwiązywania coraz to trudniejszych zagadek z kodowania. W sklepach dostępnych jest także wiele modeli robotów, których sterowanie wymaga od właściciela napisania odpowiedniego programu.

Prowadzący zajęcia ma także do dyspozycji wiele bezpłatnych stron internetowych oraz programów z ćwiczeniami interaktywnymi. Strony te przedstawiają różne zadania, które należy wykonać, układając odpowiednio graficzne klocki z kodem programu.

Programowanie polega na wprowadzeniu do komputera pewnych oczekiwań programisty za pomocą odpowiedniego języka zrozumiałego dla komputera. Często są to języki trudne i niezrozumiałe nawet dla dorosłych. Gdy nauka programowania pojawiła się w szkole, konieczne było opracowanie specjalnych „łatwych do zrozumienia przez ucznia” języków programowania. Jednym z pierwszych „edukacyjnych środowisk do programowania”

⁵ <https://lekcjaenter.pl/blog/o-zaletach-technologii-w-procesie-nauczania>

⁶ <http://mobolab.org.pl/dlaczego-warto-uczyc-sie-programowania/>

była Logomocja. Program wymagał od ucznia wpisania kolejno odpowiednich prostych poleceń tekstowych, których uruchomienie skutkowało ruchem żółwia na ekranie, który chodząc po ekranie, tworzył rysunek. W późniejszym czasie pojawiły się inne edukacyjne środowiska programistyczne, których tekstowe komendy zostały zastąpione przez graficzne kolorowe bloki z napisami. Graficzny interface umożliwia zastosowanie ich u coraz młodszych uczniów. Wśród takich obiektowych środowisk do nauki programowania nauczyciele mogą wykorzystywać m.in.: Scratch, Baltie, Blockly, Kodable, Tynker. Wiele z dostępnych programów dostępnych jest także w wersji online. Dobrą wiadomością dla nauczycieli jest to, że wciąż pojawiają się nowe, także bezpłatne narzędzia, z których mogą korzystać.

Programowanie z pomocą takich narzędzi staje się frajdą dla ucznia. Daje mu możliwość bycia twórcą od początku do końca. To dziecko wymyśla sposób rozwiązania problemu i podejmuje próbę rozwiązania go. Nauczyciel nie musi sztywno trzymać się tematu w podręczniku w sposób, jaki jest tam zapisany. Może dodawać własne pomysły i zadania. „Programowanie może być wspaniałą zabawą, która uczy jak być sobą i jak przezwyciężać własne słabości.”⁷

Jednym z popularniejszych środowisk do nauki programowania jest Scratch. Jest to narzędzie bezpłatne, które występuje w wersji instalacyjnej oraz online. W szkole podstawowej uczniowie pracują z nim od drugiej klasy. Jedną z głównych zalet programu jest intuicyjność. Instrukcje, z których korzysta uczeń, posiadają różne kolory i są logicznie pogrupowane w kategoriach w zależności od ich funkcji. Dzięki temu widząc instrukcję w danym kolorze wiemy, do jakiej kategorii poleceń ona należy. Kształt tych elementów także podpowiada uczniowi, jak właściwie go wykorzystać. Możemy także własnoręcznie wpisywać parametry, które potrzebne nam są do osiągnięcia zamierzonego celu. W Scratchu uczniowie nie muszą się uczyć na pamięć skomplikowanych formuł. Zadaniem ucznia jest skupienie się na praktycznym rozwiązaniu problemu. Olbrzymią zaletą Scratcha jest szybka możliwość sprawdzenia efektów swojej pracy, przez co w tworzonym programie można wprowadzić poprawki lub go rozbudowywać. Rozwój Scratcha sprawił, że stał się on pewnego rodzaju serwisem społecznościowym, na którym możemy udostępniać nasze projekty innym i dyskutować o nich.

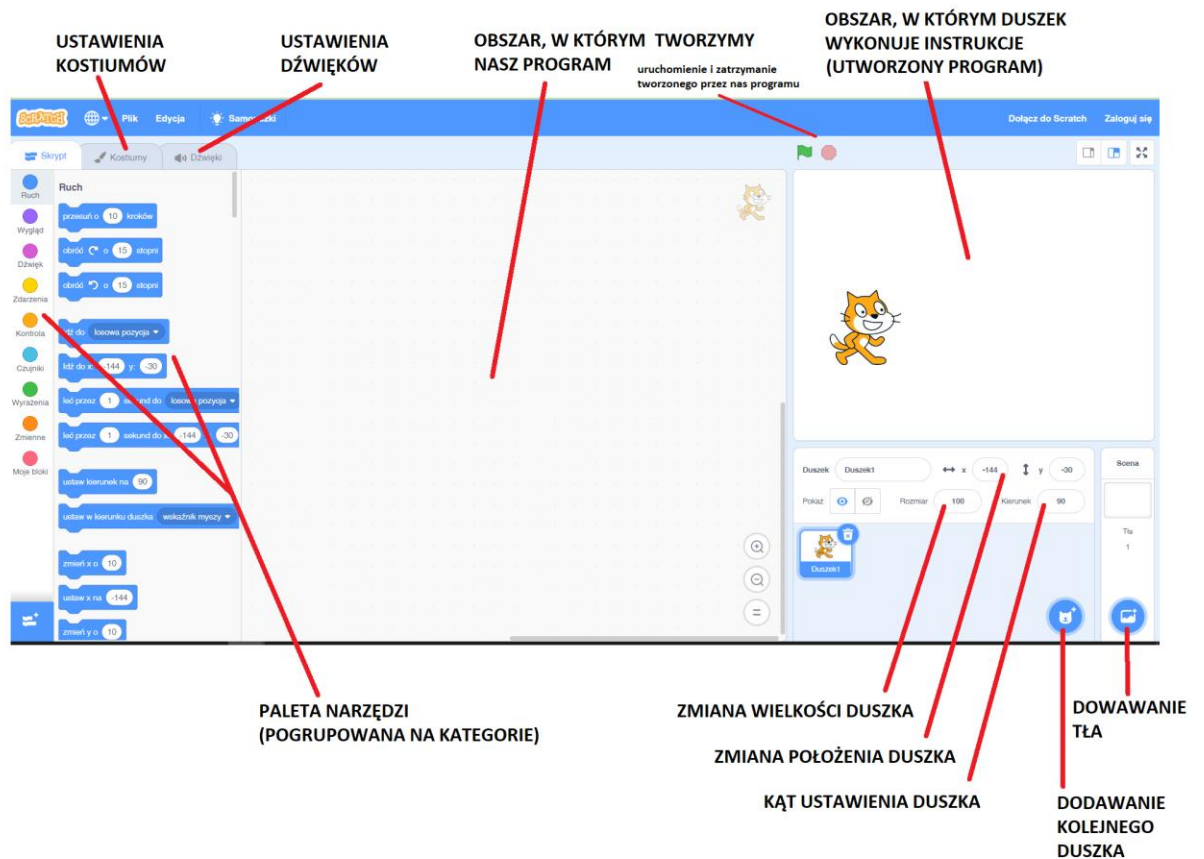
Szczegółowe elementy (tzw. konstrukcje) programowania, które powinien poznać uczeń, wskazuje nam podstawa programowa. Są one podzielone w zależności do wieku uczniów. W opracowaniu tym chciałbym przedstawić kilka prostych ćwiczeń do wykorzystania na lekcji. Nie jest to podręcznik kodowania w Scratchu. Są to pewne pomysły, w których niezbędne jest wykorzystanie części umiejętności prawdziwego programisty.

Tworząc te ćwiczenia, skupiłem się na tym, aby służyły realizacji podstawy programowej i wyczerpywały część zagadnień z tego zakresu.

⁷ <https://www.superbelfrzy.edu.pl/linux-i-wiioo/jak-zachecic-nauczycieli-informatyki-do-programowania/>

Program SCRATCH 3.0

Obszary pracy programu.



Jeśli chcemy zapisać efekty naszej pracy lub rozpocząć ją od nowa, należy wybrać polecenie z menu: Plik. Jeśli istnieje potrzeba zmiany języka, dokonamy tego, klikając na niewielki globus w prawym górnym rogu programu.

Odczytując poniższe rysunki należy, pamiętać, że kolor konkretnego użytego klocka odpowiada jego grupie kategorii i tam należy go szukać.

Przykłady zadań na: programowanie prostych historyjek wg własnych pomysłów i pomysłów opracowanych z innymi uczniami.
Korzystanie z pojedynczych poleceń oraz sekwencji sterujących obiektem na ekranie.
Wykorzystanie poleceń iteracyjnych (powtórzeń).

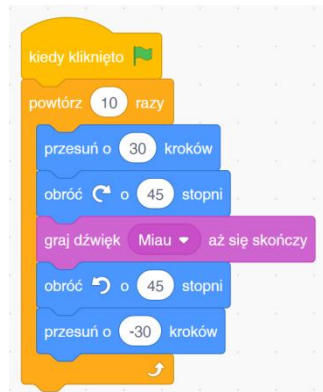
Zadanie 1.

