

poradnik metodyczny

dla kodyjących nauczycielek

i nauczycieli

Karolina Żelazowska



SZTAFETA
POMYSŁÓW
I METOD



Zaprogramuj
Przyszłość



Karolina Żelazowska

SZTAFETA
POMYSŁÓW
I METOD

Warszawa 2018

Publikacja jest współfinansowana przez Unię Europejską ze środków Europejskiego Funduszu Rozwoju Regionalnego w ramach Programu Operacyjnego Polska Cyfrowa na lata 2014–2020 w obrębie projektów „Zaprogramuj przyszłość”.

Współpraca merytoryczna: Joanna Apanasewicz

Redakcja: Iwona Brzózka-Złotnicka

Korekta językowa: Katarzyna Gańko

Skład i opracowanie graficzne: Filip Makowiecki

Scenariusze są udostępniane na licencji Creative Commons Uznanie autorstwa–Na tych samych warunkach 4.0 Międzynarodowe (CC BY-SA 4.0).

W publikacji wykorzystano zdjęcia i ilustracje:

Okladka: Tirachard Kumtanom / Pexels.com (<https://www.pexels.com/photo/background-clips-eraser-notebook-584305>); **s. 17:** *Programmer Icon Flat* – macrovector / Freepik (https://www.freepik.com/free-vector/programmer-icon-flat_1537329.htm); **s. 21:** *Infographic Flow Chart* – Katemangostar / Freepik (https://www.freepik.com/free-vector/infographic-flow-chart-slide-template_1430736.htm); **s. 23:** *Sandwich logo* – Eightonesix / Freepik (https://www.freepik.com/free-vector/sandwich-logo_1090871.htm); **s. 25:** *Hand drawn robot character collection* – Freepik (https://www.freepik.com/free-vector/hand-drawn-robot-character-collection_1511346.htm); **s. 40:** *Dog Race Run* – LaBruixa / Pixabay.com (<https://pixabay.com/en/dog-race-fun-animal-pet-play-run-644111>); **s. 42:** *Painter with Paint Bucket* – Freepik / Flaticon.com (https://www.flaticon.com/free-icon/painter-with-paint-bucket_72014); **s. 50:** *Black cat* – Freepik / Flaticon.com (https://www.flaticon.com/free-icon/black-cat_1123934), *Tree* – Freepik / Flaticon.com (https://www.flaticon.com/free-icon/tree_49009), *Tree* – Freepik / Flaticon.com (https://www.flaticon.com/free-icon/tree_616541)

SPIS TREŚCI

Do biegu		5
Gotowi		9
Dokąd?		11
Jak?		13
Start		15
A jak algorytm		16
B jak binarny kod		24
C jak ciąg trasy		48
D jak debugowanie		57
E jak ekstra dodatek		60
F jak finał		67



DO BIEGU



Potrzeba umiejętnego i bezpiecznego wykorzystywania technologii sprawia, że zagadnienia związane m.in. z programowaniem wykroczyły poza wąskie kręgi specjalistów. Pojawianie się tych treści w przestrzeni szkolnej oraz wykorzystywanie elementów wywodzących się z podstaw programowania w edukacji odbierane jest jako pozytywny trend, który warto wzmacniać.

Nie bez powodu zatytułowaliśmy poradnik **„Sztafetą pomysłów i metod”**, a całej strukturze rozdziałów nadałyśmy układ przypominający bieg. Z jednej strony takie ujęcie oddaje istotę tworzenia tego materiału, jaką jest chęć dzielenia się swoim wieloletnim doświadczeniem – swoistego przekazywania dalej wiedzy, inspirowania kolejnych zawodników, jak w edukacyjnej sztafecie. Z drugiej zaś sygnalizuje, że o programowaniu w edukacji wczesnoszkolnej warto mówić, korzystając z różnych narracji, wplatając treści programistyczne w hasła czasem pozornie dalekie od komputerów i kodów.

Aby wystartowała sztafeta, potrzebna jest grupa ludzi, zespół połączony wspólnym celem. Zawodnikami mogą być wszyscy edukatorzy otwarci na ideę *playful learning*, czyli rozwój poprzez aktywną zabawę, niosącą elementy rozwijania umiejętności i pogłębiania oraz poszerzania wiedzy. Tak aby kierując się zasadą *pozwól mi działać, a zrozumiem wkraczać razem z dziećmi w świat kodowania*.

Celem niniejszego *Poradnika* jest **wspieranie nauczycieli chcących współtworzyć przestrzeń umożliwiającą uczniom działania wykorzystujące specyficzne rozumowanie wywodzące się z programowania, także do rozwiązywania problemów z innych dziedzin.**

O sukcesie w sztafecie decyduje nie tylko wysiłek poszczególnych zawodników, ale i precyzja oraz zgranie w chwili przekazywania sobie pałeczki. W kontekście tworzenia tego poradnika są to selekcja licznych propozycji aktywności i zestawienie ich w optymalnej do startu ilości, jak również opracowanie ich w sposób przystępny dla początkujących nauczycieli.

Tak jak w sporcie struktura *Poradnika* jest podzielona na etapy. W części wstępnej, jak w czasie rozgrzewki, zawarte zostały treści przygotowujące, wprowadzające w zagadnienia podstaw programowania. Dzięki przeglądowi idei zawodnicy mogą przygotować się do ćwiczeń zasadniczych. Zawarte w drugim rozdziale treści skupiają się wokół kwestii podstaw programowania w kontekście edukacji wczesnoszkolnej. Czym jest programowanie? Jak i z pomocą jakich narzędzi wprowadzać je u najmłodszych? Odpowiedzi na te pytania poprzedzają kluczowy dział zawierający propozycje aktywności uporządkowanych w pięciu grupach.

Poradnik metodyczny zawiera gotowe rozwiązania. Jednocześnie opisane w nim aktywności zachęcają do konstruowaniu różnorodnych rozwiązań i interpretacji zgodnie z potrzebami dzieci oraz indywidualnym stylem pracy każdego nauczyciela. Wiele propozycji bazuje na pracy w parach lub w małych grupach. Stwarza to warunki do rozwijania przez uczniów umiejętności komunikacji, radzenia sobie z trudnymi sytuacjami, planowania i podejmowania odpowiedzialności za własny proces uczenia się. Otwarta formuła proponowanych zabaw, gier i ćwiczeń edukacyjnych sprawia, że materiał może być wsparciem i inspiracją zarówno dla debiutujących, jak i doświadczonych praktyków.

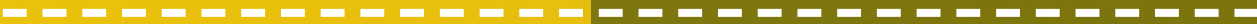
Układ *Poradnika* umożliwia realizację materiału zarówno linowo, ze stopniowym wprowadzaniem zagadnień w kolejności wypracowanej w toku kilkuletnich działań grupy trenerów pracujących z dziećmi z całej Polski, ale pozawala także na wykorzystywanie poszczególnych aktywności w sposób niezależny. W pierwszej serii poruszone są kwestie przepisu, sekwencji oraz precyzji komunikatów, które prowadzą do celu; pojawiają się też hasła ścieżek prostych ruchów umożliwiających poruszanie się w czterech wymiarach (pravo, lewo, góra, dół). Kolejnym zagadnieniem są pętle (powtórzenia). Dalej – podsumowanie sekwencji i ścieżek ruchu, w tym obrotów, z wykorzystaniem elementów muzyki. Następnym krokiem jest przesyłanie wiadomości (nadawanie, odbieranie) oraz instrukcja warunko-

wa. Ćwiczenia kończy układ współrzędnych w zabawach oraz synteza, podsumowanie i wykorzystanie wszystkich umiejętności. Przygotowane materiały zostały wzbogacone o informacje, które zabawę zamieniają w szeroką i wielopoziomową aktywność. Kontekst i wprowadzenie do zabawy stanowią więc istotną część tego podręcznika.

Jak każdą sztafetę, tak i ten poradnik metodyczny kończy meta, będąca tutaj krótkim podsumowaniem. Aby jednak do finału, dotrzeć należy najpierw rozpocząć bieg, a zatem..



