

Programuję samodzielnie

Wanda Jochemczyk, Katarzyna Olędzka

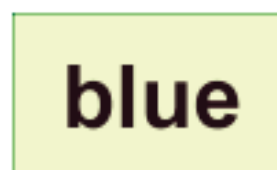


Rysunek 1. Kotek – symbol środowiska Scratch zaprasza do pracy na <https://scratch.mit.edu>

Obecnie obserwujemy nie tylko ogólny wzrost zainteresowania samym programowaniem, ale także liczne próby korzystania z elementów programowania przez osoby, które nie są informatykami. Przykładem może być wykorzystanie automatycznego przetwarzania tekstów do badań naukowych opartych na języku, tworzenie i obróbka grafiki z wykorzystaniem różnych algorytmów przez artystów czy też programistyczna synteza dźwięku i muzyki przez osoby z branży muzycznej. Programować mogą i mali – nasi uczniowie – w specjalnie przygotowanych dla nich środowiskach, i trochę starsi – my nauczyciele. Możemy przygotowywać różne projekty przydatne w naszej pracy dydaktycznej. Przykładem takiego projektu jest gra w pary, którą można przygotować samodzielnie w środowisku Scratch.

Zabawy tego typu kształtują umiejętność logicznego myślenia oraz ćwiczą spostrzegawczość, a przede wszystkim umiejętność zapamiętywania. Jednak dużo więcej uczymy się, przygotowując taką aplikację – rysujemy kształty, opisujemy ruch i reakcję na zdarzenia. Ważne jest to, że zapisujemy w sposób formalny nasze pomysły i przemyślenia. Niejako przy okazji programowania rozwijamy umiejętności psychologiczne, socjologiczne i mamy okazje do przemyśleń o charakterze etycznym. Uczymy się myśleć w kategoriach matematycznych, wykorzystując algebrę i geometrię.

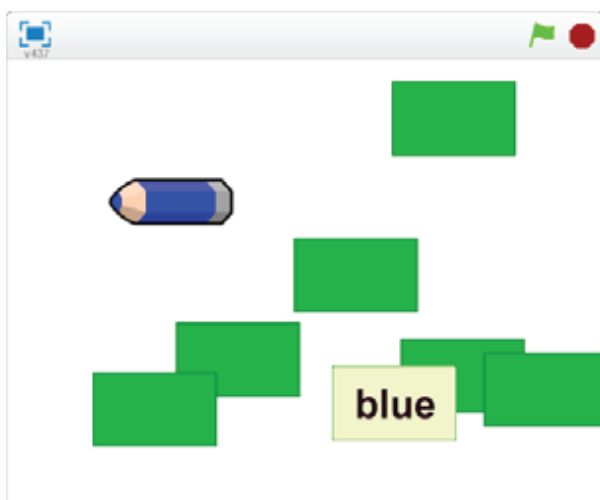
Wracając do naszej aplikacji, opiszmy przebieg gry – najpierw w języku naturalnym, potem za pomocą bloczków w Scratch. W naszym przykładzie trzeba dopasować kredkę z odpowiednim kolorem do napisu w języku angielskim oznaczającego ten kolor.



Rysunek 2. Para kart – rysunek i odpowiadający mu napis

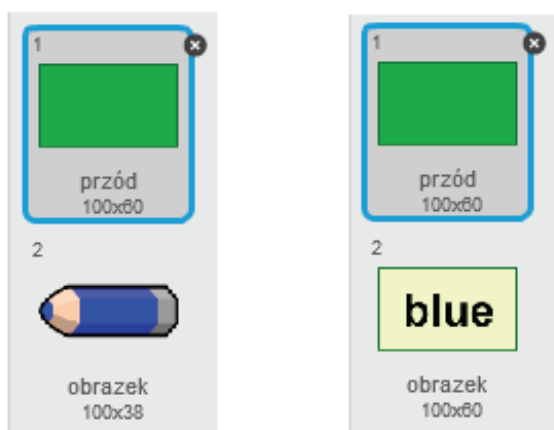
Po kliknięciu w start – zieloną flagę – nastąpi uruchomienie projektu. Pojawią się wszystkie karty, będą one odwrócone tak, by nie było napisów i obrazków, czyli przodem. Następnie gracz klika w dwie karty i odkrywa obrazki. Na rysunku zostały odkryte dwie karty stanowiące parę. Czytając ten artykuł, możemy od razu się chwilę pobawić –

zapraszamy na stronę Scratch. Gotowe projekty znajdują się pod adresem <https://scratch.mit.edu/studios/1342172> – proszę poklikać i spróbować schować wszystkie karty będące parami.



