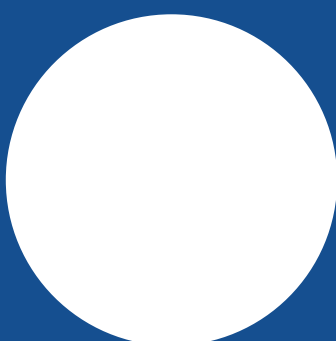
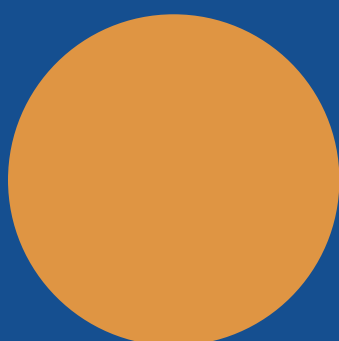
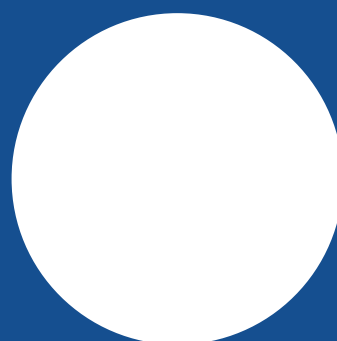
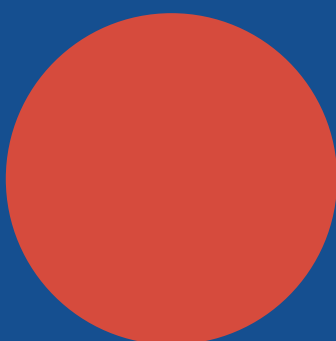
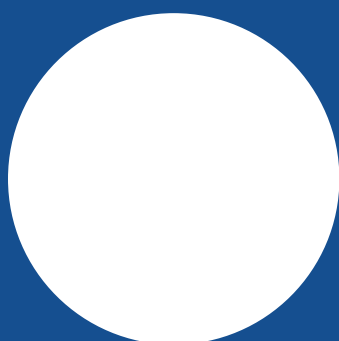
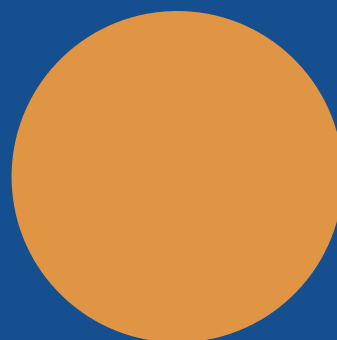
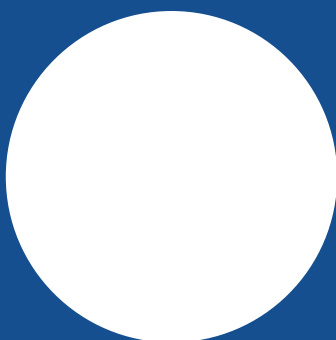
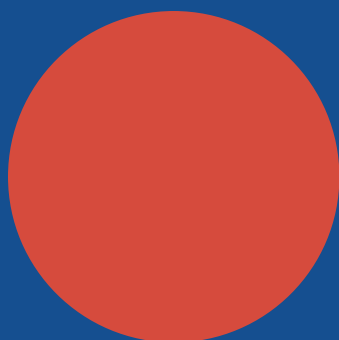


PORADNIK METODYCZNY

JAK UCZYĆ PROGRAMOWANIA
W KLASACH 1-3



PORADNIK METODYCZNY

JAK UCZYĆ PROGRAMOWANIA
W KLASACH 1-3



AUTORKA

Anna Świć

KOREKTA

Maria Krawczyk

OPIEKA MERYTORYCZNA I KOORDYNACJA

Iwona Kolosek

Elżbieta Jankowska

PROJEKT GRAFICZNY I SKŁAD

 RZECZYOBRAZKOWE

WYDAWCA

Fundacja Rozwoju Społeczeństwa Informacyjnego

ISBN 978-83-955081-2-7



Podręcznik dostępny na licencji Creative Commons: Uznanie autorstwa 4.0. Międzynarodowe. Pewne prawa zastrzeżone na rzecz Fundacji Rozwoju Społeczeństwa Informacyjnego i autorki. Zezwala się na dowolne wykorzystanie materiałów, w tym utworów, tworzenie i rozpowszechnianie ich kopii w całości lub we fragmentach, wprowadzanie zmian i rozpowszechnianie utworów zależnych – pod warunkiem zachowania niniejszej informacji licencyjnej i wskazania autorów oraz Fundacji Rozwoju Społeczeństwa Informacyjnego jako właścicieli praw do tekstu. Tekst licencji dostępny na stronie: <https://creativecommons.org/licenses/by/4.0/deed.pl>.

FRSI FUNDACJA
ROZWOJU
SPOŁECZEŃSTWA
INFORMACYJNEGO

MSCDN 
MAZOWIECKIE SAMORZĄDOWE
CENTRUM DOSKONALENIA NAUCZYCIELI

Podręcznik powstały w ramach projektów „Misja: programowanie – podregion ostrołęcki” i „Misja: programowanie – podregion płocki” realizowanych przez Fundację Rozwoju Społeczeństwa Informacyjnego oraz Mazowieckie Samorządowe Centrum Doskonalenia Nauczycieli. Działanie współfinansowane przez Unię Europejską w ramach Europejskiego Funduszu Rozwoju Regionalnego.



 misja: programowanie

Unia Europejska
Europejski Fundusz
Rozwoju Regionalnego



Spis treści

1. PROGRAMOWANIE – CO TO ZNACZY?	→ Jak zacząć uczyć programowania? Jak przezwyciężyć obawy? → Propozycje zabaw rozwijających logiczne, algorytmiczne myślenie → Warunki, pętle, procedury, debugowanie błędów – łatwiejsze, niż myślisz → Pomału i systematycznie – o czym pamiętać, ucząc programowania? → Programowanie jako element zintegrowanych zajęć w edukacji wczesnoszkolnej
2. ROLA UCZNIA I ROLA NAUCZYCIELA W PROCESIE NAUKI PROGRAMOWANIA	→ Uczeń stroną aktywną, kreatywną, poszukującą, nauczyciel w roli moderatora, delikatnie nakierowującego i wspierającego uczniów w procesie programowania → Stereotypy w nauce programowania – jak je przezwyciężyć?
3. OD ZABAW „NA DYWANIE” DO PRACY W APLIKACJACH TABLETOWYCH I NA PLATFORMACH INTERNETOWYCH	→ Doświadczenie, ruch i zabawa pierwszym krokiem w świat programowania → Offlinowe wprowadzenie języka programowania Scratch i platformy code.org → Zbiór aktywności ułatwiających płynne przejście od pracy bez sprzętu do zajęć z wykorzystaniem nowoczesnych technologii
4. ROBOTY EDUKACYJNE – JAK JE ODPOWIEDZIALNIE WPROWADZIĆ DO ZAJĘĆ Z ELEMENTAMI PROGRAMOWANIA	→ Kiedy jest najlepszy czas na wprowadzenie robotów do zajęć dydaktycznych? → Jestem robotem – aktywności wprowadzające uczniów w świat robotyki → Metodyka przed narzędziem – jak efektywnie wykorzystywać roboty edukacyjne na zajęciach?
5. W POSZUKIWANIU INSPIRACJI...	→ Social media – grupy tematyczne miejscem wymiany doświadczeń nauczycieli → Strony internetowe i blogi warte odwiedzenia



1

PROGRAMOWANIE – CO TO ZNACZY?