GOTOWI



{1} Mitchel Resnick,
Let's teach kids to code,
2013, online:
[https://www.youtube.be
/watch?v=Ok6LbV_6bqaE](https://www.youtube.be/watch?v=Ok6LbV_6bqaE)
[dostęp 20.09.2018].

Jeśli sięgniemy do słownika języka polskiego, dowiemy się, że **programowanie** to między innymi pisanie programów komputerowych. Czy to oznacza, że na tym ma polegać praca w szkole z najmłodszymi? Oczywiście, że nie! Wprowadzając elementy podstaw programowania do edukacji, skupiamy się na rozwijaniu różnorodnych zagadek i problemów. Interesuje nas rozwijanie kreatywności, choć obok zabawy i przyjemności będą pojawiać się momenty trudne, wymagające zaangażowania, nauki i mierzenia się z problemami. Doświadczenie i obeznanie z technologią, jakie charakteryzuje najmłodsze pokolenia, zamieniane będą na biegłość w korzystaniu z technologii, rozumianą jako umiejętność wyrażenia siebie i prezentowania swoich pomysłów. Tak pojmowane programowanie jest pojęciem umożliwiającym dzieciom połączenie świata wirtualnego z realnym, otwiera przed najmłodszymi nowe możliwości.

W trakcie zabaw uczestnicy poszukują odpowiedzi na szereg pytań związanych z technologią: Jak działa komputer? Jak to możliwe, aby kolorowy film był tylko ogromem zer i jedynek? Czy programowanie to magia? Jak człowiek może rozmawiać z robotem? Dzieci mają też możliwość poznać nowy smak tworzenia – kreowania z wykorzystaniem technologii. Komputer przestaje być tylko narzędziem odtwarzania, biernego odbioru.

Jednak kodowanie to nie tylko nauka o technologii i z technologią. Współpracując z dziećmi, warto pamiętać o tym, że w przypadku podstaw programowania cel jest szerszy niż stworzenie jakiegoś produktu czy gry. Dorosły staje się wsparciem dla uczniów, a „**dzieci uczą się kodować, ale co ważniejsze kodują, by się uczyć**” {1}.

Nauka programowania zmienia podejście do całego procesu uczenia się, wyprowadzając dzieci z linearnego śledzenia wyznaczonej ścieżki, a otwierając przed nimi mapę, którą uczą się odczytywać, samodzielnie wybierając kierunki eksploracji. Kodowanie w szkole może, a wręcz powinno obejmować działania poprzedzające wprowadzenie języka programowania. Pracę z technologią warto uzupełniać o tzw. programowanie

bez prądu. Takie aktywności, realizowane bez użycia technologii, umożliwiają ćwiczenie istotnych elementów – jak konsekwencja, wytrwałość czy umiejętność współpracy w grupie. Co ważne, zabawy offline można realizować w dowolnym miejscu i momencie, czyniąc z nich działania o wysokim stopniu uniwersalności i dostępności.

Jednocześnie nauka poprzez zabawę „oznacza też przyswajanie sobie wiedzy (...) Gdy tylko dziecko dowiaduje się czegoś nowego, natychmiast w to się bawi, żeby przyswoić sobie nowe zjawisko i żeby przeistoczyło się ono z czystej koncepcji w coś uchwytnego” [2]. Co istotniejsze z perspektywy wprowadzania elementów programowania, warsztat nauczycieli edukacji wczesnoszkolnej stanowi szeroki wachlarz gier i zabaw. Warto z niego zatem korzystać, stale rozwijać i pogłębiać.

{ DOKĄD?

Dzięki nauce podstaw programowania dzieci mają szansę rozwijać się jako kreatywni twórcy. A ponieważ proces tworzenia związany jest z trudnościami, mogą ćwiczyć się w byciu ekspertami w rozwiązywaniu problemów. Radzenie sobie z niestandardowymi wyzwaniami o różnym

stopniu trudności i poziomach skomplikowania wydaje się stanowić istotę programowania. Niektóre z problemów wymagają szeregu czynności, wielokrotnych korekt i cierpliwej pracy – często nie jednej osoby, a całego zespołu. To sprawia, że programowanie sprzyja rozwijaniu umiejętności przybliżających dzieci do pozycji specjalistów od współpracy i komunikacji. Ta ostatnia zaś umiejętność może być rozwijana nie tylko poprzez prowadzenie dialogu w zespole pracującym nad rozwiązywaniem problemów, ale także w toku przedstawiania własnych projektów, dzielenia się pomysłami czy inspirowania innych.

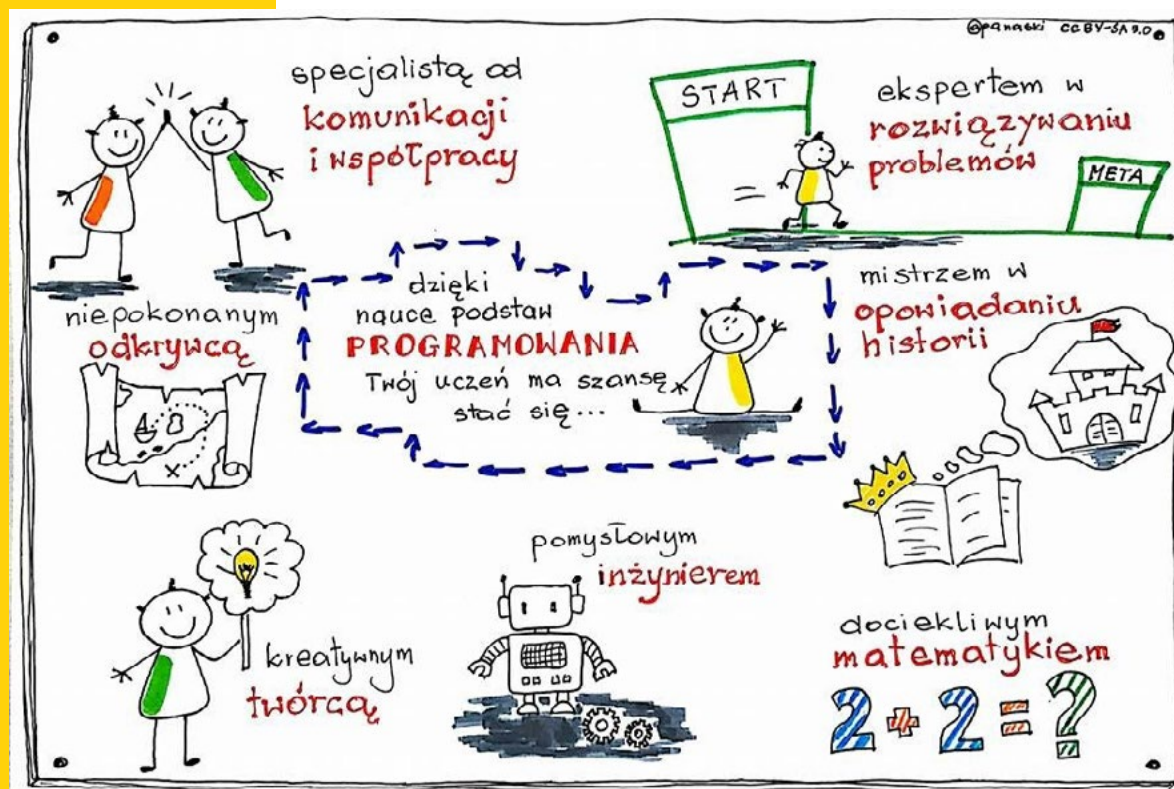
Zatem wprowadzenie podstaw programowania w edukacji wczesnoszkolnej to także przybliżenie dzieci do pozycji mistrzów w opowiadaniu historii. A osadzenie zadań w przestrzeni nauk ścisłych wzmacnia je w stawaniu się dociekliwymi matematy-

{ 2 } André Stern, *Zabawa. O uczeniu się, zaufaniu i życiu pełnym entuzjazmu*, Wydawnictwo Element, Gliwice 2017.