Zakoduj kotka, aby zatańczył i zaśpiewał.

Zadanie można rozwiązać na wiele sposobów, ucznia ogranicza tylko pomysłowość i wyobraźnia. Przykład takiego programu przedstawia rys.1



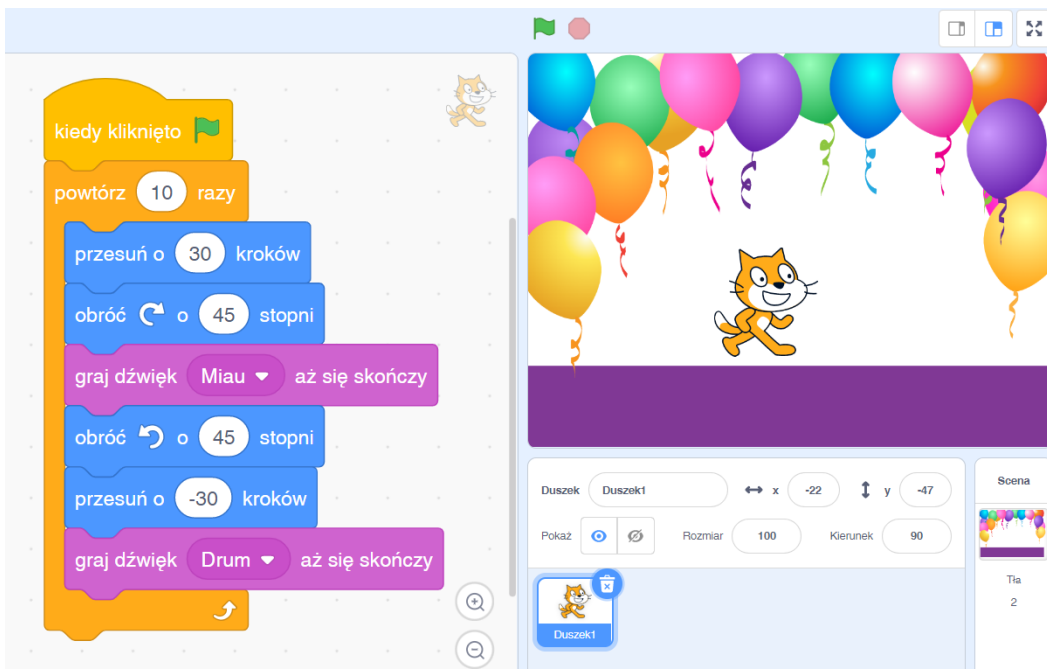
rys.1



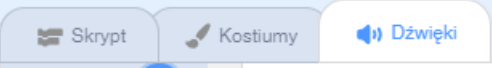


rys.2

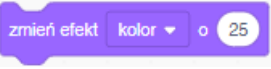
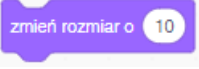
Liczbę kroków oraz kąt skręcania możemy zmieniać .

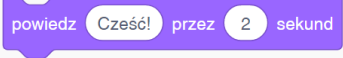
Jeśli chcemy, aby polecenie wykonywało się wielokrotnie, możemy wykorzystać pętlę.(rys.2)

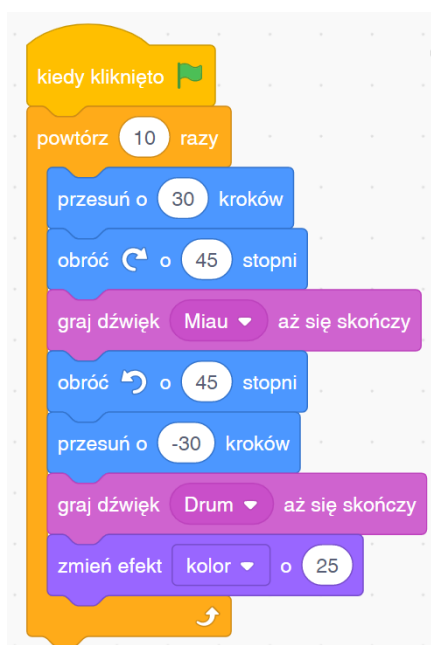


rys.3

Po wejściu w zakładkę dźwięki  oraz w menu dźwiękowym klikając na , możemy z biblioteki dodać wybrany dźwięk do naszego programu (np. perkusję). Uczniowie klikając na , mogą także dodać tło wg uznania.(rys.3).

Swój program możemy urozmaicać np. poprzez dodanie klocka *zmień kolor*  lub *zmień wielkość*  (rys.4). Przy naszym duszku mogą

pojawiać się także różne chmurki z napisami np.: 



rys.4

Ciekawsze historyjki możemy wykonać, dodając kolejnego duszka.

Aby dodać kolejnego duszka, należy go dodać z biblioteki duszków poprzez plusik w prawym dolnym rogu ekranu.

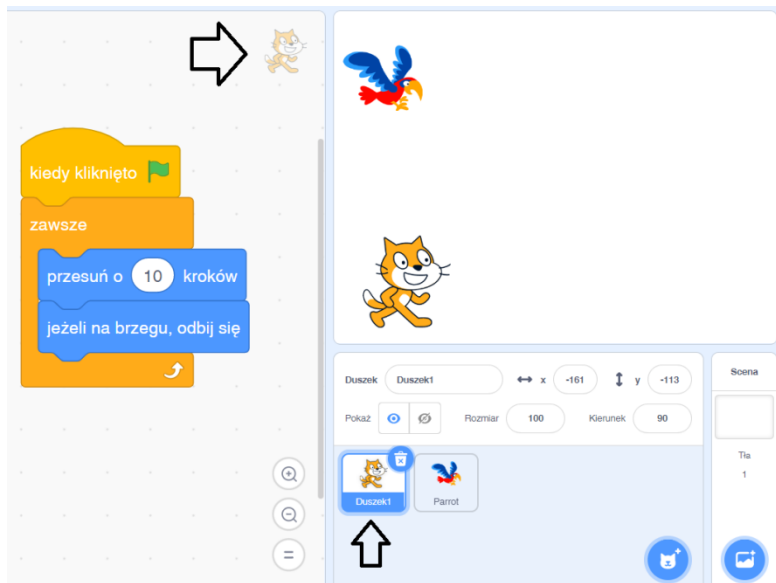
Zadanie 2.

Zakoduj dwa duszki, aby spacerowały po ekranie i wydawały dźwięki.

Rozmiar każdego dodanego duszka możemy zmienić, wpisując odpowiedni parametr w polu

Rozmiar 100

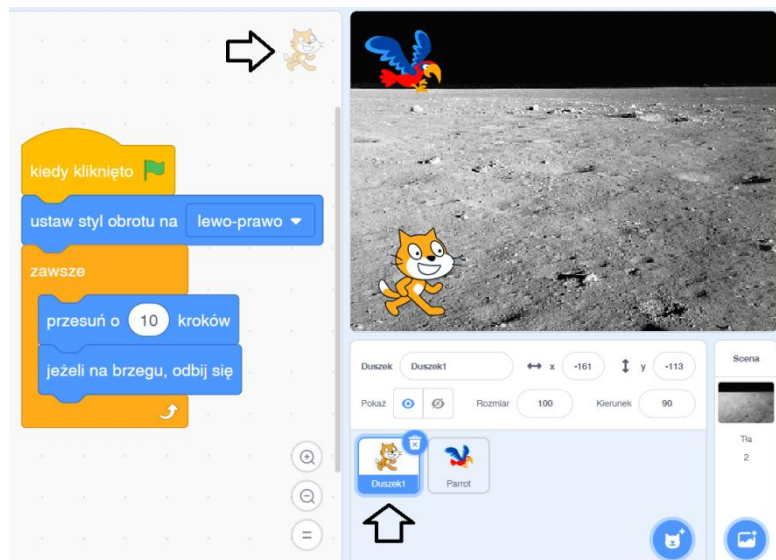
Przykład rozwiązania takiego zadania przedstawia rys.5. Ponieważ użyta została *pętla zawsze*, skutkiem podanego rozwiązania będzie ciągły ruch kota oraz odbijanie się od krawędzi ekranu.



rys.5

Czarne strzałki ukazują nam, który duszek jest właśnie kodowany (rys.5).

Do historyjki możemy także dodać wybrane tło. Aby uniknąć obracania się duszka do góry nogami, program należy uzupełnić o instrukcję z rys 6.