■ JAK ZACZĄĆ UCZYĆ PROGRAMOWANIA? JAK PRZEZWYCIĘŻYĆ OBAWY?

- Słowo „programowanie” może budzić w nauczycielach lekką obawę. Wynika ona głównie z tego, że często jest to dla nich tematyka, z którą do tej pory nie mieli do czynienia. W dalszym ciągu hasło „programowanie” przywołuje na myśl komputer i wielogodzinne wstukiwanie na klawiaturze niekończącej się długości kodu, w języku zrozumiałym jedynie dla nielicznych. Biorąc pod uwagę dodatkowo brak dostępu do sprzętu komputerowego, który dotyczy wielu placówek, i obowiązków włączania elementów programowania do zajęć edukacyjnych już od klasy pierwszej, te obawy mogą wydawać się całkiem uzasadnione. Często nauczyciele uważają, że brak wykształcenia informatycznego, lub przynajmniej technicznego czy matematycznego, uniemożliwia opanowanie zagadnień związanych z programowaniem, a tym bardziej jego nauczanie w szkole. Nic bardziej mylnego. Programowania uczą nauczyciele różnych specjalności: poloniści, matematycy, bibliotekarze... tu nie występują żadne ograniczenia. Sama nie posiadam wykształcenia informatycznego (jestem pedagogiem przedszkolnym), a z powodzeniem uczę podstaw programowania dzieci i dorosłych.
- Wprowadzanie podstaw programowania nie musi być wcale trudne, śmiało mogą poradzić sobie z tym zadaniem nauczyciele edukacji wczesnoszkolnej. Ważne, żeby byli świadomi, że programowanie nie jest równoważne z tworzeniem linijek kodu w programie komputerowym, a w początkowej fazie nauki programowania komputer nie musi się nawet pojawić. Dodatkowo wiele aktywności okołoprogramistycznych nauczyciele od dawna wykonywali ze swoimi uczniami, nie mając świadomości, że są z nimi powiązane. Robiąc je w pełni świadomie, oczywiście zwiększą efektywność swojej pracy. Ważne jest, żeby spróbowali, a sami przekonają się, że każde kolejne zajęcia, każde kolejne aktywności jest im łatwiej zorganizować. Potem pomysły pojawiają się same, a czas przygotowania do zajęć ulega zdecydowanemu skróceniu.
- **Dlaczego warto uczyć podstaw programowania?** Patrząc na zawody, które wykonywali nasi rodzice czy dziadkowie i na zawody wykonywane aktualnie, można zauważyć, że wiele z tych wykonywanych kiedyś praktycznie przestało istnieć. Pracujemy za to w profesjach, których dawniej w ogóle nie było. Szybki postęp, zmiany cywilizacyjne powodują, że nie jesteśmy w stanie przewidzieć, czym będą zajmować się za kilkanaście lat nasi uczniowie, jacy specjaliści będą poszukiwani na rynku pracy. Są jednak uniwersalne kompetencje, wysoko cenione niezależnie od wykonywanego zawodu. To umiejętność logicznego, algorytmicznego myślenia, poszukiwania rozwiązań stawianych problemów, łączenia faktów i wyciągania wniosków, umiejętność pracy zespołowej, robienie czegoś dla dobra ogółu, ale z poszanowaniem praw jednostki. Wszystkie te umiejętności rozwijają uczniowie, ucząc się programowania. Jeśli dodamy do tego fakt, że świat jest coraz bardziej skomputeryzowany, coraz więcej



czynności wykonywanych jest przez maszyny, to dojdziemy do wniosku, że znajomość podstaw programowania jest uczniom niezbędna. To język, który po prostu trzeba znać.

■ PROPOZYCJE ZABAW ROZWIJAJĄCYCH LOGICZNE, ALGORYTMICZNE MYŚLENIE

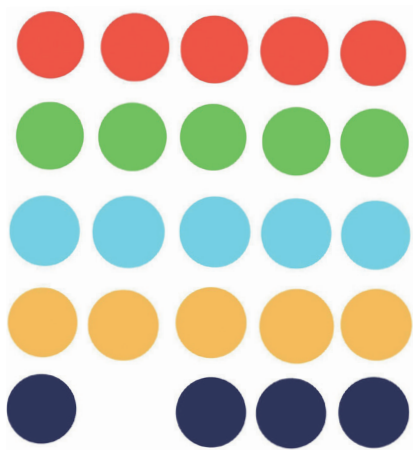
- **Jak najłatwiej zainteresować uczniów programowaniem?** Poprzez aktywności, które będą odbierane przez nich jako zabawa, których wykonywanie będzie radością. Musimy mieć świadomość, że nie zaczniemy nauki od tworzenia skryptów, najpierw trzeba uczniów do tego przygotować. Popracować nad rozwijaniem umiejętności i kompetencji, które będą przydatne w nauce programowania. Pierwsze aktywności mogą być nakierowane na ćwiczenia z zakresu klasyfikowania, porównywania, szukania logicznych prawidłowości. Idealne na wstępie będą wszelkiego rodzaju łamigłówek, zabawy typu sudoku, „bystre oczko”, popularna *Mucha wyszła*, gry planszowe logiczno-strategiczne. Warto również zaproponować aktywności polegające na klasyfikowaniu, grupowaniu elementów, kierując się określonymi kryteriami (wielkość, kolor, liczebność itp.). Przy tego typu zabawach możemy wykorzystać dowolny materiał np. zakrętki, klocki, kolorowe guziki. Okresowo idealne będą też materiały naturalne, takie jak: kasztany, orzechy, liście. Liczbę elementów i kategorie, którymi będziemy kierować się przy klasyfikowaniu, dobieramy do wieku i możliwości rozwojowych uczniów.
- Kiedy różnego rodzaju zabawy związane z grupowaniem elementów mamy za sobą, możemy zaproponować uczniom ułożenie sudoku. Warto zrobić to zespołowo i na większym formacie niż A4. Ruch, związany z pracą na macie, powoduje, że dzieci bardzo chętnie wykonują zaproponowane im aktywności. Na takich zajęciach idealnie sprawdzi się mata edukacyjna lub płaszczyzna podzielona na kratownicę. Sudoku możemy układać z elementów w różnych kolorach np. kolorowych karteczek, klocków, jednorazowych kubeczków. Potrzebujemy tyle elementów z danego koloru, ile będziemy mieć kolorów, przykładowo: wybierając 5 kolorów, należy wziąć po 5 elementów w każdym z wybranych kolorów. W sumie będziemy mieć 25 elementów. Za każdym razem musi nam wyjść kwadrat. Zadanie polega na rozłożeniu wszystkich elementów w taki sposób, żeby kolor nie powtarzał się ani w linii pionowej, ani w linii poziomej. Najłatwiej będzie, jeśli poszukamy na to jakiegoś sposobu, jakiegoś algorytmu.