{ 3 } Mitchel Resnik, *Life-long Kindergarten. Cultivating Creativity through Projects, Passion, Peers, and Play*, John Wiley & Sons, Hoboken 2017.

kami, pomysłowymi inżynierami czy niepokonanymi odkrywca-
kami. Dzieci w swojej aktywności nie tylko definiują problem,
odkrywają i programują rozwiązanie czy testują opracowaną
ścieżkę rozwikłania problemu. Programowanie wykracza poza
realizację sekwencji działań, a wprowadza w sferę kultywowa-
nia kreatywności poprzez projekty, pasje i zabawę w gronie ró-
wieśników ^{3}.

Aktywności podejmowane w edukacji wczesnoszkolnej są nie
tyle układaniem kodu, co rozwijaniem podstawowych umiejęt-
ności przydatnych tak w procesie programowania, jak w życiu
w ogóle. Współczesny świat to miejsce dynamicznego rozwo-
ju i zmian, co sprawia, że stajemy przed wyzwaniami, z którymi
nikt wcześniej nie miał okazji się zmierzyć. Takie cechy, jak in-
nowacyjność i kreatywność, stają się kluczowe w perspektywie
konieczności wymyślania nowych rozwiązań, nie tylko w zawo-



dach związanych z technologią. Umiejętność pracy w grupie i komunikatywność – rozumiana m.in. jako umiejętność klarownego formułowania myśli – są wręcz niezbędne do sprawnej współpracy przy realizacji projektów, niezależnie od dziedziny wiedzy i przedmiotu.

Transformacja świata wynikająca z rozwoju technologii może otworzyć przestrzeń dla wielu ludzkich sukcesów, interesujących karier, ale niewątpliwie będzie także stwarzać problemy społeczne {4}. Komputeryzacja i robotyzacja świata stawiają przed edukacją nowe wyzwania, z którymi ludzie mogą się mierzyć, rozwijając w sobie to, co jest niedostępne technologii i robotom. Dzieci powinny mieć przede wszystkim szansę rozwijania równoległe kompetencji miękkich, takich jak niezależne myślenie, wartości i praca zespołowa {5}. Istotne jest tutaj wzmacnianie kreatywności, umiejętności badawczych oraz umiejętności znajdowania informacji. Nauka podstaw programowania nie oznacza więc, że każdy uczeń będzie w przyszłości programistą, ale będzie mógł w swoim życiu wykorzystywać elementy związane z szeroko rozumianym programowaniem.

W edukacji wczesnoszkolnej możemy realizować programowanie w korelacji międzyprzedmiotowej. Poprzez pracę metodą projektów lub wplatanie aktywności w ramowe zajęcia możliwa jest integracja kodowania z różnymi obszarami wynikającymi z realizacji podstawy programowej, a tym samym promowanie umiejętności czytania i pisania, matematyki, naukę, inżynierii i sztuki {6}.

{ JAK?

Rynek wydawniczy oraz producenci materiałów edukacyjnych i zabawek dość szybko zareagowali na pojawiające się potrzeby, wypełniając swoje oferty szeregiem produktów dotyczących programowania z dziećmi. Bogactwo propozycji sprawia, że przetestowanie wszystkich rozwiązań przez jedną osobę przestaje być możliwe – zarówno ze względów ekonomicznych, jak i czasowych.

{ 4 } Jack Ma on the IQ of love – and other top quotes from his Davos interview, 2018, online: www.weforum.org/agenda/2018/01/jack-ma-davos-top-quotes [dostęp 20.09.2018].

{ 5 } The future of education, according to experts at Davos, 2018, online: www.weforum.org/agenda/2018/01/top-quotes-from-davos-on-the-future-of-education [dostęp 20.09.2018].

{ 6 } Marina Umaschi Bers, *Coding as a Playground: Programming and Computational Thinking in the Early Childhood Classroom*, Routledge, Nowy Jork 2018.

Jak zatem odnaleźć się w gąszczu materiałów, informacji, robotów? Jak i które narzędzia wybrać? Jak odróżnić gadżet od narzędzia? Na te i podobne pytania nie ma jednoznacznej odpowiedzi. Każdy nauczyciel, kierowany własnym doświadczeniem lub potrzebami swojej grupy, może wskazać inne produkty jako najbardziej wartościowe.

Istotnymi i wspólnymi cechami dobrych materiałów mogą być:

- **dostępność**, czyli możliwość zaopatrzenia się w nie niezależnie od miejsca zamieszkania;
- **przystępność**, czyli właściwa relacja ceny jednostkowej do funkcjonalności i możliwości wykorzystania;
- **funkcjonalność**, czyli możliwość opisanie danego narzędzia z perspektywy jego funkcji wynikających z konkretnych umiejętności kształtowanych podczas wykorzystywania go;
- **wielozadaniowość**, czyli możliwość wykorzystania jednego materiału w kilku aktywnościach i grach;
- **ciągłość i spójność**, czyli możliwość kompletowania materiałów tak, by wzajemnie się uzupełniały oraz umożliwiały pracę z coraz większą grupą uczniów.

Jednocześnie należy pamiętać, że żaden gadżet, robot czy tablet nie zastąpi relacji z drugim człowiekiem, a edukacja, również ta związana z programowaniem, czyli sferą technologiczną, powinna opierać się na komunikacji i współpracy. Te zaś wynikają m.in. z uważności, jaką ma dorosły względem potrzeb i możliwości grupy, z przestrzeni otwartej na popełnianie błędów, także tych będących udziałem samych nauczycieli, z elastyczności i otwartości na zmianę oraz z czasu – również tego, jaki rezerwujemy sobie, aby spokojnie przygotować się do zajęć. Te elementy sprawiają, że w poradniku nie ma scenariuszy zajęć, a propozycje aktywności, przy których wprowadzenia do danej zabawy są równie istotne jak sam opis jej potencjalnego przebiegu.



START



{ A JAK ALGORYTM

W pierwszej części, w sposób intuicyjny, na poziomie percepcji uczestników zajęć, wprowadzone zostaną podstawowe pojęcia programistyczne. Zastanowimy się: kim jest programista? Czym są sekwencje? Jak precyzyjnie przekazywać instrukcje? Czym są algorytmy i w jaki sposób można je wyrażać/konstruować? Gdzie w życiu codziennym znajdziemy algorytmy? Uczniowie przekonają się, że człowiek, wykonując jakieś czynności, wspiera się doświadczeniem i intuicją, natomiast robot czy komputer wykonuje działania ściśle według instrukcji, którą otrzyma.

PROGRAMISTA, CZYLI KTO?

Zanim odpowiemy sobie na to pytanie, możemy wykonać małą rozgrzewkę – zająć się tematami nieco odległymi od komputerów i programowania (choć niezupełnie).

Kto pracuje w szpitalu? Zapewne jedną z pierwszych odpowiedzi będą „lekarz” oraz „pielęgniarka”. Tak, to oczywiście dobre odpowiedzi. Choć już po chwili możemy je uzupełnić o bardziej szczegółowe informacje. Pojęcie „lekarz” można doprecyzować, podając miejsce pracy (lekarz w izbie przyjęć, oddział) lub specjalizację (specjalista chorób wewnętrznych, pediatra itp.). Pielęgniarka naczelna zarządza pracą pielęgniarek i pielęgniarzy, w tym epidemiologicznych oraz anestezjologicznych. Personel szpitala stanowią również dyrektor oraz dyrektor do spraw medycznych. Sprawna praca szpitala zależy też od obecności pracowników administracyjno-technicznych, w tym specjalistów wdrażających i nadzorujących działanie systemu informatycznego (co poniekąd przybliży nas znowu do kwestii technologii).

Podobnie złożony układ możemy odnaleźć, zadając pytanie o pracowników szkoły, urzędu czy fabryki. I choć na co dzień upraszamy pewne pojęcia, aby szybciej się komunikować, to

warto pamiętać, że czasem zbytne uproszczenia prowadzić mogą do nieporozumień. Dobrze jest w odpowiednim momencie właściwie precyzować swoje wypowiedzi.