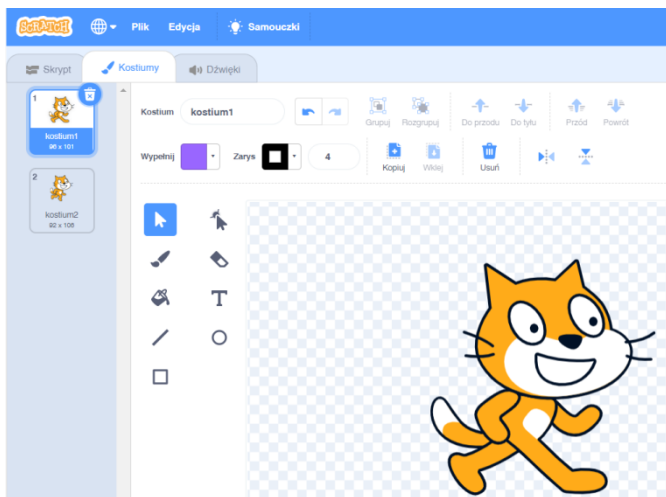
rys. 6

W zakładce kostiumy


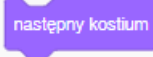


możemy sprawdzić, na ile

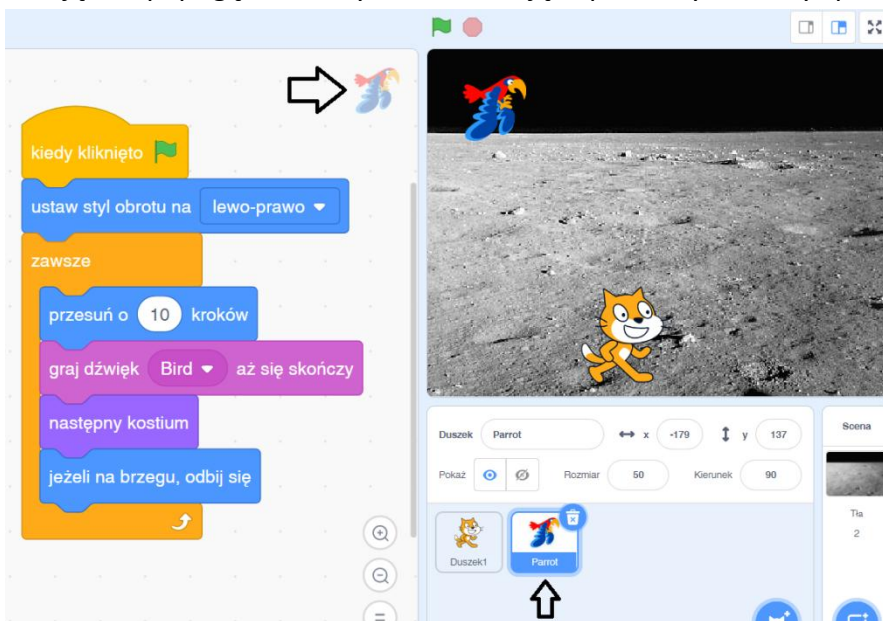
sposobów może być przedstawiana nasza postać. (rys.7)



rys. 7

W kategorii *wygląd*  możemy odnaleźć instrukcję . Efektem jej wykonania będzie wrażenie ruchu naszej postaci.

Klikając na papugę, możemy zakodować ją w podobny lub inny sposób.(rys.8)

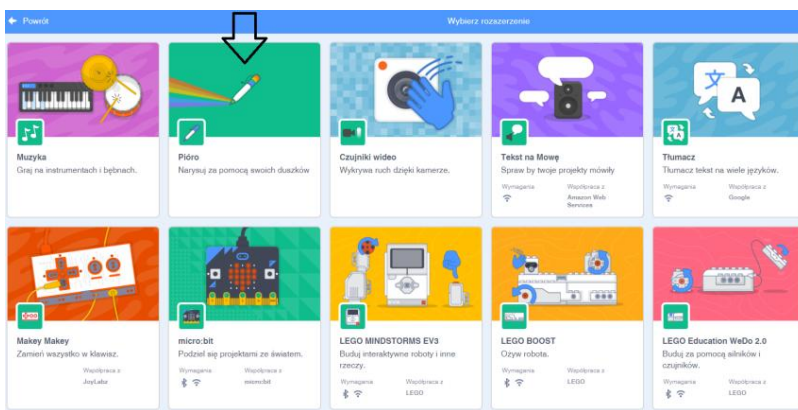
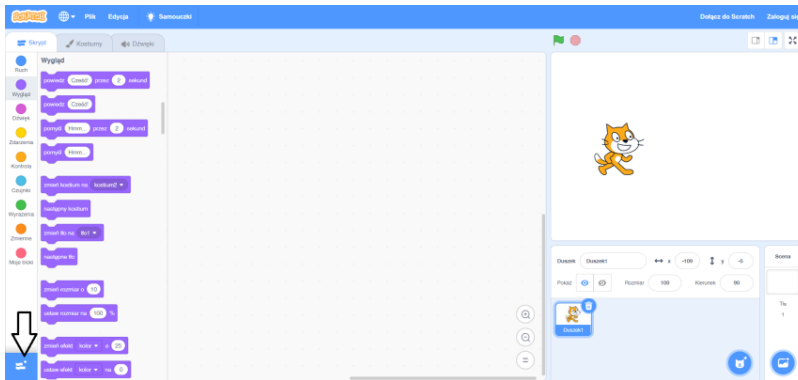


rys.8

W tym samym programie papuga będzie latała o 10 kroków do przodu, odbijała się od krawędzi, wydawała dźwięki i machała skrzydłami (zmieniała kostium).

Takie historyjki można urozmaicać o własne pomysły (zmienianie koloru, dodanie chmur z napisami itp.).

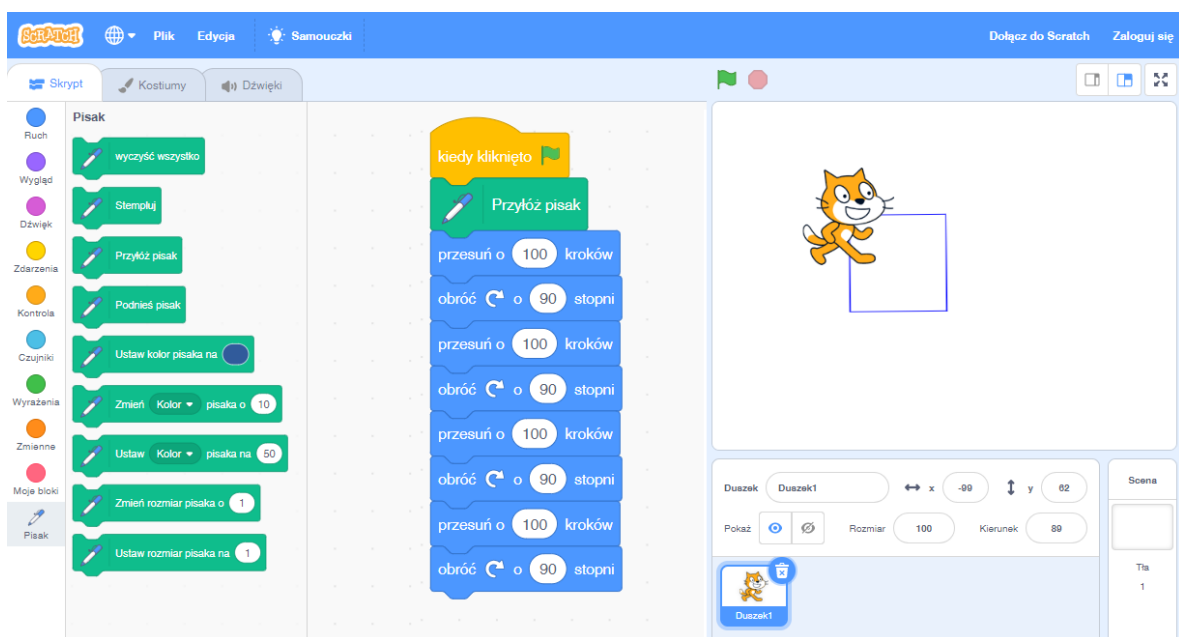
Układając zadania dla uczniów, możemy wykorzystać jedno z dodatkowych grup narzędzi. Do programu możemy dodać grupę narzędzi „pióro” (rys. 9) . Umożliwia ona tworzenie programów, w których nasz duszek rysuje różne figury.



rys.9

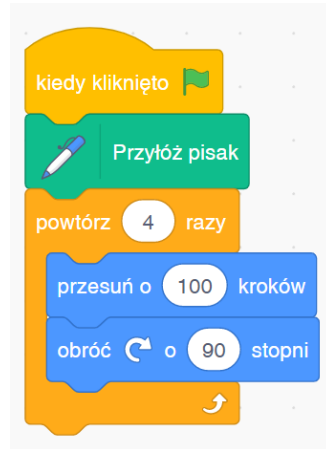
Zadanie 3.

Zakoduj duszka, aby na ekranie wykonał kwadrat.