- Układając kolory po skosie, uda się wykonać zadanie bardzo sprawnie. Pierwszy kolor układamy po przekątnej (na rysunku to 5 czerwonych kół), następnie układamy linię równoległą, na którą składają się 4 niebieskie koła, pozostaje nam 1, które przekładamy w róg po przeciwnej stronie. Kolejna linia to żółte koła. Układamy 3 z nich równoległe do 4 niebieskich, a 2 żółte równoległe do 1 niebieskiego. Kolejny kolor to zielony i linia po skosie składająca się z 2 kół, oraz po przeciwnej stronie przekątnej z 3 kół. Puste miejsca na planszy uzupełniamy kolorem granatowym. Identyczny efekt osiągniemy, jeżeli ułożymy jedną linię składającą się z każdego koloru, a następnie w każdej kolejnej będziemy układać

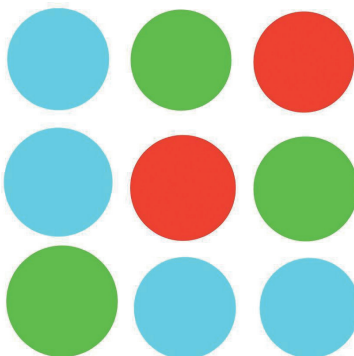
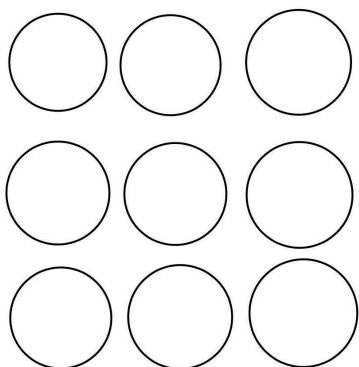


elementy, przesuwając je o 1 pole. Kiedy widzimy, że uczniowie dobrze radzą sobie z wykonywanym zadaniem, możemy podnieść stopień trudności, dodając więcej elementów. Warto, żeby sudoku układane było w niewielkich zespołach. Sposób komunikacji między uczestnikami zabawy ma kluczowy wpływ na jej przebieg. Uczniowie muszą brać pod uwagę siebie nawzajem, działać wspólnie, bo inaczej osiągnięcie sukcesu i prawidłowe wykonanie zadania okaże się niemożliwe.



- Tak ułożoną planszę możemy wykorzystać jako bazę do kolejnej aktywności. Wystarczy zdjąć 1 z kolorowych elementów z planszy (nie ma znaczenia, w którym kolorze i z którego miejsca go zdejmujemy). Po zabraniu z planszy 1 elementu powstaje puste miejsce, które pozwala nam na przesuwanie sąsiadujących z nim elementów o jedną pozycję na planszy. Możemy przesunąć go do góry, do dołu, w prawo lub w lewo. Manipulując w ten sposób elementami, musimy dojść do sytuacji, w której każda linia będzie zawierała elementy wyłącznie w jednym kolorze. Ważne jest, aby elementy w kolorze, który został zdjęty, były ustawiane na końcu, w przeciwnym razie, tworząc linie z innych kolorów, będziemy psuć tę, która jest już ustawiona. Zadanie jest dość trudne, więc warto zacząć od planszy, która składa się z nie większej liczby pól niż 25.
- Podobna aktywność, tylko o znacznie podniesionym stopniu trudności, to ustawienie poprzez przesuwanie elementów konkretnego układu kolorów na planszy. W tym przypadku nie dążymy do pogrupowania elementów kolorystycznie, tylko do odwzorowania konkretnego ich ustawienia. Kiedy uczniowie nie mają problemu ze sprawnym wykonaniem zadania, możemy dodać warunek, że należy wykonać powierzoną im pracę w określonej liczbie ruchów.

- ↓ Z aktywności, które warto zaproponować dzieciom na początku ich przygody z programowaniem, dobrym wyborem może być układanie elementów według kodu kolorystycznego. Potrzebujemy dwóch pokratkowanych płaszczyzn: na jednej ustawiamy wzór, drugą zostawiamy pustą. Na tej pustej uczniowie odwzorują kod, czyli ustawiają elementy analogicznie jak na wzorze. Najpierw kod podaje nauczyciel, następnie można poprosić uczniów, aby dla siebie wzajemnie ustawiali kolorystyczne kody.



■ WARUNKI, PĘTLE, PROCEDURY, DEBUGOWANIE BŁĘDÓW – ŁATWIEJSZE, NIŻ MYŚLISZ

- **Warunki, pętle, procedury, debugowanie błędów** – to nie brzmi dobrze. Mimo tego wprowadzenie tych pojęć programistycznych może okazać się dużo łatwiejsze, niż nam się wydaje. Od czego zacząć? Najłatwiej od poszukania ich w codziennych sytuacjach, bliskich nam, bliskich uczniom. Tłumacząc na zrozumiałym dla każdego przykładzie, będzie dużo łatwiej przełożyć to na język programowania. Zacznijmy od **warunków**. Gdzie możemy ich szukać? Z pierwszych nasuwających się naturalnie przykładów może to być dobór stroju do pogody. Jeżeli świeci słońce i jest na dworze ciepło, to zakładamy koszulkę z krótkim rękawem, jeżeli pada deszcz, to zakładamy przeciwdeszczowy płaszcz lub zabieramy ze sobą parasol. To „jeżeli” to właśnie warunek. Innym przykładem będzie instrukcja przejścia przez przejście dla pieszych. Patrzymy w lewo, jeśli nic nie jedzie, to patrzymy w prawo, jeśli jedzie, to czekamy, aż przejedzie, i wtedy patrzymy w prawo. Następnie, jeżeli nic nie jedzie, to patrzymy w lewo lub, w przypadku kiedy jedzie, czekamy, aż przejedzie, i wtedy patrzymy w lewo. Jeżeli z lewej strony nic nie jedzie to wchodzimy na przejście dla pieszych i przechodzimy na drugą stronę. Podobnie w przypadku przejścia na drugą stronę w miejscu, gdzie znajduje się sygnalizacja świetlna. Jaki warunek musi zostać spełniony, jeżeli chcemy przejść na drugą stronę ulicy? Musi zapalić się zielone światło, w przeciwnym razie musimy stać i czekać. Warunki pojawiają się w wielu zadaniach związanych z programowaniem. Podczas tworzenia skryptu, muszą zostać zawsze uwzględnione. Zanim przejdziemy do aplikacji tabletowych lub pracy z wykorzystaniem platform internetowych, możemy przygotować karty pracy, w których uczniowie przećwiczą prawidłowe rozumienie, czym jest warunek. Wystarczy, że na pokratkowanej kartce zaznaczymy start i stop. Zadaniem uczniów będzie utworzenie kodu – drogi łączącej te dwa punkty ze sobą. Kod mogą utworzyć, rysując strzałki wskazujące kierunek przemieszczania się. Jeśli jedyną podaną przez nas informacją będzie miejsce oznaczające początek drogi oraz miejsce, do którego należy dotrzeć, to okaże się, że jest dużo prawidłowych rozwiązań, że prawidłowy kod nie będzie tylko jeden, lecz znacznie więcej. Dodając warunki, czyli np. miejsca, przez które droga musi przebiegać lub wręcz przeciwnie, miejsca, przez które droga nie może zostać wytyczona, liczba dostępnych rozwiązań się zmniejszy, a poziom trudności zadania wzrośnie, bo uczeń będzie musiał, tworząc kod, uwzględnić podany warunek. Czym więcej warunków, tym trudniejsze zadanie i mniej możliwych, prawidłowych rozwiązań.
- Jedną z bardziej lubianych już przez najmłodszych uczniów aplikacji jest **Kodable**. Kodable to gra nastawiona na naukę programowania. Można w nią grać w aplikacji tabletovej lub w wersji online na komputerach z dostępem do internetu. Bohaterem gry jest mała, puchata kulka Fuzzy, która porusza się po planszy przypominającej labirynt. Musi dotrzeć do mety, po drodze zbierając wszystkie występujące w labiryncie monety. O sposobie poruszania się kulki decyduje kod, który dla niej utworzymy. Dostępne komendy to strzałki kierunkowe: „do góry”, „do dołu”, „w prawo”, „w lewo”. Każda ze strzałek nadaje kierunek poruszania się Fuzzy’ego. Ważne jest to, że kulka, otrzymawszy pojedynczą komendę, nie przesuwa się o jedno pole, tylko podąża we wskazanym kierunku do końca