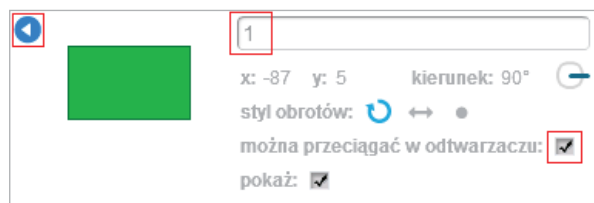
Rysunek 3. Przykładowy zrzut z ekranu w trakcie gry

Zacznijmy w takim razie realizować ten projekt. Na początek przygotujmy duszki będące odpowiadającymi sobie parami. Każdy z nich powinien mieć dwa kostiumy – rysunek **przodu** (zielony prostokąt o wymiarach 100x60) oraz odpowiedni **obrazek**. Nazwy kostiumów nie są przypadkowe, są identyczne dla wszystkich duszków. Będziemy się do nich odwoływać podczas tworzenia projektu.



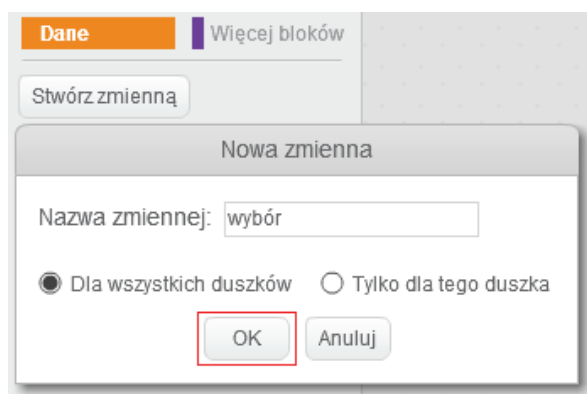
Rysunek 4. Duszek 1 (po lewej) i Duszek 2 (po prawej)

Musimy też nazwać odpowiednio kolejne duszki tak, aby ułatwić komunikację między nimi. Dlatego duszek pierwszy ma nazwę 1, duszek drugi 2 itd. Drugą właściwością duszka, którą należy ustawić, jest opcja **przeciągania** duszka w odtwarzaczu.



Rysunek 5. Zmiana nazwy i ustawienie stylu obrotów

Musimy też w zakładce **Dane** utworzyć zmienną o nazwie „wybór”, której wartość będzie znana wszystkim duszkom, będziemy w niej przechowywać numer pierwszego klikniętego duszka. Podczas tworzenia projektu warto zawsze mieć widoczną wartość zmiennej (zaznaczony przy nazwie zmiennej), aby obserwować prawidłowość działania skryptów.



Rysunek 6. Tworzenie nowej zmiennej

To są nasze ustawienia wstępne, teraz zaczynamy tworzyć skrypty. Pierwszy z nich będzie **dla sceny** – będziemy uruchamiany po kliknięciu w **zieloną flagę**, a w nim będziemy ustawiać wartości początkowe dla gry – zmiennej **wybór** przypisujemy wartość 0, gdyż nic nie zostało wybrane, oraz tworzymy zmienną **ile_jeszcze**. Będzie ona miała wartość zależną od liczby par kart do odkrycia w danym projekcie. W naszej aplikacji są 4 pary, czyli 8 kart. Zmienna **ile_jeszcze** po kliknięciu w zieloną flagę ma wartość 4.



Rysunek 7. Skrypt zielonej flagi dla strony

Następnie budujemy skrypty dla poszczególnych duszków. Po kliknięciu w zieloną flagę duszek powinien przemieścić się na losowo wybraną przez siebie pozycję, zmienić kostium na **przód**, czyli zielony prostokąt, i się pokazać.



Rysunek 8. Skrypt zielonej flagi dla duszków

Dodatkowo przygotujmy skrypt, który będzie zmieniał obrazek lub napis w czystą kartę – przód. Będzie on wyglądał następująco:



Rysunek 9. Komunikat dla duszka 1 – zakryj kartę!