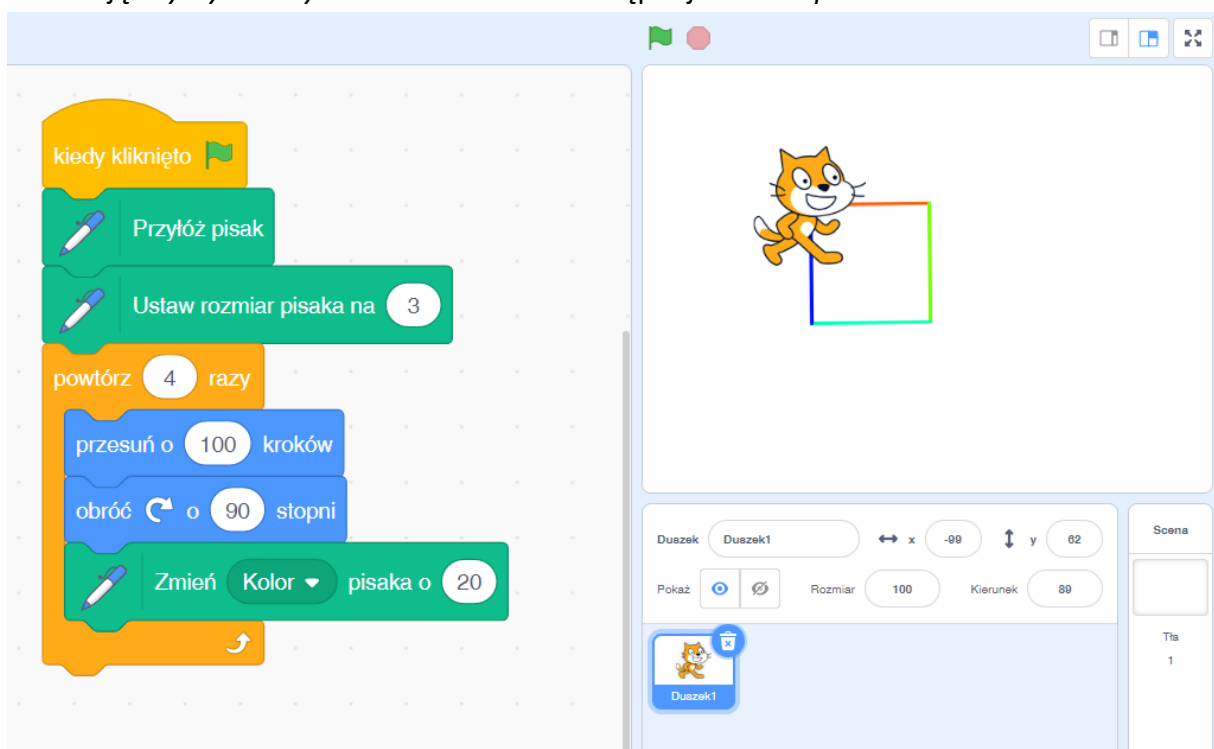
rys.10

Rysunek 10 przedstawia przykład, który uczniowie mogą opracować krok po kroku. Program ten należy w końcowej fazie skrócić do poleceń, które mogą być wykonane w pętli. Jeśli kwadrat ma cztery boki, może wyglądać jak na rys.11.



rys.11

Zadanie możemy urozmaicić o dodawanie grubości pisaka oraz kolorów (rys.12). Gdyby w trakcie ćwiczenia było potrzebne wyczyszczenie sceny z pisaka, należy kliknąć na instrukcję *wyczyść wszystko* dostępnej w dziale *pisak*.



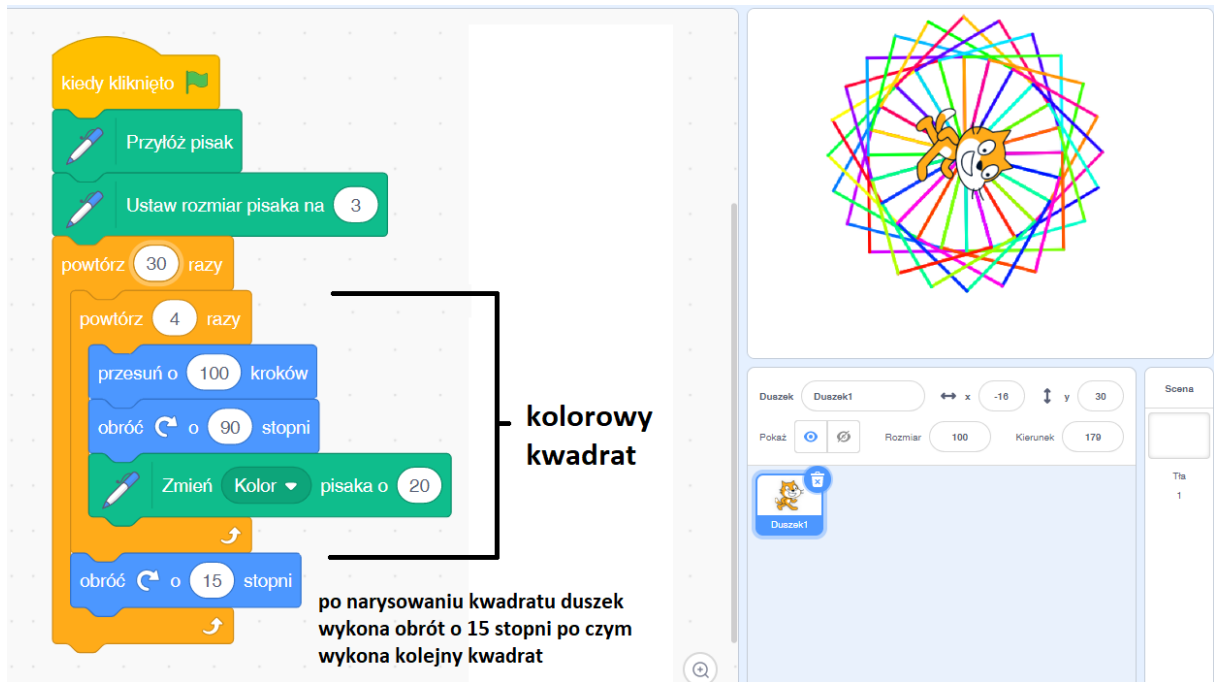
rys.12

Gdyby nasz duszek po narysowaniu kwadratu zmieniał swoją pozycję, mógłby rysować je w różnych miejscach sceny. Powyższe zadanie jest doskonałą bazą do wykonania zadania nr 4.

Zadanie 4.

Zakoduj duszka, aby na ekranie wykonał serwetkę lub kolorową serwetkę.

Kluczem do rozwiązania zadania jest uświadomienie uczniom, że jeśli duszek zmieni swoją pozycję, będzie mógł narysować kwadrat ponownie w innym miejscu, np. rysuje wielokrotnie ten sam kwadrat, ale po każdym z nich lekko zmienia kierunek.



rys.13

Podobne zadania z piórem można rozwiązywać na różne sposoby. Korzystając z różnej liczby kroków oraz obracając się o różne kąty, nasz duszek może wykonywać inne figury lub kompozycje np. : ośmiokąt, gwiazdkę itp.

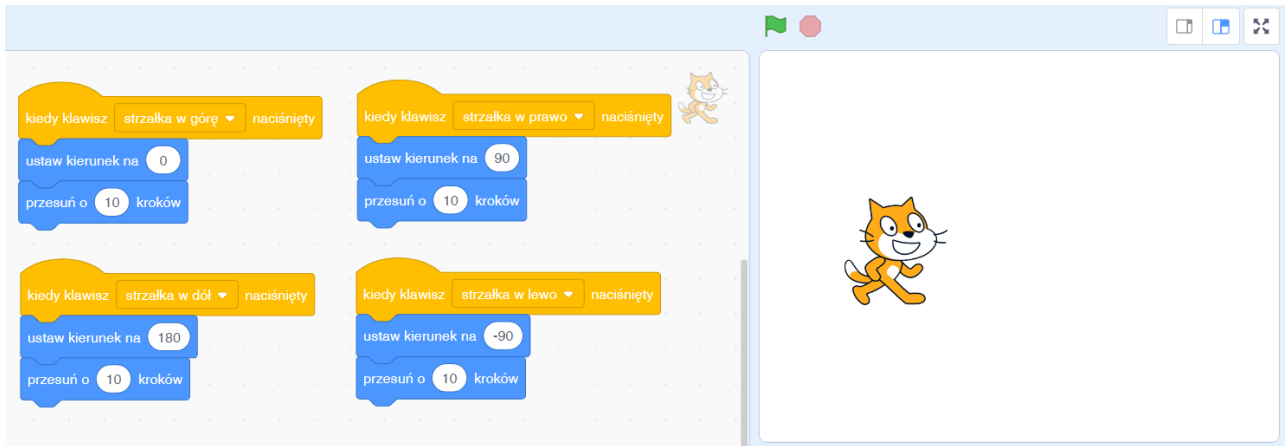
Zadanie 5.

Zakoduj duszka (lub duszki) w sposób umożliwiający poruszanie się nim na ekranie za pomocą klawiatury lub myszki.