	A	B	C	D	E	F	G	H	I	J
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										

	A	B	C	D	E	F	G	H	I	J
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										

● KILKA GOTOWYCH PLANSZ DO DRUKU MOŻNA POBRAĆ POD PONIŻSZYM ADRESEM: [HTTPS://DRIVE.GOOGLE.COM/DRIVE/FOLDERS/0B1KLPKM2VBSZX3DFAXFTA1BJEEO](https://drive.google.com/drive/folders/0B1KLPKM2VBSZX3DFAXFTA1BJEEO)

drogi. W pewnym momencie okazuje się, że przemieszczając się w ten sposób, nie ma możliwości zebrania wszystkich monet. Kulka do zebrania wszystkich fantów potrzebuje zmiany kierunku na odcinku prostym, czyli takim, na którym teoretycznie nie może tego zrobić. I tu dochodzimy właśnie do warunków. Jeśli na trasie, w labiryncie pojawia się inny kolor, to jest to miejsce, w którym warunkowo możemy zmienić kierunek przemieszczania się Fuzzy'ego. Wystarczy, że w kodzie, na strzałce określającej kierunek ruchu po zmianie nałożymy kolor, który jest w labiryncie w miejscu zmiany kierunku. W łatwy sposób możemy przygotować karty pracy przypominające powyżej opisaną aktywność. Będzie nam potrzebna do tego kartka z kratownicą 10x10 i pokolorowanymi polami tworzącymi labirynt. Pod spodem uczniowie przy pomocy strzałek będą rozpisywać kod. Na rysunku obok nie są dorysowane monety, które Fuzzy musi łączyć, poruszając się po planszy. Możemy je dorysować w wybranych przez nas miejscach. Zamiast monet możemy narysować inne przedmioty, przez które będzie prowadziła droga. Mogą to być przykładowo owoce i słodczyce albo grafiki kojarzące się z daną porą roku, wykonywanym zawodem. Uczeń będzie musiał utworzyć taki kod, żeby droga prowadziła wyłącznie przez owoce, a omijała słodczyce. W innym przypadku droga będzie musiała prowadzić wyłącznie przez ilustracje kojarzące się ze wskazaną porą roku. Warto tematykę dobrać pod aktualnie realizowane treści.

→ Rysunek obok przedstawia przykładową planszę z narysowanymi literami O, które Fuzzy musi wybierać. Plansza zawiera warunek w postaci koloru fioletowego.

! Pod planszą podane jest gotowe rozwiązanie (skrypt utworzony ze strzałek kierunkowych).

- Wydrukowane plansze ● warto zalaminować, wystarczy wtedy używać pisaka ścieralnego i będą do wielokrotnego wykorzystania. Możemy również rozdać uczniom puste kratownice i kredki, a następnie poprosić ich o przygotowanie plansz dla siebie wzajemnie w parach. Kiedy labirynty będą gotowe, uczniowie wymieniają się planszami i wykonują przygotowane przez koleżankę lub kolegę zadanie. Uczniowie bardzo lubią przygotowywać dla siebie zadania, tworzenie zadań ma również ten plus, że pracujemy nad dodatkowymi umiejętnościami oraz rozwijamy u uczniów kreatywność: nie tylko wykonują zadanie, odtwarzają coś, ale są stroną aktywną, twórczą. To ważny aspekt nauki programowania na tym etapie edukacyjnym.
- Kolejnym pojęciem programistycznym, z którym spotkają się uczniowie, będzie **pętla**, czyli powtórzenie czegoś. Liczba powtórzeń może być różna. Czym jest pętla, łatwo możemy zaobserwować, analizując cykliczność dni tygodnia, dnia i nocy, miesięcy czy godzin. W tych przypadkach mamy „pętlę zawsze”, stale powtarzającą się sekwencję elementów. Dzień występuje po nocy, a po dniu noc i znowu dzień, i znowu noc, i znowu... Moglibyśmy więc noc i dzień wziąć w pętlę i ustawić liczbę powtórzeń na „zawsze”.
- Aktywności ćwiczące prawidłowe rozumienie pojęcia „pętla” można łatwo zorganizować. Wystarczy zaproponować uczniom „zakodowaną gimnastykę”. Wspólnie wymyślcie symbole graficzne dla kilku ćwiczeń



gimnastycznych np. dla skłonu, podskoku, wymachu nogi itp., następnie ułóżcie je, biorąc dwa, trzy symbole w nawias i dodając przed nim liczbę. Ta liczba wskaże nam, ile razy mamy powtórzyć daną sekwencję następujących po sobie ćwiczeń gimnastycznych. Na podobnej zasadzie możemy przygotować kartę pracy. Rysujemy kilka prostych graficznie rysunków, bierzemy je w nawias, dodajemy przed nawiasem liczbę. Uczeń będzie musiał odkodować zapis i narysować odpowiednią sekwencję grafik we właściwej liczbie.

- Innym pojęciem programistycznym, które możemy w łatwy sposób wprowadzić, są **procedury**. Programy komputerowe mogą składać się z wielu wierszy kodu i zapanowanie nad taką konstrukcją nie jest łatwe. Niezbędne jest wtedy wyodrębnienie fragmentu kodu i nadanie mu jakiejś nazwy. Ważne, żeby nazwa sugerowała znaczenie tego konkretnego fragmentu kodu, ułatwi to wstawienie procedur we właściwych miejscach. Wyobraźmy sobie, że chcemy utworzyć skrypt, który spowoduje przemieszczenie się człowieka po kwadracie o bokach długości równej 3 kroków osoby wykonującej zadanie. Jak taki skrypt mógłby wyglądać? Wykonaj 3 kroki w przód, obróć się o 90 stopni w prawo, wykonaj 3 kroki w przód, obróć się o 90 stopni w prawo, wykonaj 3 kroki w przód, obróć się o 90 stopni w prawo, wykonaj 3 kroki w przód, obróć się o 90 stopni w prawo. Teraz spróbujmy wprowadzić procedurę: wykonaj 3 kroki w przód, obróć się o 90 stopni. Nazwijmy ją bok kwadratu. Jak teraz będzie wyglądał kod? Wykonaj bok kwadratu, bok kwadratu, bok kwadratu, bok kwadratu. Tu dodatkowo można zauważyć, że zasadne będzie wprowadzenie pętli i powtórzenie procedury bok kwadratu 4 razy. Taki zabieg znacznie skróci kod.