Rzecz ma się podobnie w dziedzinie technologii. Powszechnie używa się hasła „informatyk”, choć coraz częściej obok pojawia się pojęcie „programista”. Naturalnie obydwie te nazwy są pewnym uproszczeniem.



W trakcie rozmowy z dziećmi zapytajmy o to, co wiedzą o programowaniu, kim jest programista, czym się zajmuje, gdzie pracuje, jakie znają urządzenia, które dadzą się zaprogramować.

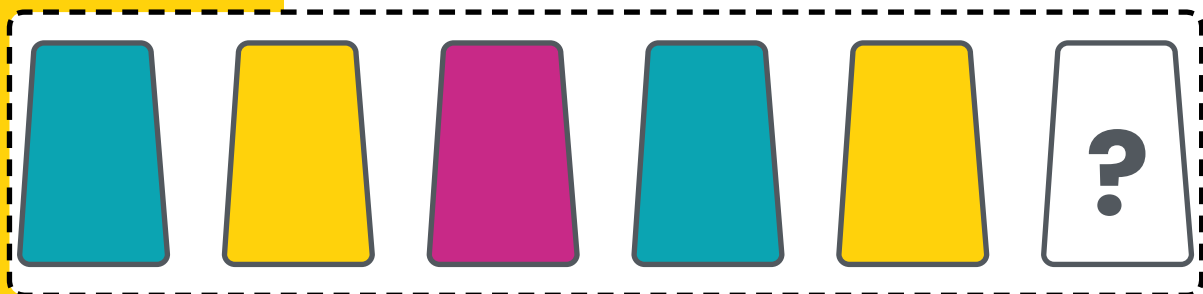
W pracy z najmłodszymi uczniami istotne jest **zasygnalizowanie różnorodności** zajęć, które wykonuje „programista” (programista aplikacji lub systemowy, inżynier oprogramowania, projektant oprogramowania, analityk systemów itp.; na finalny efekt pracy mają wpływ również osoby współpracujące z twórcą kodu, np. tester).

Istotną kwestią jest również zasygnalizowanie kwestii **cech i umiejętności programisty**, rozumianych tutaj jako kompetencje miękkie w procesie tworzenia (kreatywność, komunikacja, umiejętność współpracy).

SEKWENCJE

Jedną z bazowych umiejętności programistycznych jest umiejętność logicznego myślenia, czyli rozumowanie związane ze zdolnością do integrowania informacji. Dzięki niemu możliwe jest zrozumienie sensu wypowiedzi czy czytanej książki, sprzyja też rozwiązywaniu poważniejszych problemów. Umiejętność logicznego wnioskowania można w pewnym stopniu trenować poprzez rozwiązywanie różnych zadań i zagadek – to dlatego jednym z pierwszych zadań w nauce podstaw programowania może być aktywność związana z sekwencjami, czyli ciągami logicznymi. Taki uporządkowany ciąg znaków, symboli, zdarzeń itp. stanowi strukturę większego systemu, a jego elementy występują w nim zgodnie z określoną regułą lub formułą.

Rozkładamy przed uczniami zestaw kolorowych kubeczków w określonej kolejności, np.:



Pytamy, jaki kubek należy ułożyć na kolejnych miejscach. Co będzie na miejscu 10, 15? Pytamy, jaka część tego układu się powtarza. Prosimy uczniów, aby ułożyli własny ciąg logiczny i udali kilka podobnych pytań swoim kolegom.

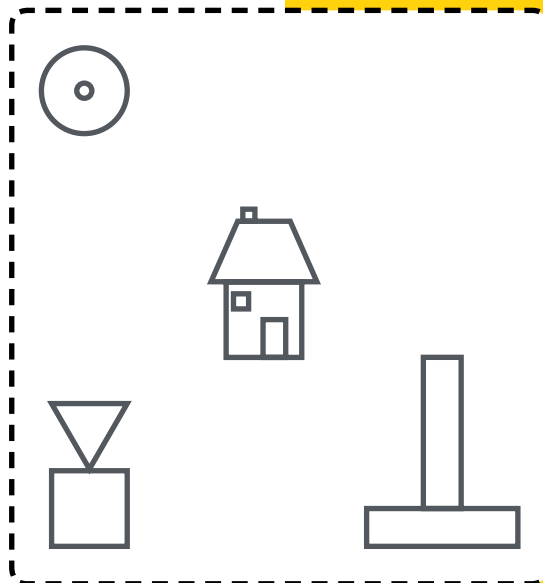
NARYSUJ TO, CO JA

Umiejętność precyzyjnego przekazywania informacji i ich odbioru jest przydatna nie tylko w procesie programowania. Warto ją rozwijać, by poprawiać swoją skuteczność i efektywność w codziennym życiu.

„Narysuj to, co ja” jest jedną z propozycji zabaw, które mogą sprzyjać rozwijaniu umiejętności komunikacji, realizowanej poprzez dostatecznie szczegółowe opisywanie stanu, z dostosowaniem języka do możliwości odbiorców.

Na kartce (np. post-it) przygotowujemy ilustrację zawierającą kilka elementów, których poziom skomplikowania uzależniony jest od poziomu i możliwości grupy. Przykładowo:

Chętny uczeń podejmuje się przekazania pozostałym dzieciom informacji, co znajduje się na karteczce. Całość procesu odbywa się wyłącznie z wykorzystaniem słów, bez pokazywania ilustracji oraz bez użycia gestów. Komunikaty mogą być na różnym poziomie szczegółowości oraz odnosić się do indywidualnej wiedzy i doświadczeń. Przykładowo:



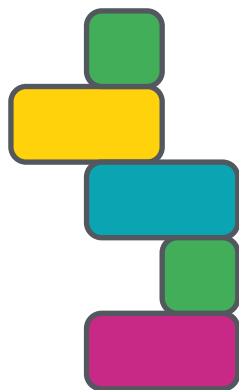
- W lewym górnym rogu narysuj okrąg i w środku małe kółko.
- Punktem startowym jest lewy górny narożnik. Odmierz około 2 centymetry w prawo i 2 centymetry w dół. Ten punkt jest środkiem okręgu o średnicy 1,5 centymetra. Narysuj okrąg, a następnie zaznacz środek, rysując koło o średnicy 3 milimetrów.

Pozostali uczniowie mają za zadanie wykonać swój rysunek, korzystając jedynie z poleceń, które usłyszą i z tego, co podpowiada im doświadczenie. W trakcie tej zabawy grupa rysująca na swoich kartkach nie może zadawać pytań ani konsultować się.

Czy wszyscy w równym stopniu wykonali zadanie? Co było łatwe? Co sprawiało trudność? Jakie komendy się sprawdzały i u którego odbiorcy? Jak inaczej sformułować pewne polecenia, by były uniwersalne i przystępne dla całej grupy?

ZBUDUJ TO, CO JA

W komunikacji czasem zadajemy dodatkowe pytania lub precyzujemy swoje wypowiedzi. W przypadku programowania oczywiście są możliwe korekty kodu, jednak generalnie instrukcja, jaką wydamy maszynie, czy program, który napiszemy, będzie zrealizowany tak, jak je zaprogramujemy. Maszyna nie będzie zadawała pytań i nie będzie dążyła do precyzowania zadania. Błąd, który znajduje się na początku programu, może wykluczyć wykonanie zadania, nawet jeśli dalsza część kodu jest poprawna. Zatem od początku należy podjąć się próby przewidzenia i zaplanowania swoich decyzji. Czasem dopiero po wykonaniu całego programu jest możliwość zweryfikowania i ewentualnie poprawienia kodu. Podkreślamy, jak ważna jest precyzja – zarówno w sposobie komunikacji, jak i kolejności przekazywanych informacji.



Taką precyzję można ćwiczyć między innymi podczas aktywności, która jest modyfikacją zabawy związanej z ilustrowaniem, czyli podczas zabawy konstrukcyjnej „Zbuduj to, co ja”.