Rozwiązując to zadanie, zamiast skorzystania z uruchamiającej program od razu zwykłej flagi

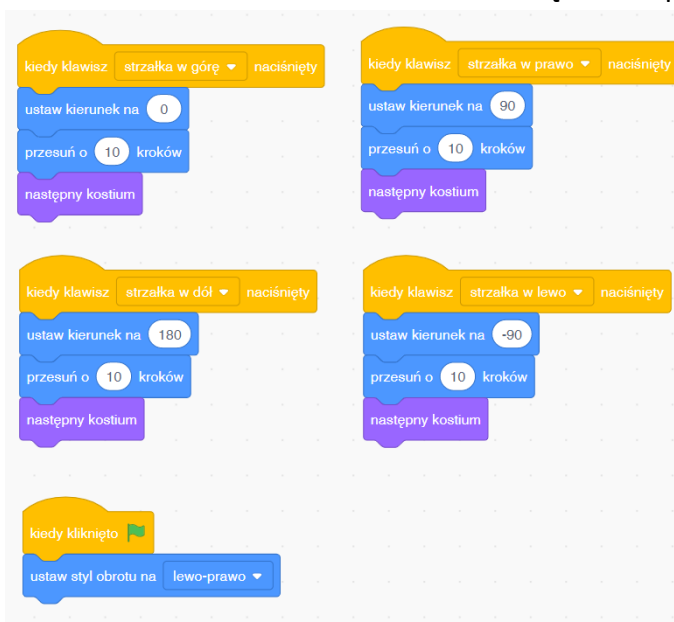


możemy wykorzystać inne narzędzie (rys.14)



rys.14

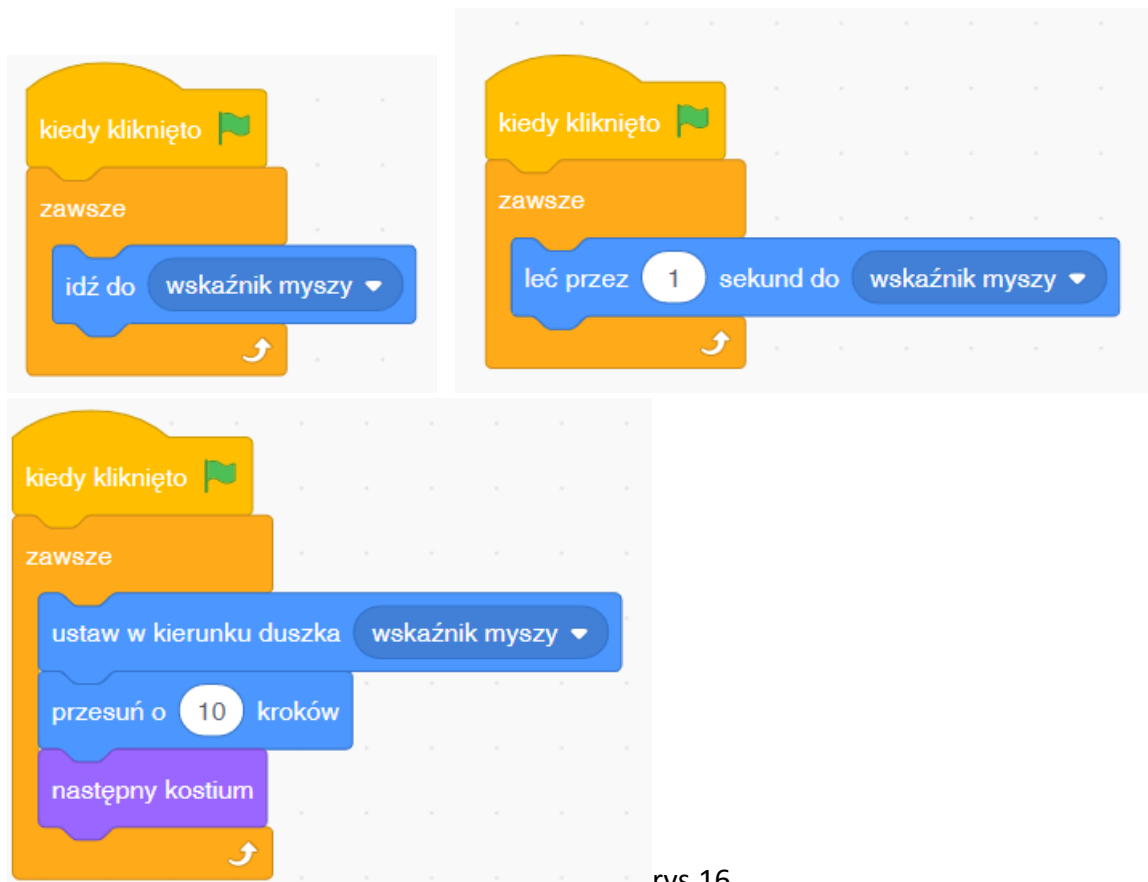
Powyższy rysunek przedstawia cztery sekwencje odpowiedzialne za ruch w konkretnym kierunku. Aby zmienić parametr kierunku, wystarczy na niego kliknąć i ustawić go strzałką. Możemy także zmienić rodzaj naciskanego klawisza. Cztery sekwencje umożliwiają wykonanie się krótkich programów w momencie naciśnięcia wybranego przycisku. Aby ruch naszej postaci był efektywniejszy, wykonanie zadania można uzupełnić o dodanie zmiany kostiumu oraz ustawienie nie obracania się duszka podczas zmiany kierunku. (rys.15)



rys.15

Sposobów na sterowanie duszkiem za pomocą kursora myszki jest kilka.

Poniżej przedstawiam przykłady, które możemy rozbudować w zależności do naszych oczekiwań.



rys.16

Przykłady zadań na:

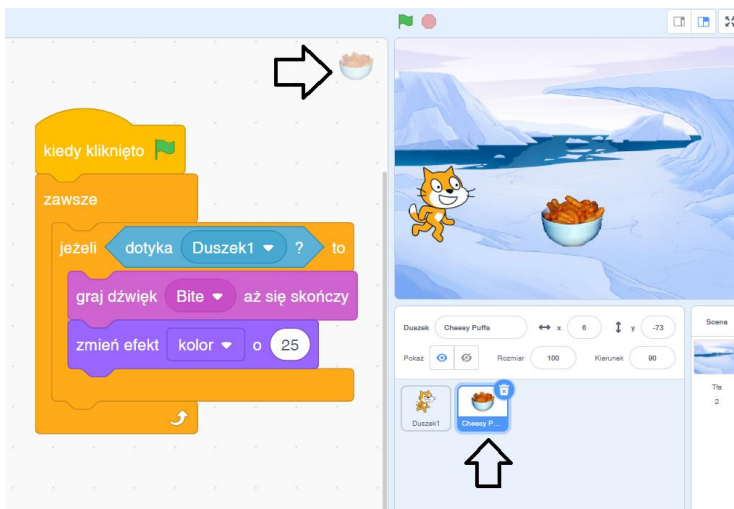
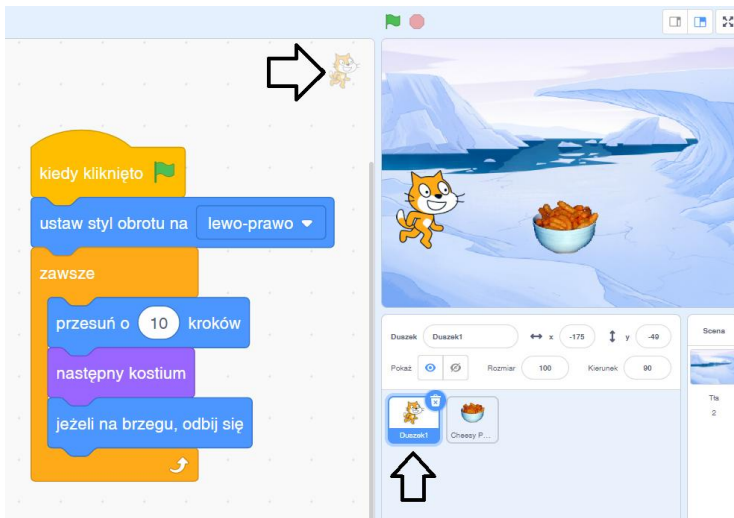
Programowanie z wykorzystaniem instrukcji warunkowych oraz zastosowania zmiennych.

Zadanie 5.

Wykonaj program z wykorzystaniem dwóch duszków.

Jeden z nich powinien spacerować po ekranie i jeżeli spotka się z drugim duszkiem, wykona się jakaś akcja, np. zmiana koloru lub duszek wyda dźwięk.

Podczas kodowania należy zwrócić uwagę, dla którego duszka układamy instrukcje.