	A	B	C	D	E	F	G	H	I	J
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										

- Umiejętność znalezienia błędu w kodzie, a następnie skorygowanie go, nie należy do łatwych zadań. Posługując się językiem programowania, mamy do czynienia z **debugowaniem błędów**. Programiści muszą sprawdzić program, odszukać ewentualne pomyłki i naprawić je. Niejednokrotnie znalezienie błędu okazuje się trudniejsze niż utworzenie skryptu od początku. Kody bywają długie i wychwycenie nieprawidłowej komendy może okazać się nie lada wyzwaniem. Pracując z uczniami, wprowadzając im elementy programowania do zajęć dydaktycznych, możemy również zaproponować aktywności, w których pojawi się coś na wzór pierwszych prób debugowania błędów. Przygotujmy jakieś zadanie, np. przejście z miejsca na miejsce, i rozpiszmy do niego kod przy pomocy strzałek kierunkowych. Celowo jedną lub dwie z nich ustawmy w złym kierunku. Teraz pora na aktywność uczniów. Najpierw pójdą zgodnie z kodem, co pozwoli im zaobserwować, że przemieszcili się w złym kierunku, nie dotarli do miejsca, w którym powinni się znaleźć. Następnie muszą ustalić, co spowodowało, że nie udało im się dotrzeć na miejsce, i poprawić kod, czyli ustawienie strzałek kierunkowych. Na końcu należy ponownie sprawdzić prawidłowość skryptu, pokonując trasę zgodnie z poprawionym kodem. Takich przykładów możemy wykonać więcej.

- ! Następnie możemy zaproponować uczniom podobną aktywność w formie karty pracy, przygotowanej na bazie labiryntów, które omówione były wyżej.
- Rozwiązanie umieszczone pod labiryntem nie jest prawidłowe, zawiera błąd, który uczeń musi znaleźć, a następnie wskazać sposób jego naprawienia. I podobnie jak w zabawach z warunkami tu również warto oddać



tworzenie zadań uczniom i poprosić, żeby samodzielnie przygotowali dla koleżanek i kolegów karty z debugowaniem błędów.

! *Jeżeli połączymy w jednej karcie i warunki, i debugowanie błędów, poziom trudności zadania zostanie znacznie podniesiony. Powyższe przykłady możemy oczywiście przenieść na większy format i zorganizować podobne aktywności na macie edukacyjnej.*

■ POMAŁU I SYSTEMATYCZNIE – O CZYM PAMIĘTAĆ, UCZĄC PROGRAMOWANIA?

- Uczenie programowania to proces – długotrwały i systematyczny. Musimy mieć świadomość, że pojedyncze zajęcia wzbogacone o elementy programowania nie wniosą zbyt wiele, dopiero regularne aktywności efektywnie i bezpiecznie wprowadzą uczniów w świat programowania. Nie musimy się więc spieszyć, tylko w tempie swoich uczniów i swoim realizować kolejne aktywności. Nic się nie stanie, jeśli nie uda nam się przerobić czegoś, co było zawarte w danym scenariuszu. Ważny jest efekt, który osiągniemy po realizacji całego procesu. Pośpiech w tym przypadku absolutnie nie jest wskazany, podążajmy za dziećmi, za ich tempem rozwoju. Istotne jest to, żeby wszystkie zajęcia z elementami programowania tworzyły spójną całość, wynikały jedno z drugich, poszerzały i rozwijały nabyte wcześniej umiejętności. Warto wybrać najbardziej odpowiadające potrzebom naszych uczniów i naszym narzędzia i z ich wykorzystaniem organizować aktywności. Wprowadzanie za każdym razem nowych pomocy nie zwiększy efektywności zajęć, wręcz przeciwnie, może tylko zaburzyć proces nauki podstaw programowania. Pamiętajmy, że każde kolejne narzędzie to trochę inny sposób programowania i za każdym razem wymaga wprowadzenia, więc częste zmienianie narzędzi programistycznych w początkowej fazie nauki może na nią negatywnie wpłynąć.

■ PROGRAMOWANIE JAKO ELEMENT ZINTEGROWANYCH ZAJĘĆ W EDUKACJI WCZESNOSZKOLNEJ

- Nauka programowania nie musi, a nawet nie powinna być traktowana jako oddzielna, wyodrębniona jednostka lekcyjna. Elementy programowania z powodzeniem można włączać do zajęć dydaktycznych z różnych dziedzin edukacji – łączyć programowanie i zajęcia rozwijające kompetencje językowe lub programowanie i zajęcia kształtujące kompetencje matematyczne, artystyczne, społeczne, a nawet wychowanie fizyczne. Wplatając elementy kodowania do różnych zajęć dydaktycznych, mamy większe szanse na regularne wprowadzenie elementów programowania naszym uczniom. Nawet jeśli początkowo wydaje nam się to trudnym zadaniem, to z każdymi kolejnymi zajęciami będzie to łatwiejsze do zrealizowania. Spróbujmy traktować elementy programowania jako naturalną rzecz na zajęciach, coś, co jest częścią nich, pewnym rytuałem. W dalszej części przedstawię kilka propozycji włączających elementy podstaw programowania do zajęć z różnych dziedzin edukacji.





ZAKODOWANA GIMNASTYKA

- W zaproponowanej aktywności kierujemy się kodem składającym się z cyfr i kolorów. Cyfry to liczba powtórzeń, kolor to rodzaj ćwiczenia. Kubek to pionek, który będzie przemieszczał się po planszy na podstawie wydawanych przez nauczyciela komend słownych: „do góry”, „do dołu”, „w lewo” i „w prawo”.

	A	B	C	D	E	F	G	H	I	J
1		1+2								
2										
3			3-1		15+7					
4		2+8								
5		6-4				21				
6										
7		17		4+4			23			
8					7+9					
9			5-4			9-3		5+5		
10										

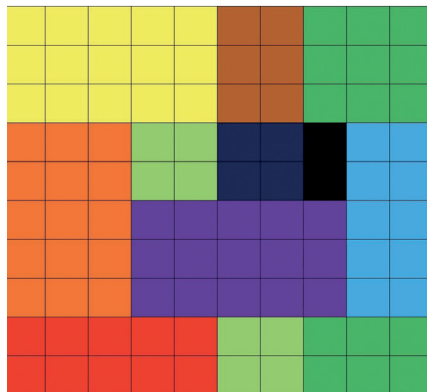
	A	B	C	D	E	F	G	H	I	J
1		15								
2										
3				25		5				
4		32								
5				9			6			
6										
7		24		7	3					
8					21					
9		6		8			9			
10										