Uczniowie dobierają się w pary. Każda para otrzymuje klocki, np. po dwa z każdego koloru i długości, równo rozdzielone pomiędzy osoby z pary. Uczniowie siadają plecami do siebie. Jeden z nich buduje ze swoich klocków pewien układ, minibudowlę, wykorzystując dowolną liczbę klocków ze swojego zestawu. Przykładowo:

{ Wskazówka: To zadanie można zmodyfikować. Umieszczamy w pudełku pewną budowlę, dzielimy uczniów na mniejsze zespoły i każdy zespół wybiera tzw. gońca.

Twórca pierwszego układu klocków przekazuje informacje na temat budowy z klocków w taki sposób, aby druga osoba z pary odtworzyła układ. Uwaga: ważną zasadą jest wprowadzenie jednorazowości komunikatu. Oznacza to, że zdanie opisujące dany fragment może zostać podane tylko raz, a osoba wykonująca budowlę według instrukcji nie może zadawać dodat-

kowych pytań. Po skończonym zadaniu uczniowie porównują swoje budowle.

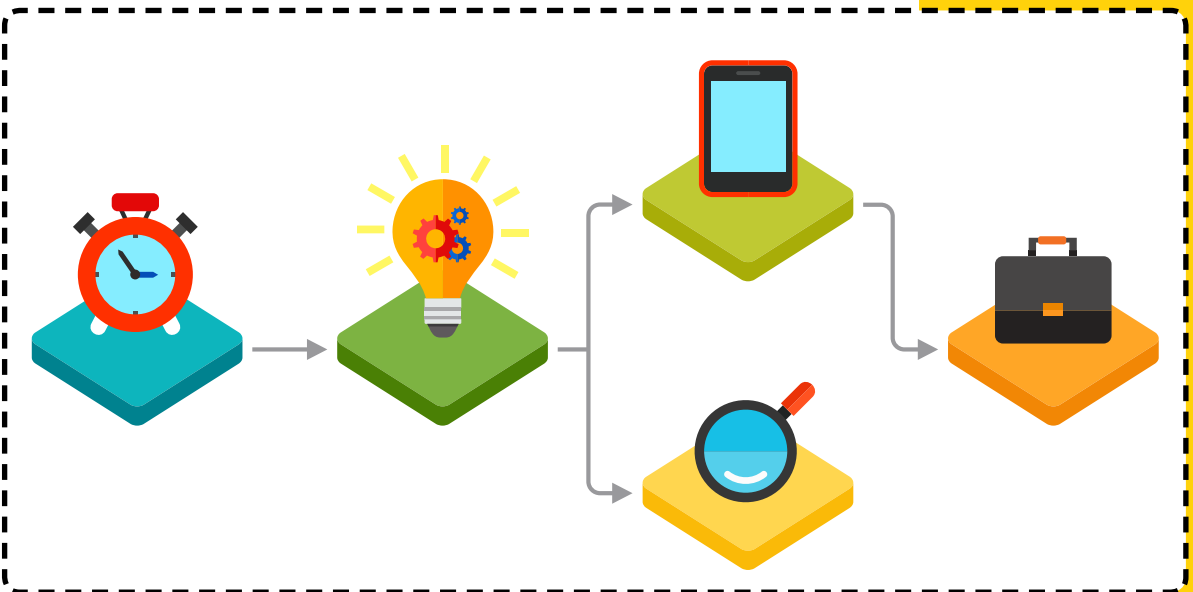
Czy udało się wykonać zadanie? Co było łatwe? Co sprawiało trudność?

ALGORYTMY W NASZYM ŻYCIU

Mówiąc o programowaniu, trudno pominąć algorytm, który potocznie rozumiany jest jako sposób postępowania, schemat czy przepis. Na co dzień stosowanie takiego uproszczenia jest zasadne. Jednak w kontekście wprowadzenia podstaw programowania warto pamiętać, że choć nie ma jednej uniwersalnej definicji algorytmu, to jest to pojęcie szersze niż ciąg czynności, które musimy wykonać, aby wypełnić jakieś zadanie.

Historia algorytmów sięga czasów starożytnych oraz pierwszych opisów greckiego matematyka i filozofa Euklidesa. Uczony wymyślił pierwszy znany przepis na realizację zadania, jakim był algorytm znajdowania największego wspólnego dzielnika

Goniec podbiega do pudełka, obserwuje, jak wykonana jest budowla, a następnie wraca do grupy i przekazuje jej informacje. Zadaniem zespołu jest zbudowanie tej samej budowli, która znajduje się w pudełku w oparciu jedynie o obserwację i przekaz informacji „gońca”.



{7} Piotr Fulmański, *Wstęp do informatyki Algorytmy i struktury danych*, 2015, online: http://fulmanski.pl/zajecia/wdi/zajecia_20152016/wyklad_pres/pres_pl_algorithm.pdf [dostęp 20.09.2018].

Artur Bartoszewski, *Algorytmy – metody prezentacji i zapisu*, 2015, online: [http://www.bartoszewski.pr.radom.pl/pod_pi/wstep\(pd\)_4.pdf](http://www.bartoszewski.pr.radom.pl/pod_pi/wstep(pd)_4.pdf) [dostęp 20.09.2018].

dwóch dodatnich liczb całkowitych. Samo słowo algorytm pochodzi z nieco późniejszego okresu i wywodzi się od nazwiska matematyka arabskiego, Muhammada ibn Musa al-Chorezmiiego. Opisał on procedury (np. dodawania, odejmowania, mnożenia i dzielenia zwykłych liczb dziesiętnych) w tak precyzyjny sposób, że to właśnie od jego nazwiska pochodzi słowo algorytm {7}.

Zanim jednak dzieci poznają definicję algorytmu – w matematyce oraz informatyce określanego jako skończony, uporządkowany ciąg jasno zdefiniowanych czynności, koniecznych do wykonania pewnego zadania – powinny dowiedzieć się, że algorytmy mają sześć cech odróżniających je od niealgorytmów:

- wyróżniony początek, czyli moment, który inicjuje;
- wyróżniony koniec, który oznacza zakończenie realizacji algorytmu;
- określony stan, do którego dążymy (warunek efektywności);
- jednoznaczne i zrozumiałe w interpretacji komunikaty;
- wykonanie zadania zamyka się w określonej liczbie czynności (warunek skończoności);
- całość procedury możemy zastosować do podobnych zadań (warunek uniwersalności).

Spróbujmy w toku rozmowy z uczniami opracować własną, wstępną wersję.

Proponowaną aktywnością w tej części jest swobodna rozmowa, w trakcie której dzieci podejmą się próby znalezienia odpowiedzi na pytanie, czym jest algorytm. Możliwe, że uczniowie sami zaproponują definicję składającą się z przykładów będących prostymi algorytmami obecnymi w życiu codziennym, takimi jak mycie zębów, ubieranie się na sanki albo realizacja przepisów kulinarnych. Na etapie wczesnoszkolnym nie ma potrzeby wprowadzania bardziej precyzyjnych i formalnych wyjaśnień. Wystarczy intuicyjna definicja oparta na skojarzeniach. Choć warto w trakcie zajęć zasygnalizować cechy decydujące o tym, że jest to algorytm, a nie tylko ciąg czynności, metoda czy schemat.

Pracę nad zagadnieniem algorytmów warto wesprzeć materiałami filmowymi pokazującymi znaczenie algorytmów w życiu codziennym.