Dla instrukcji *jeżeli* należy dodać z kategorii **Czujniki** odpowiedni warunek np. :

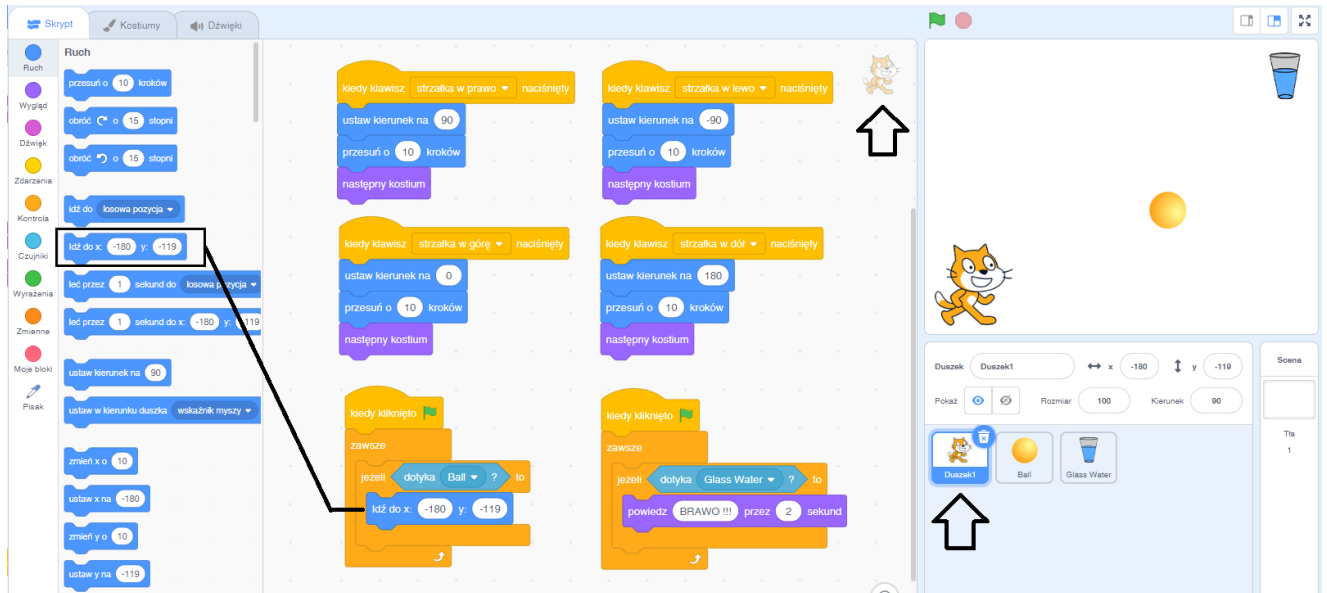
Dotyka jakiegoś duszka , dotyka krawędź, jakiegoś koloru itp.

W powyższym przykładzie nazwa Duszek 1 to właśnie nazwa kota.

Wszystko, co umieścimy wewnątrz instrukcji *jeżeli*, wykona się tylko, gdy spełniony zostanie określony przez nas warunek. W podanym przykładzie umieszczony został dźwięk Bite, który znajduje się w bibliotece dźwięków.

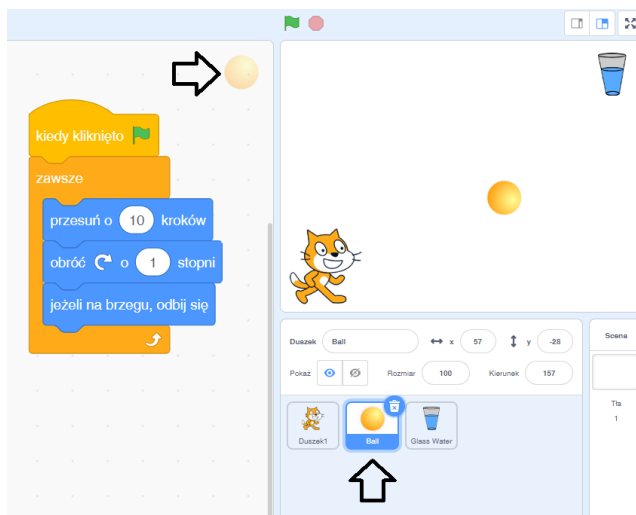
Zadanie 6.

Wykonaj grę z wykorzystaniem trzech duszków.
Pierwszy sterowany klawiszami będzie musiał dojść do drugiego,
który będzie ustawiony w rogu planszy.
Trzeci z duszków będzie mu w tym przeszkadzał.



rys. 19

Na powyższym przykładzie cztery górne sekwencje odpowiadają za sterowanie obiektem na ekranie. Dwie dolne to instrukcje warunkowe. Pierwsza określa skutek zetknięcia z piłką (która będzie się poruszać), druga określa akcję, jaka wykona się, gdy kotek dojdzie do szklanki. Zastosowanym przykładem skutku spotkania kota z piłką jest instrukcja z wpisanymi współrzędnymi miejsca startu kotka. Jeśli więc spacerujący po planszy kot dotknie latającej piłeczki, to przeniesie się do punktu, z którego wystartował (lewy dolny róg). Wielkość duszków można zmniejszyć, aby gra była ciekawsza. Dla szklanki z wodą kodu wprowadzać nie musimy. Przykład instrukcji dla mającej latać piłeczki znajdziemy na poniższym rysunku.

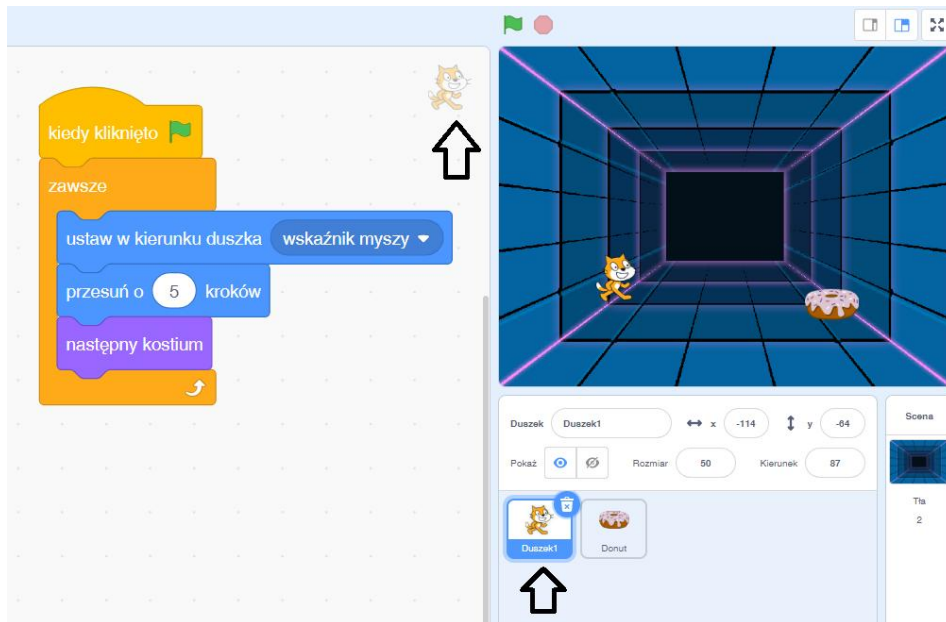


rys.20

Zadanie 7.

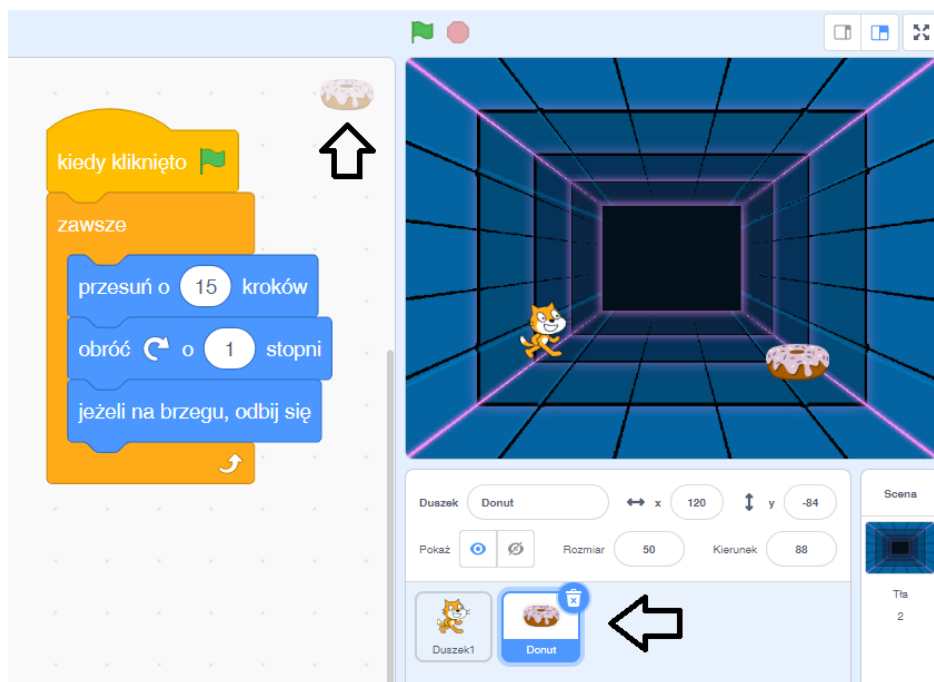
Wykonaj grę, w której duszek (kotek) sterowany kursorem myszy będzie gonił poruszające się po planszy ciastko (lub myszkę). Jeśli je dotknie, zdobywa punkt. Gra kończy się, gdy zdobędzie określoną liczbę punktów.