ZAKODOWANY UKŁAD CHOREOGRAFICZNY

- Układ ruchów zakodowany w postaci sekwencji symboli przedstawiających różne ćwiczenia gimnastyczne może być świetną propozycją na zajęcia wychowania fizycznego. W taki sposób możemy zakodować uczniom również prosty taniec. Wystarczy narysować symbole graficzne odpowiadające różnym aktywnościom, a następnie ułożyć je w całość, zaczynając od startu, a kończąc na stopie. Dodając do tego muzykę, stworzymy wartościowe zajęcia dla młodszych uczniów. Taką aktywność możemy wykorzystać również na zajęciach integracyjnych z rodzicami czy różnego rodzaju uroczystościach.
- Edukacja matematyczna jest wprost idealna do tego, by łączyć ją z programowaniem. Porządkowanie, tworzenie zbiorów, segregowanie to przecież elementy wspólne, które obecne są i w edukacji matematycznej, i w kodowaniu. Wprowadzanie układu współrzędnych również idealnie połączy się z programowaniem. Dyktanda graficzne, w których uczniowie odkodowują wzór na bazie podawanych przez nauczyciela kolorów na kratownicy i ich miejsc, to bardzo lubiana przez uczniów aktywność. Z czasem będą nie tylko szybko odczytywać współrzędne, ale i nauczą się samodzielnie tworzyć kolorowe wzory i zakodowywać je dla kolegów i koleżanek. W zależności od stopnia skomplikowania wzoru będzie to bardzo dobre ćwiczenie zarówno dla uczniów w klasie I, jak i w klasie III.
 - Omawiane wyżej labirynty będą idealne do połączenia z edukacją matematyczną. Wystarczy, że na wybranych polach umieścimy działania matematyczne. Dodatkowo określając sposób przejścia labiryntu, również warto wpleść w to zadanie matematyczne. Możemy pokonywać labirynt w taki sposób, aby suma mijanych pól była np. większa od 100, parzysta, nieparzysta itp. Stopień trudności działań dopasujemy do wieku uczniów, ich możliwości rozwojowych. Pamiętajmy też o aktualnie realizowanym materiale i tak przygotowujemy zadanie, aby było z nim spójne.

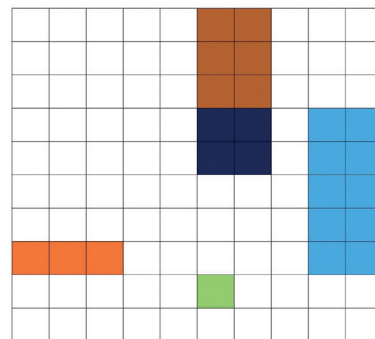
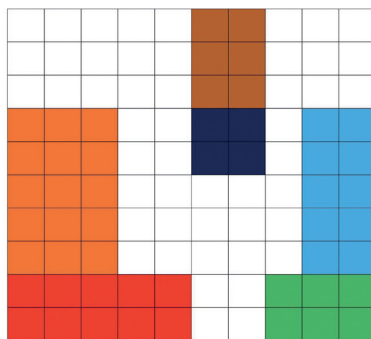
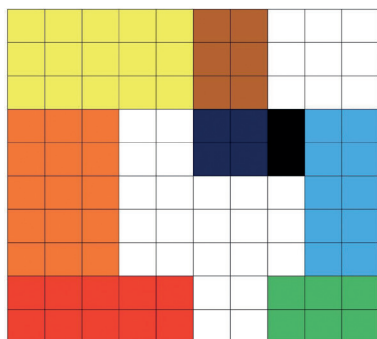


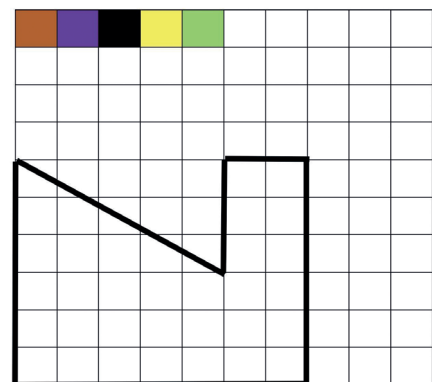
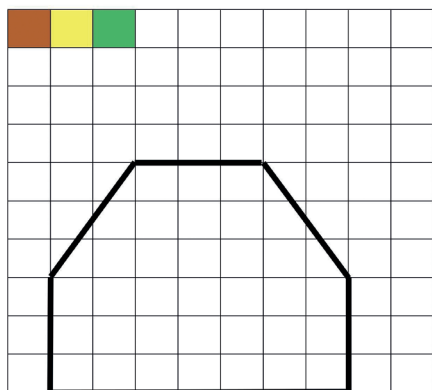
LOGICZNE ŁAMIGŁÓWKI



- Kolejną propozycją, która idealnie sprawdzi się podczas zajęć edukacji matematycznej, będą logiczne łamigłówki. Znajomość figur geometrycznych można ćwiczyć z uczniami, rozwijając jednocześnie logiczne myślenie, które z kolei nierozdzielnie łączy się z programowaniem. Co będzie nam potrzebne do zorganizowania takich aktywności? Kolorowa plansza do pocięcia na mniejsze figury, czysta kartka lub kratownica i nożyczki oraz klej. Zaczniemy od przygotowania kolorowych figur. Rozcinamy planszę z figurami w taki sposób, żeby każda z figur była oddzielnie. Dodatkowo fioletowy prostokąt przecinamy w miejscu przekątnej tak, żeby powstały dwa identyczne trójkąty. Podobnie postępujemy z brązowym prostokątem i jednym jasnozielonym kwadratem. W ten sposób otrzymujemy następujące figury geometryczne:

1. żółty prostokąt
 2. pomarańczowy prostokąt
 3. czerwony prostokąt
 4. ciemnozielony prostokąt
 5. jasnoniebieski prostokąt
 6. czarny prostokąt
 7. jasnozielony kwadrat
 8. granatowy kwadrat
 9. ciemnozielony kwadrat
 10. 2 jasnozielone trójkąty
 11. 2 fioletowe trójkąty
 12. 2 brązowe trójkąty
- ↓ Zadaniem uczniów będzie złożenie tych wszystkich figur w jedną figurę – duży kwadrat. Uczniowie otrzymują odpowiedzi w postaci kartki z narysowaną kratownicą, na której część figur jest już zaznaczona. Należy uzupełnić te, których brakuje. Im mniej zaznaczonych figur na kartce, tym zadanie ma wyższy poziom trudności. Uczniowie mogą układać figury na pustej kratownicy (wtedy warto ją zalaminować, by była do wielokrotnego wykorzystania) lub przyklejać na kartce.





→ Na bazie tych samych kolorowych figur geometrycznych możemy zaproponować uczniom kolejną aktywność. Tym razem zadanie polega na ułożeniu figury, która będzie miała identyczny ze wskazanym na kartach zarys. Kolorowe kwadraty to wskazówka, których figur należy użyć w danym przykładzie. Trudność polega na tym, że nie wiemy, ile figur z danego koloru będzie nam potrzebnych i w jakim kierunku należy je ułożyć. Bawiąc się w ten sposób, nie tylko rozwijamy spostrzegawczość i logiczne myślenie u uczniów, ale również powtarzamy właściwości figur geometrycznych.