Filmy w języku angielskim:

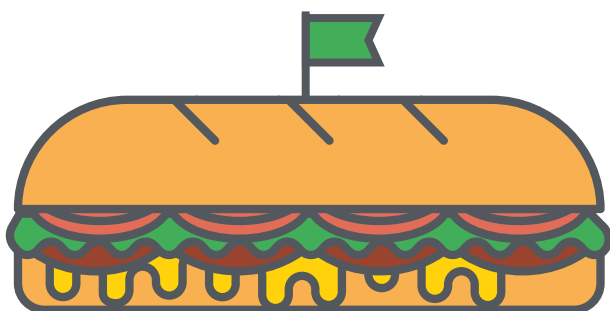
- Unplugged – Real-Life Algorithms: Paper Airplanes
<https://youtu.be/AWqo8Gxtrjs>
- Unplugged – Real-Life Algorithms: Planting a Seed
<https://youtu.be/FHsuEh1kJ18>
- Unplugged – Real-Life Algorithms: Dice Game
<https://youtu.be/wQpgBCKivAs>

Film w języku polskim:

- Jak to działa? Algorytmy
<https://vod.tvp.pl/video/jak-to-dziala.algorytmy,24941486>

PYSZNE ALGORYTMY

Zabawą umożliwiającą wprowadzenie pojęcia algorytm jest aktywność „Robimy kanapki”, która za sprawą prostoty przygotowania oraz wysokiej obrazowości zyskała sporą popularność.



Na stole przygotowujemy składniki oraz narzędzia, np. opakowanie chleba tostowego, masło, dżem, talerz oraz nóż. Następnie prosimy uczniów, aby na kartkach zapisali instrukcję wykonania kanapek z dżemem. Dzieci mogą pracować w parach

lub indywidualnie. Po skończonej pracy nauczyciel podejmuje się próby wykonania kanapki według podanej instrukcji, odczytując polecenia bez interpretacji oraz pomijając swoje doświadczenie oraz intuicję. Zatem wszelkie niejasne i nieprecyzyjne zapisy będą skutkować wykonaniem zadania niezgodnie z intencją twórców. W trakcie wykonywania czynności mogą – a nawet powinny – pojawić się więc głosy uczniów, że nauczyciel robi coś źle, że nie to mieli na myśli.

Realizowanie instrukcji literalnie pozwoli zweryfikować uczniom swoje przepisy i uświadomić, jak ważne jest precyzyjne konstruowanie poleceń. Maszyny nie potrafią się domyślać, a jedynie wykonują polecenia mechanicznie.

W części podsumowującej można zapoznać się z realizacją podobnego zadania przez inne dzieci. W internecie znajdziemy sporo materiałów prezentujących procedurę wykonywania kanapek, na przykład: Exact Instructions Challenge – THIS is why my kids want to kill me. | Josh Darnit https://youtu.be/cDA3_5982h8

Zadanie to znakomicie zamknie rozmowa podsumowująca to, co wydarzyło się w czasie zajęć. Poza istotnymi aspektami, jakimi są kolejność wykonywanych czynności i precyzja w opracowaniu algorytmów, warto poruszyć kwestię ich przydatności w naszym życiu w kontekście komunikacji z komputerami, robotami i innymi maszynami. Warto też poruszyć kwestię radzenia sobie z wyzwaniem, opanowania frustracji w sytuacji, gdy przygotowany przez nas projekt nie jest dostatecznie dobrze dopracowany.

{ B JAK BINARNY SYSTEM

Komputery są urządzeniami, które wykonują polecenia zaprogramowane przez człowieka. Ponieważ podstawą elektroniki jest prąd elektryczny, który w układach elektronicznych albo płynie, albo nie (a ściślej: ma niską lub wysoką wartość napięcia), komunikacja ludzi z kompute-

rami i wszystkie informacje zapisane w pamięci komputera zostały zakodowane wyłącznie przy pomocy dwóch stanów: włączony (1) i wyłączony (0).

W tej części poradnika znajdują się propozycje aktywności wprowadzających uczniów w podstawy systemu binarnego oraz komunikacji z komputerem.

SYSTEM BINARNY

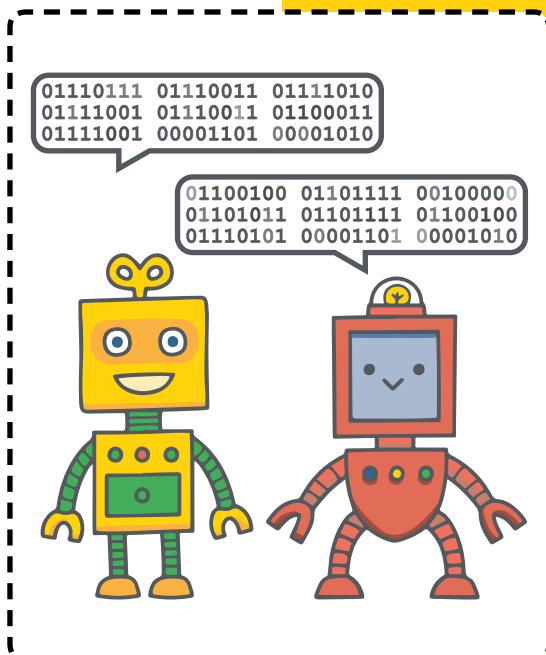
Przygodę z systemem binarnym można rozpocząć od zaprezentowania uczniom obrazu symbolizującego zapis zerojedynekowy oraz od zainicjowania rozmowy.

Zapytaj, czy dzieci wiedzą, co to jest. Z czym kojarzy im się powyższa grafika? Czy kiedykolwiek spotkały się z takimi obrazami i gdzie je widziały? Powiedz, że w tym zapisie kryje się jakaś wiadomość i że dalsza część lekcji pozwoli zrozumieć, w jaki sposób komputery „odczytują” informacje.

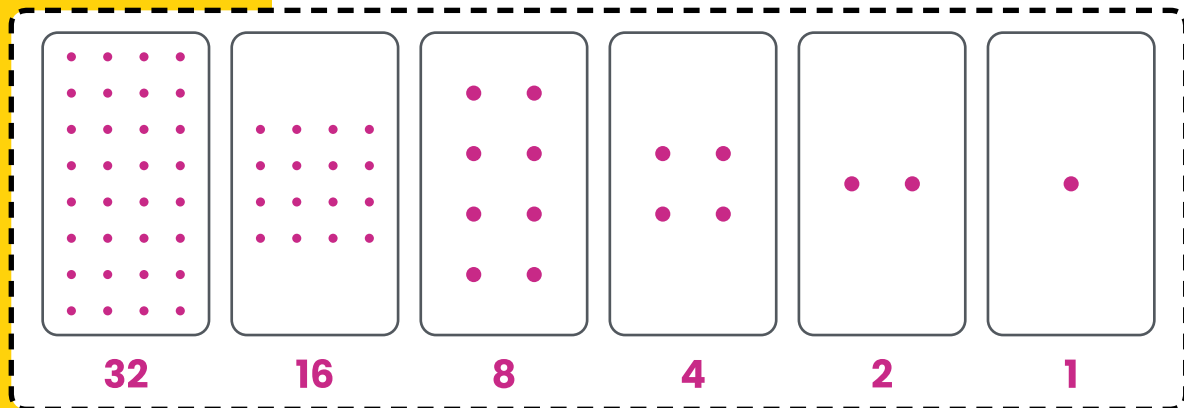
Przygotuj karty dostępne w załączniku. Warto wydrukować je na papierze, który z drugiej strony będzie np. czarny. Opcjonalnie można połączyć dwie kartki – białą z nadrukiem oraz od spodu kolorową.

Zaprezentuj karty i poproś dzieci, aby je opisały. Zwróćcie wspólnie uwagę, że liczby podwajają swoją wartość. Poproś uczniów, aby ułożyli te karty w jakiś logiczny sposób. Najprawdopodobniej ułożą je rosnąco. Jeśli pojawi się taki układ, należy doprecyzować narastanie od prawej, do lewej strony, analogicznie do zasady budowania liczb w systemie dziesiętnym.