Teraz kolejno budujemy skrypty duszka 1 i 2. Analizujemy to, co będzie się działo po kliknięciu w jego postać. Po pierwsze duszek powinien się odkryć – pokazać swój obrazek lub tekst. Po drugie, jeśli będzie to pierwsza kliknięta karta (**wybór** ma wartość **0**), to zmienna ta przyjmuje wartość taką, jaką duszek ma nazwę. Przykładowo, po kliknięciu w kartę o nazwie 1, zmienna **wybór** ma wartość **1**.

W przypadku gdy mamy już odstoniętą jedną kartę i klikamy w drugą, to sprawa jest bardziej skomplikowana. Może to być przecież zła karta lub karta od pary. Jak to poznać? Tworząc duszki, nazywalimy je kolejno, czyli parę tworzą duszki 1 i 2, 3 i 4 itd... Teraz przygotujmy cały skrypt. Wszystkie akcje po kliknięciu w duszka polegają na nadawaniu lub odbieraniu komunikatów.



Rysunek 10. Skrypty główne dla duszków 1 i 2. Proszę porównać

Po przygotowaniu skryptów dla duszka 1 i 2, przetestujmy je. Powinny już działać prawidłowo, to znaczy po kliknięciu w duszka 1 i 2 (w dowolnej kolejności) oba duszki powinny się chować. Pozostaje nam w identyczny sposób dorobić kolejne pary duszków, nazywając je odpowiednio 3, 4 itd. Zanim jednak zaczniemy kopiować i modyfikować skrypty dla kolejnych duszków, należy jeszcze duszkom 1 i 2 dodać fragment dotyczący zakończenia gry – klocki wstawiamy po klocku **ukryj**

Jeśli zostaną odkryte wszystkie pary, chcemy, aby pokazały się gratulacje. W tym celu tworzymy kolejnego duszka, który po nadaniu komunikatu **koniec** pokazuje się, a po kliknięciu w zieloną flagę chowa się.



Rysunek 12. Duszek gratulacje!



Rysunek 13. Skrypty dla duszka gratulacje

Mając gotowy mechanizm takiej gry, można podłączyć dowolne pary obrazków – i mamy ciekawy projekt edukacyjny, który możemy wykorzystać na zajęciach dydaktycznych.

Seymour Papert napisał: *Widziałem tysiące reklam oprogramowania z obietnicami typu – taka dobra zabawa, że Twoje dziecko nie będzie wiedziało, że się uczy. (...) Jestem przerażony tym przestaniem. Ta reklama sugeruje, że uczenie się jest jakąś wstrętną pigułką, którą trzeba osłodzić zabawą lub grą. (...) Natomiast jedną ze wspaniałych rzeczy, jaką komputer może zrobić, jest odwrócenie tego*



Rysunek 11. Dodatkowy fragment skryptu

tak, aby doświadczać pewnego rodzaju przyjemności, jaką się widzi, obserwując dziecko lub uczonego. Oboje uczą się cały czas i rzeczywiście to lubią².

Nie bójmy się więc programować, zachęcajmy nasze dzieci do twórczej, chociaż czasami wymagającej nauki. Dziecko w roli badacza potrafi wspaniale się bawić, doznaje wielu wrażeń, ale jednocześnie uczy się, odkrywając to, co nieznanne, a zarazem fascynujące. Pozostaje więc ufać, że niemały wysiłek włożony przez nauczycieli w stawianie uczniom odpowiednich wyzwań i wspieranie ich nie pozostanie bezowocny.

Bibliografia i netografia

1. Jochemczyk W., Olędzka K. *Czego uczymy się, programując. Refleksje o nauczaniu programowania w szkole podstawowej*, Meritum nr 3(34)/2014, s. 82-92.
2. Jochemczyk W., Olędzka K. *Programowanie na zajęciach komputerowych*, XI Konferencja Informatyka w Edukacji, Toruń, 1-2 lipca 2014.
3. Papert S. *The Connected Family: Bridging the Digital Generation Gap*, Longstreet Press, 1996.
4. Strona domowa i portal społecznościowy środowiska i języka Scratch, <http://scratch.mit.edu>

Wanda Jochemczyk i Katarzyna Olędzka są nauczycielkami konsultantkami w Ośrodku Edukacji Informatycznej i Zastosowań Komputerów w Warszawie.

² Papert S. *The Connected Family: Bridging the Digital Generation Gap*, Longstreet Press, 1996, s. 50.