- Również zajęcia z wykorzystaniem Scratcha będą idealnym sposobem łączenia treści programistycznych z tymi, które realizujemy w ramach innych dziedzin edukacji. Scratch pozwala na tworzenie animacji, więc świetnie sprawdzi się na zajęciach językowych. Uczniowie mogą za pomocą takiej animacji przedstawić treść wiersza, lektury, bohatera. Dopasowując wygląd duszka czy sceny, rozwijają swój zmysł artystyczny. Scratch i edukacja matematyczna również będą tworzyć bardzo udany duet. Programowanie ruchów duszka to idealny sposób na utrwalanie wiadomości dotyczących układu współrzędnych. Program daje również możliwość programowania prostych gier, w tym gier matematycznych.
- W programowaniu bardzo ważny jest precyzyjny przekaz informacji. Nadawca i odbiorca muszą porozumiewać się w tym samym języku, tym samym kodem. Ćwiczenie w tym zakresie idealnie połączy się z edukacją rozwijającą możliwości językowe. Wystarczy, że na pokratkowanej planszy jedna grupa narysuje obrazek, a następnie rozpisze go za pomocą instrukcji słownej, drugi zespół, słuchając instrukcji, będzie według niej próbował ten obrazek odtworzyć. Trudność zadania możemy podnieść, umawiając się z uczniami na rozpisanie instrukcji przy pomocy rymowanki. Analogiczne zadanie możemy zorganizować, prosząc uczniów o utworzenie budowli z kilku klocków i, podobnie jak w propozycji z obrazkiem, stworzenie instrukcji, na bazie której ktoś ustawi z klocków taką samą konstrukcję.

2

ROLA UCZNIA I ROLA NAUCZYCIELA W PROCESIE NAUKI PROGRAMOWANIA

■ UCZEŃ STRONĄ AKTYWNA, KREATYWNĄ, POSZUKUJĄCĄ, NAUCZYCIEL W ROLI MODERATORA, DELIKATNIE NAKIEROWUJĄCEGO I WSPIERAJĄCEGO UCZNIÓW W PROCESIE PROGRAMOWANIA

- Ucząc programowania, warto zastanowić się nad rolą ucznia i rolą nauczyciela w tym procesie. Łatwiej będzie nam ustalić, jakie role powinniśmy przyjąć, jeśli wcześniej zastanowimy się, czego chcemy dla naszych uczniów, co mają wynieść dla siebie z prowadzonych przez nas zajęć. Nauka elementów programowania to między innymi nauka szukania rozwiązań stawianych nam problemów, eksperymentowania, poszukiwania, próbowania, a następnie wybrania tego najbardziej optymalnego i wprowadzenia go w życie. Uczeń będzie doświadczał i poszukiwał, jeśli mu na to pozwolimy, jeśli będzie miał możliwość zadawania pytań, rozważania różnych możliwości. Jeśli ograniczymy mu wybór tylko do odpowiedzi



„tak” lub „nie”, to nie oczekujemy po nim kreatywności. Tym samym rola nauczyciela powinna być wspierająca, jedynie lekko nakierowująca. Dobry pedagog będzie blisko, gotowy do pomocy, ale lekko w cieniu, nie będzie podawał gotowych rozwiązań, tylko zachęcał do ich poszukiwania. Nie będzie wykonywał zadań za dzieci, raczej przedstawi im problem i da czas, żeby spróbowały go rozwiązać.

- Musimy mieć świadomość, że zajęcia z elementami programowania to nie są zajęcia w ławkach, kiedy to nauczyciel opowiada, a uczniowie słuchają, ewentualnie odpowiadają na zadawane im pytania. Tu role są całkiem inne. Nie oczekujemy też idealnej ciszy, kiedy jest współpraca, wymiana zdań między uczniami; kiedy coś się tworzy, to rozmowy muszą temu towarzyszyć. Z początku możemy mieć wrażenie, że powstaje harmider, jakiś lekki nieład, ale kiedy uczniowie nauczą się pracować w zespołach, to takie aktywności będą przebiegać bardzo sprawnie.

■ STEREOTYPY W NAUCE PROGRAMOWANIA – JAK JE PRZEZWYCIĘŻAĆ?

- Myśląc o programowaniu, zdarza nam się robić to stereotypowo, powtarzając mity, które ze stanem faktycznym nie mają wiele wspólnego. Programowanie przez wielu widziane jest jako aktywność dla samotników. Lepiej poradzą sobie z nim chłopcy niż dziewczyny. To nieprawda. W nauce programowania współpraca jest kluczowa. Umiejętność pracy w zespole, szukania kompromisowych, odpowiadających wszystkim rozwiązań, jest wręcz niezbędna. Mało który programista pracuje w pojedynkę. Za najlepszymi projektami zazwyczaj stoi grupa współpracujących ze sobą ludzi. Podobnie jest z przekonaniem, że dziewczynki nie odnajdą się w świecie programowania. Nic bardziej mylnego – poradzą sobie i to bez problemu. Ważne, żeby nie powtarzać takiego stereotypowego myślenia, żeby nie wpędzać w nie dzieci. Jeśli pokażemy im, że można, że warto próbować, że dadzą radę, to z dużym prawdopodobieństwem tak właśnie będzie.

3

**OD ZABAW „NA DYWANIE”
DO PRACY W APLIKACJACH
TABLETOWYCH
I NA PLATFORMACH
INTERNETOWYCH**

■ DOŚWIADCZENIE, RUCH I ZABAWA PIERWSZYM KROKIEM W ŚWIAT PROGRAMOWANIA

- Najłatwiej jest się czegoś nauczyć, jeśli dobrze to zrozumiemy, jeśli doświadczymy tego na sobie. Samo słuchanie o czymś przynosi zdecydowanie gorsze efekty. Podobnie jest z nauką programowania. Wielu pojęć programistycznych, sposobów tworzenia skryptów możemy nauczyć się poprzez działanie, aktywnie, w ruchu. Przenosząc nabyte w ten sposób umiejętności na pracę z wykorzystaniem sprzętu komputerowego, sprawiamy, że uczniom będzie łatwiej opanować nowe zagadnienia, tym samym i nam będzie łatwiej pracować. Unikniemy „lasu



rąk” podniesionych do góry w geście prośby o pomoc, więc będziemy mogli skupić się na pomocy tym uczniom, którzy faktycznie potrzebują indywidualnego wsparcia.

■ OFFLINOWE WPROWADZENIE JĘZYKA PROGRAMOWANIA SCRATCH I PLATFORMY CODE.ORG

- **Scratch** to bardzo popularny wizualny język programowania. Został zaprojektowany przez Mitchela Resnicka. Scratch umożliwia tworzenie interaktywnych historyjek, gier czy muzyki w bardzo łatwy sposób. Komendy w Scratchu mają kształt puzzli, które przeciągamy i układamy w określonym porządku. W ten sposób tworzony jest kod, który przypisany jest do konkretnego obiektu. Wygląd duszków możemy pobierać z galerii, edytować, tworzyć lub importować z zewnątrz. Dodatkowym atutem Scratcha jest również społecznościowy serwis, pozwalający na umieszczanie stworzonych w Scratchu programów oraz pobieranie prac stworzonych przez innych użytkowników.
- **code.org** to platforma internetowa, na której mamy do dyspozycji wiele aktywności o charakterze programistycznym. Proponowane na niej gry, zadania mają różny stopień trudności, więc nie jest problemem znalezienie takich, które będą odpowiadać naszym uczniom. Filmy instruktażowe dostępne na platformie znacznie ułatwiają pracę nauczycielowi.
- Zanim zaproponujemy uczniom działania na platformie code.org lub w Scratchu, warto zasugerować uczniom podobne aktywności, ale bez komputera. Pobawmy się nimi, proponując uczniom zamienienie się w któregoś z bohaterów gry lub duszka ze Scratcha. Programując offlinowo wybranego bohatera, używajmy zbliżonych komend do tych, które pojawiają się w programie.