Karty z kropkami symbolizują wartości liczbowe zapisane w systemie binarnym. Takie pojedyncze cyfry liczby binarnej to

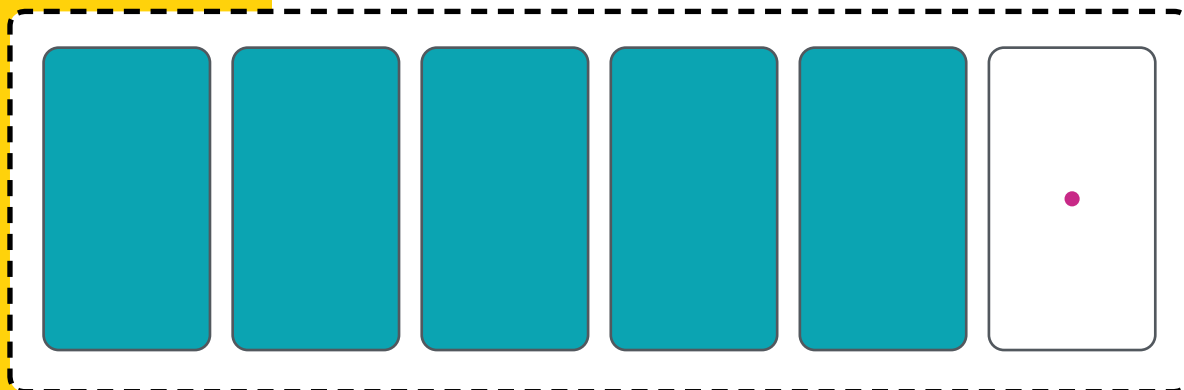


bit. Nazwa bit jest skrótem wyrażenia *binary digit* (z ang. – cyfra binarna). To najmniejsze jednostki informacji, dzięki którym komputery przechowują, zapisują, odczytują różne dane.



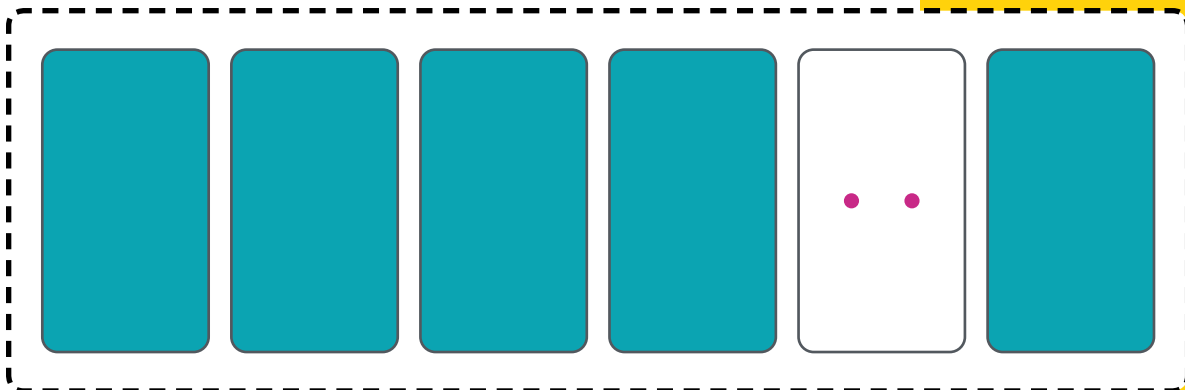
Karta z bitem odwrócona kolorową stroną oznacza brak, 0. Widoczne kropki wskazują 1.

Zaprezentuj poniższy układ kart. Ile jest widocznych kart? Ile z nich ma ustawienie w pozycji 0, a ile w pozycji 1? Ile jest widocznych kropek na karcie?



Komputer liczbę 1 zapisze przy pomocy pustych i pełnych bitów, czyli zer i jedynek w następujący sposób: 000001.

Zaprezentuj kolejny układ i postępuj analogicznie do wcześniejszego ćwiczenia.



Poproś, aby uczniowie połączyli się w pary i rozdaj każdej zestaw kartoników. Zadaniem dzieci będzie ułożenie kart tak jak na początku, a następnie odkrycie trzeciej karty (z czterema kropkami) i zakrycie pozostałych. Zapytaj, jaka to liczba i jak zapiszemy ją przy pomocy systemu 0, 1. Zapiszcie tę liczbę np. na arkuszu rozwieszonym w łatwo dostępnym i widocznym miejscu.

Zapytaj uczniów, jak zapisać liczbę trzy, skoro nie ma tylu kropek na jednej karcie. Jeśli uczniowie wpadną na rozwiązanie, poproś, aby układali kolejne liczby i podawali je w systemie binarnym.

Gotowa plansza z liczbami może wyglądać w następujący sposób:

1	000001	7	000111	13	001101	19	010100
2	000010	8	001000	14	001110	20	010101
3	000011	9	001001	15	001111	21	010111
4	000100	10	001010	16	010000	22	011001
5	000101	11	001011	17	010010	23	011010
6	000110	12	001100	18	010011		

NAPISZ SWOJE IMIĘ

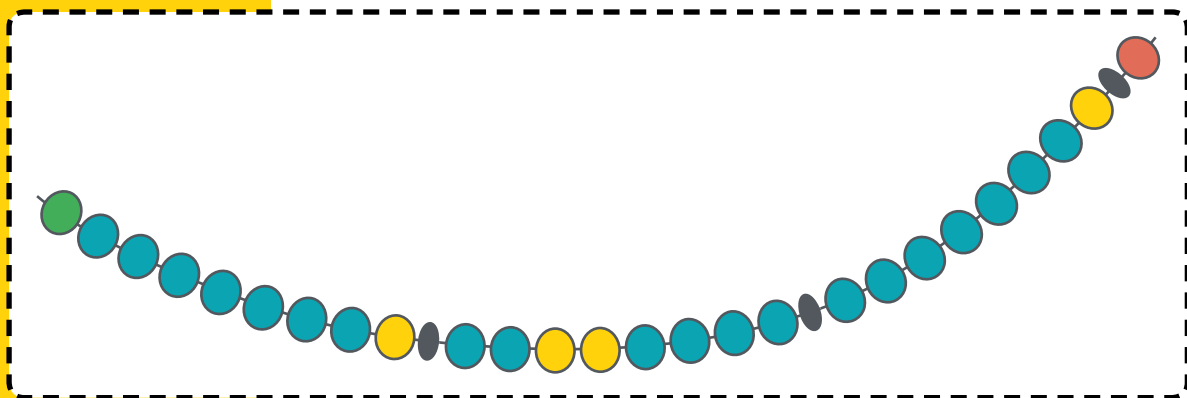
Łącząc w grupy bity 0 i 1, możemy tworzyć grupy bitów, zwane bajtami (z ang. *byte*). Takie standardowe grupy bitów można przechowywać w pamięciach komputerów, na nośnikach danych oraz przysyłać za pomocą sieci. Dzięki grupowaniu bitów w bajty możliwe jest zakodowanie informacji poprzez przyporządkowanie określonej wartości literom, cyfrom, znakom przestankowym i innym symbolom oraz poleceniom sterującym.

W tej aktywności do każdej liczby z wcześniej przygotowanej tabeli przypisana zostanie litera alfabetu łacińskiego (zatem z pominięciem polskich znaków).

A	000001	G	000111	M	001101	T	010100
B	000010	H	001000	N	001110	U	010101
C	000011	I	001001	O	001111	W	010111
D	000100	J	001010	P	010000	Y	011001
E	000101	K	001011	R	010010	Z	011010
F	000110	L	001100	S	010011		

Korzystając z takiej uproszczonej tabeli, można kodować słowa. Na przykład imię ALA z wykorzystaniem systemu binarnego zostanie zapisane jako: 000001 001100 000001.

Poproś, aby każdy uczeń zapisał swoje imię. Osoby, którym uda się skończyć szybciej, mogą zakodować jakąś krótką informację, słowo do odszyfrowania później przez grupę. Zaproponuj też rozszerzenie tabeli tak, by umieścić w niej również litery obecne w polskim alfabetcie.



KUBECZKOWE MASTERMIND

Proste aktywności związane z systemem binarnym krok po kroku przybliżają nas do zagadnienia grafik komputerowych. Zanim jednak dotrzemy do hasła „piksele”, poćwiczmy szyfrowanie informacji za pomocą barw. Kubeczkowe MasterMind to zabawa ruchowa z elementami kodowania i odkodowywania układu kolorów.

Celem zabawy jest odkodowanie przez drużynę prawidłowego układu kolorów, uprzednio zakodowanego przez „strażnika kodu”.