Zadanie możemy podzielić na dwa etapy : najpierw przygotujemy grę, a potem dodamy punktację. Dla duszka, który będzie podążał za kursorem myszy, program może wyglądać tak:



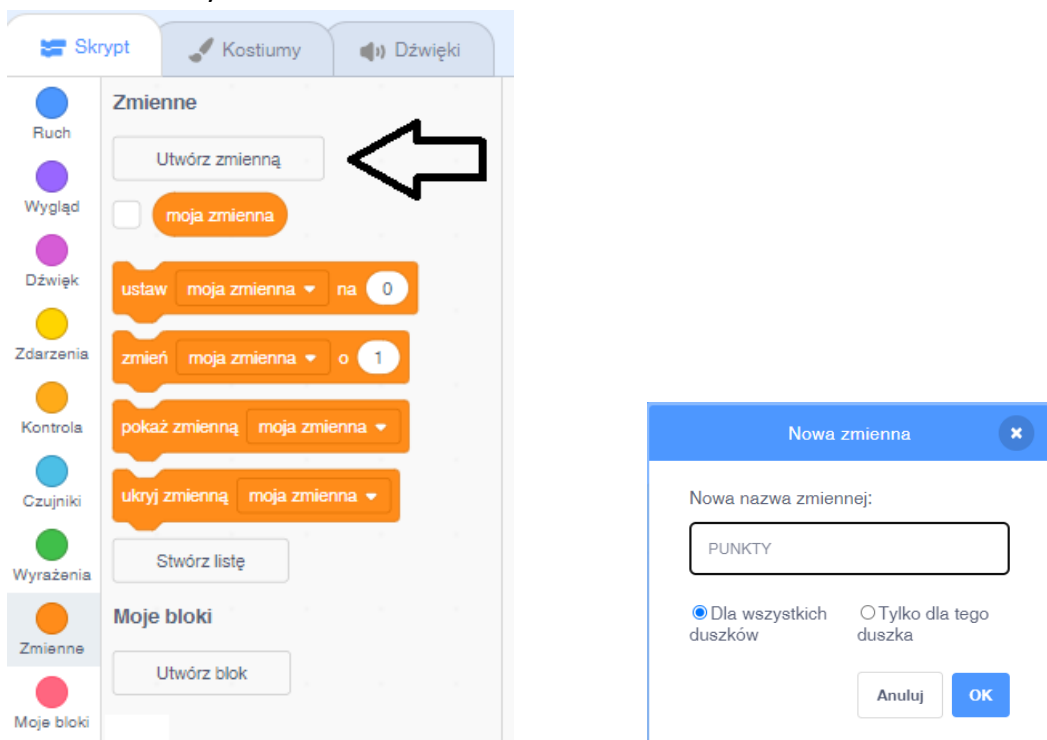
rys.21

Drugi duszek będzie się poruszał samodzielnie, zmieniając kierunek i odbijając się od ścianek planszy. Aby gra była ciekawsza, oba duszki powinny być zmniejszone.



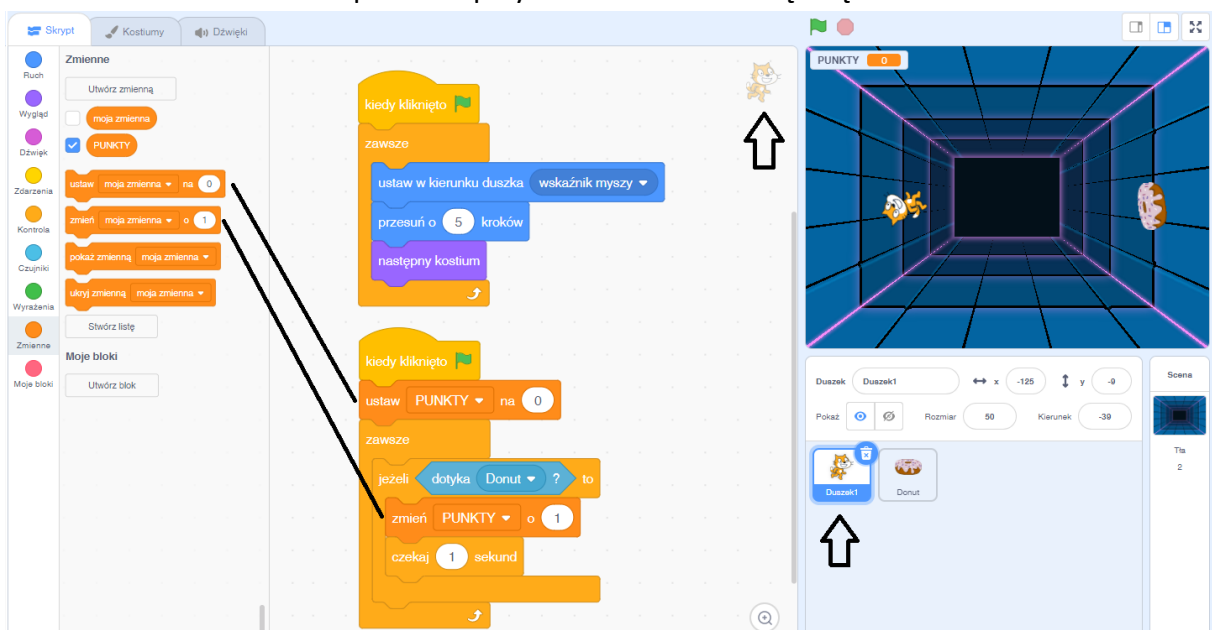
rys.22

Chcąc wprowadzić zliczanie punktów, należy zastosować zmienną. Klikając na grupę narzędzi *Zmienne*, wybieramy utworz zmienną. Nowej zmiennej nadajemy nazwę np.: "PUNKTY" i klikamy OK.

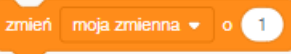



rys.23

Po stworzeniu i nazwaniu naszej zmiennej możemy z niej korzystać. Dla naszego kota należy wykonać sekwencję zliczającą punkty. W kloku *ustaw moja zmienna* za pomocą strzałeczki należy zmienić nazwę. Na początku każdej gry punkty powinny wynosić zero. Aby gra działała poprawnie, to po każdym zetknięciu się duszków zmienna PUNKTY powinna przybrać wartość o 1 większą.






rys.24

Po kloku  została zastosowana instrukcja . Zapobiega ona naliczaniu wielu punktów przy jednym złapaniu ciastka. Ostatnim etapem jest określenie, co ma się wydarzyć, jeśli kotek zdobędzie określoną liczbę punktów. Przykład takiej sekwencji przedstawia rysunek 25.



rys.25

Liczba punktów zapisywana jest w zmiennej PUNKTY. Program powinien przez cały czas sprawdzać, czy zmienna  nie jest równa określonej przez nas liczbie , której osiągnięcie zakończy grę. W zielonym dziale wyrażenia dostępne są instrukcje dotyczące obliczeń. Liczba, którą określimy jako wygraną, może być dowolna.

Zastosowanym przykładem spełnienia warunku  jest dźwięk dodany z biblioteki, chmurka BRAWO oraz ponowne ustawienie zmiennej PUNKTY na zero.

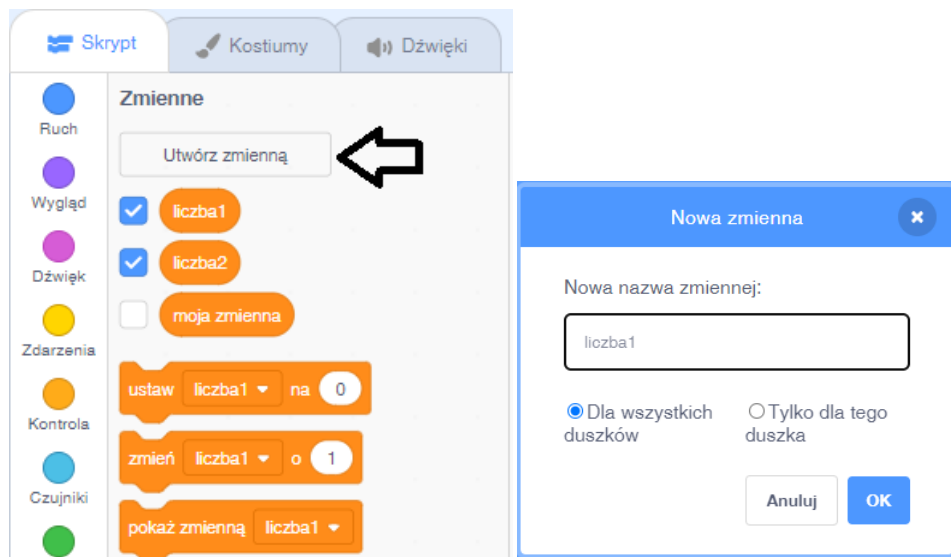
**Przykłady zadań na:
Zastosowanie instrukcji wejścia / wyjścia**


Tworząc instrukcje wejścia musimy pamiętać, że dane wejściowe, które w momencie działania naszego programu będziemy wprowadzać do komputera, będzie on musiał gdzieś przechować. Każde dane wejściowe komputer będzie przechowywał w określonej zmiennej. Jeśli jej nie ma, należy ją utworzyć i nazwać. Przechowywane w zmiennych dane mogą być przetwarzane i wyprowadzane.