4

ROBOTY EDUKACYJNE – JAK JE ODPOWIEDZIALNIE WPROWADZIĆ DO ZAJĘĆ Z ELEMENTAMI PROGRAMOWANIA

■ KIEDY JEST NAJLEPSZY CZAS NA WPROWADZENIE ROBOTÓW DO ZAJĘĆ DYDAKTYCZNYCH?

- Kiedy aktywności typowo offlinowe (bez użycia sprzętu komputerowego) mamy już za sobą, to warto wzbogacić zajęcia z elementami programowania o robotykę. Wykorzystując takie narzędzia programistyczne, uczniowie w łatwy sposób mogą przekonać się, że ich działanie ma realny efekt, że programowanie przekłada się na konkretne zachowanie jakiegoś obiektu, że to oni mają siłę sprawczą, decydują o sposobie poruszania się, zachowania danego robota. Widzą też, że to nie robot decyduje, tylko osoba, która go programuje.



■ JESTEM ROBOTEM – AKTYWNOŚCI WPROWADZAJĄCE UCZNIÓW W ŚWIAT ROBOTYKI

- **Od czego zacząć, wprowadzając zajęcia z elementami robotyki?** Podobnie jak przy innych aktywnościach nakierowanych na rozwijanie kompetencji związanych z programowaniem – od doświadczania, sprawdzania na sobie. Warto zaproponować uczniom bardzo lubianą przez nich zabawę, czyli zamienienie ich w robota. Podczas zabawy należy wybrać jedną osobę, która będzie robotem, a pozostali uczniowie programistami, którzy zaprogramują ją na takie zachowania, jakich oczekują. Zwróćcie uwagę, że robot, aby wykonywał różnego rodzaju czynności, musi otrzymywać precyzyjne komunikaty, inaczej będzie postępować losowo. Typ komend mogą uczniowie dobrać w taki sposób, żeby były tożsame z tymi, z których będą korzystać, programując prawdziwego robota.

■ METODYKA PRZED NARZĘDZIEM – JAK EFEKTYWNIIE WYKORZYSTYWAĆ ROBOTY EDUKACYJNE NA ZAJĘCIACH?

- Roboty edukacyjne są narzędziem bardzo atrakcyjnym zarówno dla dziecka, jak i dla nauczyciela. Co nie zmienia faktu, że pozostają wyłącznie narzędziem i musimy o tym pamiętać. Nie powinny stanowić celu same w sobie, a jedynie być sposobem na osiągnięcie założonych przez nas celów edukacyjnych, dodatkiem do zajęć, a nie ich główną atrakcją. Im bardziej będą neutralne, tym lepiej. Ich główna wartość powinna polegać na tym, co uczeń może z nimi zrobić, jak je wykorzystać. Pracując z wykorzystaniem robotów, należy starannie przemyśleć sposób organizacji zajęć. Zazwyczaj do dyspozycji nie mamy wielu robotów, tylko znacznie ograniczoną ich liczbę. Idealną formą zajęć w tym przypadku będzie praca w zespołach. Zadbajmy o zaproponowanie takich aktywności, w których wszyscy uczniowie będą mieli zadanie do wykonania. Praca w małych grupach skróci czas oczekiwania na urządzenie, pozytywnie wpłynie również na rozwój kompetencji społecznych. Podczas pierwszego kontaktu uczniów z nowym narzędziem warto ustalić zasady korzystania z niego, zapoznać ich ze sposobem obsługi. Pamiętajmy również, że przy pierwszych kontaktach uczniów z różnego rodzaju robotami zazwyczaj jest ogromna radość, nie oczekujmy, że od razu zrealizujemy zaplanowany scenariusz. Dajmy dzieciom czas na nacieszenie się robotem, na odrobinę aktywności swobodnych, to spowoduje, że na kolejnych zajęciach będzie można już zaproponować uczniom konkretną aktywność z wykorzystaniem tego narzędzia.

5

W POSZUKIWANIU INSPIRACJI...

■ SOCIAL MEDIA – GRUPY TEMATYCZNE MIEJSCEM WYMIANY DOŚWIADCZEŃ NAUCZYCIELI

- Niezależnie od tego, jak dobrze jesteśmy przygotowani do przeprowadzenia zajęć z uczniami, bywają momenty, kiedy czujemy, że potrzebujemy wsparcia, jakiejś inspiracji, rady kogoś, kto jest w podobnej sytuacji



jak my. Odpowiedzią mogą być grupy tematyczne na Facebooku. Jak to działa? Grupa ludzi, których coś łączy (tematyka zainteresowań, region, poglądy), zrzesza się w obrębie danej grupy na Facebooku. Takie grupy mogą różnić się dostępnością: mogą być otwarte (każdy widzi posty znajdujące się w grupie), zamknięte (posty mogą czytać i publikować członkowie grupy, osoby, które nie należą do grupy nie widzą postów, widzą tylko grupę i jej opis), tajne (posty mogą publikować i czytać wyłącznie członkowie grupy, dodatkowo grupa nie jest widoczna dla innych użytkowników). Warto poszukać grup na Facebooku o tematyce, która nas interesuje. Istnieją grupy, w których głównym tematem jest wprowadzanie elementów programowania. „Uczymy dzieci programować” to grupa, którą prowadzę. Wspólnie z innymi nauczycielami z całej Polski wymieniamy się na grupie różnego rodzaju inspiracjami i spostrzeżeniami na temat programowania. Jeśli nie znaleźliśmy grupy, która w pełni nam odpowiada, możemy rozważyć założenie takiej grupy samodzielnie i zaproszenie do niej osób, które również interesują się podobną tematyką co my.

■ STRONY INTERNETOWE I BLOGI WARTO ODWIEDZENIA

● POLECANE STRONY
kodowanienadywanie.blogspot.com
WIKI MISTRZOWIE KODOWANIA
<http://wiki.mistrzowiekodowania.pl/>

- Planując aktywności związane z programowaniem, warto szukać inspiracji w sieci. Wiele jest stron prowadzonych przez nauczycieli, fundacje, stowarzyszenia, na których możemy znaleźć bezpłatne materiały dla nauczycieli ● Często zamieszczone są tam gotowe scenariusze, karty pracy do druku, filmy instruktażowe. Ja zapraszam na stronę: kodowanienadywanie.blogspot.com, gdzie dzielę się swoją wiedzą i pomysłami na zajęcia.
- Kopalnią bezpłatnych materiałów związanych z programowaniem jest również **Wiki Mistrzowie Kodowania**. Można tam znaleźć nie tylko scenariusze zajęć (pogrupowane pod kątem języków programowania), ale również tutoriale i nagrania webinarów prowadzonych przez trenerów Mistrzów Kodowania.



O autorce

ANNA ŚWIĆ

Pedagog przedszkolny, logopeda, właścicielka firmy szkoleniowej Kodowanie na dywanie. Entuzjastka nowoczesnej edukacji, w praktyce wdraża edukację opartą o model edukacyjny STEAM, autorka wielu scenariuszy zajęć dydaktycznych dla dzieci i warsztatów dla nauczycieli. Członkini grup innowacyjnych nauczycieli: Superbelfrzy RP, Superbelfrzy Mini. Autorka i prowadząca bloga kodowanienadywanie.blogspot.com, który doczekał się swojej odsłony w druku. Trenerka w programie Mistrzowie Kodowania. Ambasadorka marki EduSense. Koordynatorka ogólnopolskiego programu Uczymy dzieci programować.