Zadanie 8.

Wykonaj kalkulator, który poprosi o dwie liczby, doda je i poda wynik.

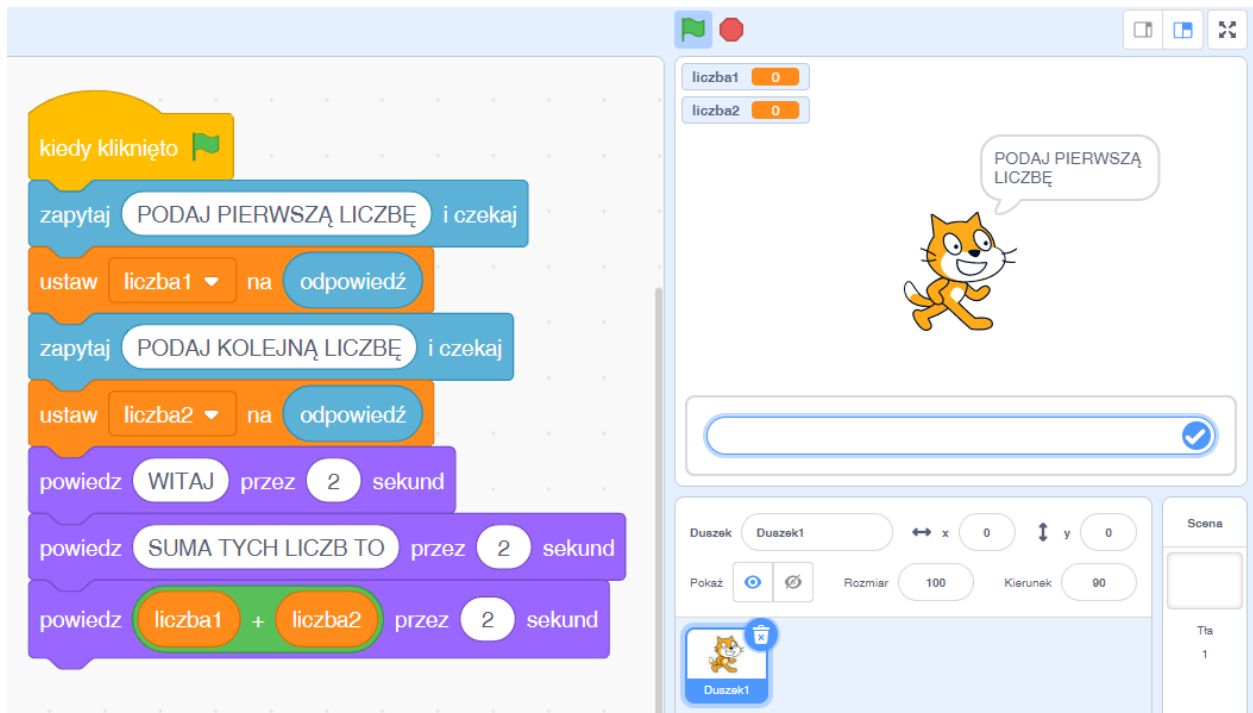
Aby komputer mógł przyjąć dwie liczby, każdą z nich będzie musiał przechować z osobnej zmiennej. Należy je utworzyć i nazwać, np. liczba1, liczba2, a, b, x...



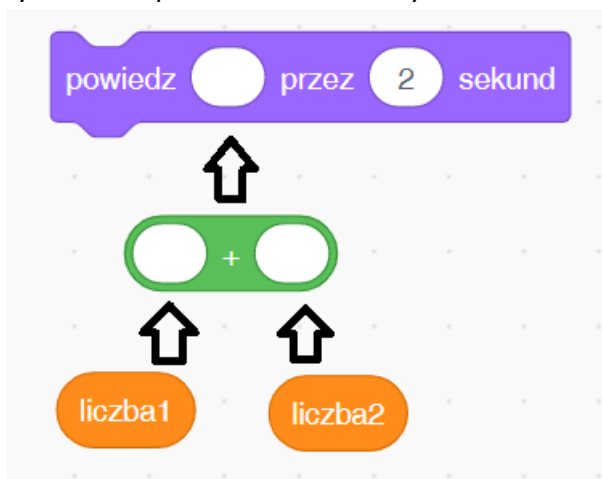
Po utworzeniu obu zmiennych należy skorzystać z instrukcji wprowadzania dostępnych w jasnoniebieskim dziale czujniki  .



Nasz program uzupełniamy o instrukcje zgodnie z zamierzeniami.
Przykład przedstawia poniższy rysunek.



Po podaniu dwóch liczb kotek się przywita i powie sumę wprowadzonych do zmiennych liczb. Ostatni klocek tworzymy korzystając z instrukcji dostępnych w obliczeniach oraz symbolu odpowiednich zmiennych.

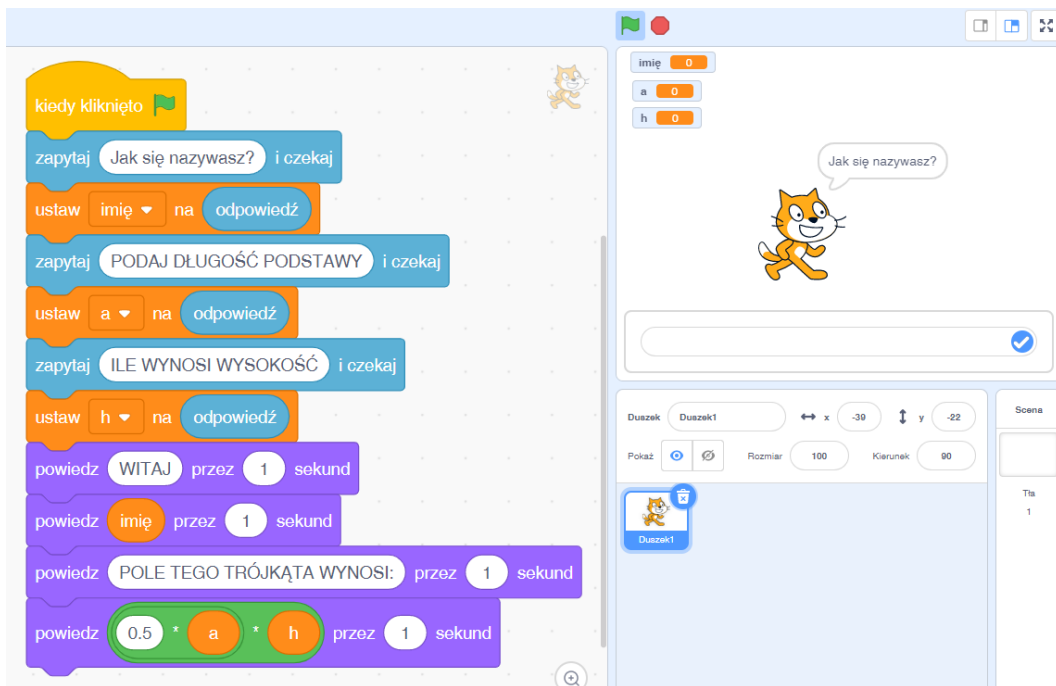


Zadanie 9.

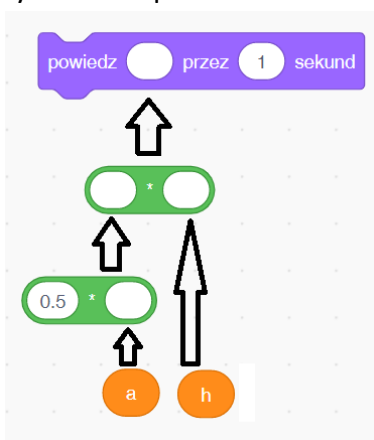
Wykonaj program, który obliczał będzie pole trójkąta.

Polecenie dodatkowe: wprowadź instrukcję tak, aby duzek zapytał cię o imię, które zapamięta.

Każde z wprowadzonych danych musi być przez komputer zapamiętane w osobnej zmiennej. Danymi mogą być liczby lub wyrazy (np. imiona). Aby program mógł przyjąć 3 parametry, musimy utworzyć 3 zmienne. Posiadając zmienne, może obliczyć pole trójkąta zgodnie ze wzorem. Ułamek $\frac{1}{2}$ w środowisku Scratch określamy sposobem dziesiętnym 0.5, używając zamiast przecinka kropki.



Ostatni klocek tworzymy, korzystając z instrukcji dostępnych w zielonych obliczeniach oraz symbolu odpowiednich zmiennych.



określa ono wzór na pole: $P=1/2*a*h$