Podziel dzieci na kilkusobowe zespoły. Przygotuj miejsce do zabawy tak, aby zapewnić przestrzeń umożliwiającą krótki bieg. W jednej części sali umieść przegrody, które będą wyznaczały stanowiska „strażników kodu”. Mogą to być pudełka lub położone na bok ławki. Ważne, aby w bezpieczny sposób można było oddzielić strażnika kodu od pozostałych członków zespołu. Strażnikiem będzie osoba wybierana do danej rundy z grona każdego z zespołów.

W trakcie każdej rozgrywki pomiędzy strażnikiem i pudełkiem znajdują się cztery kubki w czterech kolorach. Taki sam zestaw kubków otrzymuje drużyna ustawiona naprzeciwko strażnika w odległości kilku metrów. Strażnik ma za zadanie ułożyć sekwencję kolorów, ustawiając kubki za przegrodą tak, aby dru-

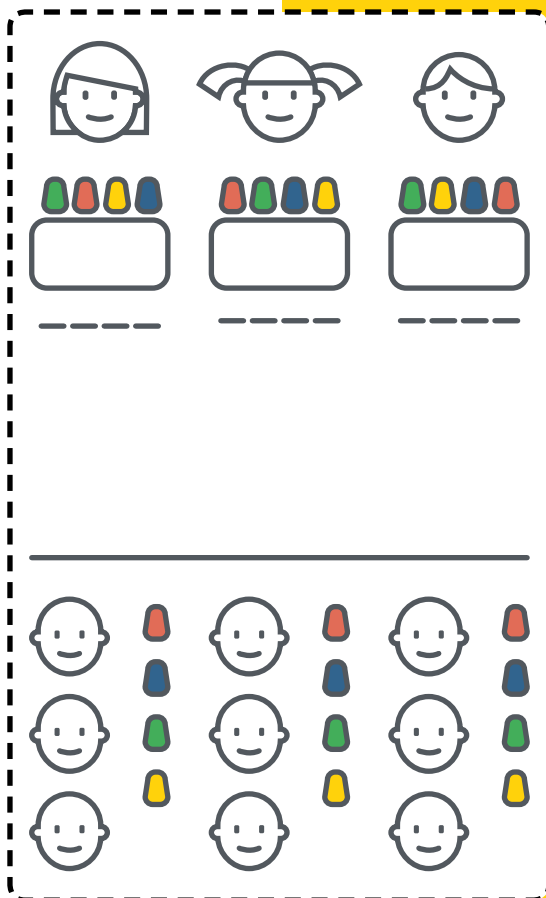
żyna ich nie widziała. W trakcie rozgrywki strażnik nie może modyfikować ustawienia swojego kodu, nie może też bezpośrednio informować zespołu o zbudowanym układzie. Może natomiast obserwować działania członków zespołu budującego swój układ klocków oraz kodować informację na temat poprawności działań członków swojej drużyny. Do tego celu nadają się małe znaczniki (np. nakrętki, klocki), które może ustawiać na prawidłowo ustawionym przez zespół kubku.

Na ustalony sygnał członkowie drużyny po kolei podbiegają do pudełka i stawiają przed nim jeden z kubków. Strażnik sprawdza, czy ustawiony kubek znajduje się w tym samym miejscu, co kubek tego samego koloru w jego sekwencji. Jeśli tak, kładzie na pudełku znacznik, wskazując poprawność ustawienia. Jeśli kubek nie znajduje się na właściwej pozycji, strażnik kodu nic nie robi ani nic nie mówi.

Gdy członek zespołu wróci do swojej drużyny, do strażnika rusza kolejne dziecko, z innym kubkiem i dostawia go obok wcześniej postawionego.

Uczniowie po kolei ustawiają – a na dalszym etapie przestawiają – kubki tak, aby uzyskać ustawienie w porządku ustalonym przez strażnika. Na etapie przestawiania kubków każdy uczeń może w jednym czasie zamienić miejscami tylko dwa kubki.

Punkt zdobywa ta drużyna, która jako pierwsza odgadnie sekwencję ułożoną przez strażnika. Strażnicy mogą się zmieniać. Gra może trwać np. do uzyskania ustalonej liczby punktów lub przez określony czas np. 10 minut.



Przykładowa rozgrywka

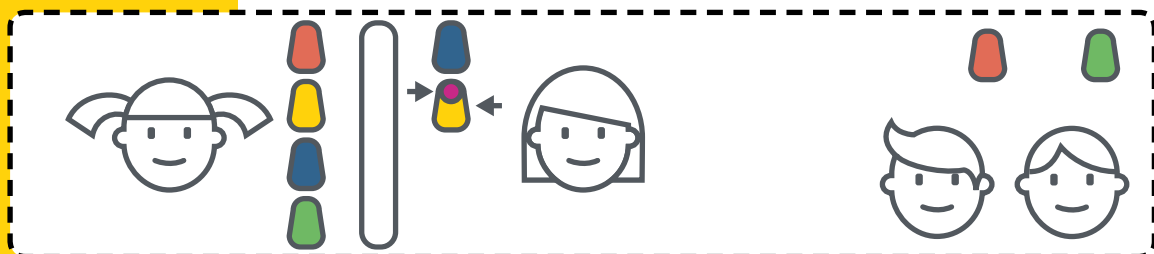
Strażnik kodu układa swoje kubki w sekwencji: czerwony, żółty, niebieski, zielony.



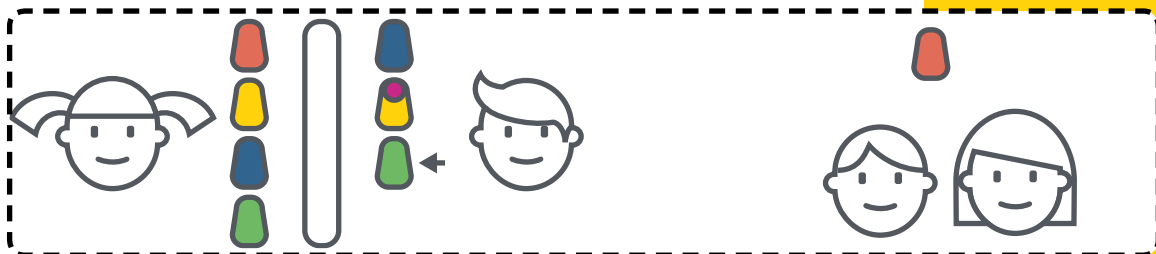
Pierwszy zawodnik rusza do strażnika, wybierając jeden z czterech kolorowych kubków. W tej sytuacji niebieski. Nie jest to poprawny kolor na tej pozycji więc strażnik kodu nie oznacza tego kubka jako prawidłowego.



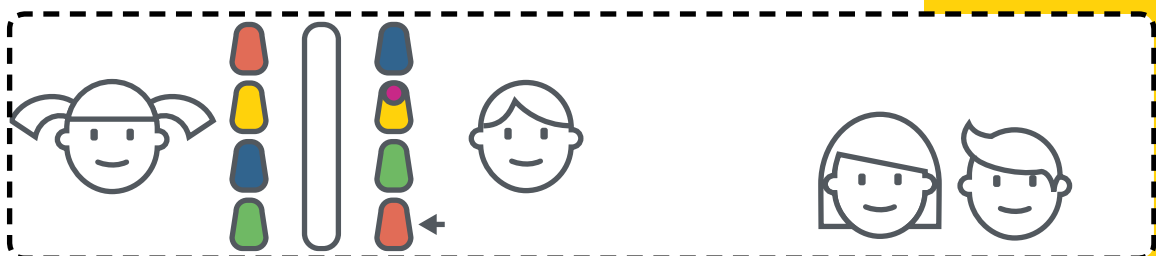
Po powrocie pierwszego zawodnika, drugi wyrusza, decydując się na przeniesienie kubeczka w żółtym kolorze. Jest to właściwy kolor kubka w tym miejscu, dlatego strażnik kodu oznacza kubek – kładzie na nim znacznik.



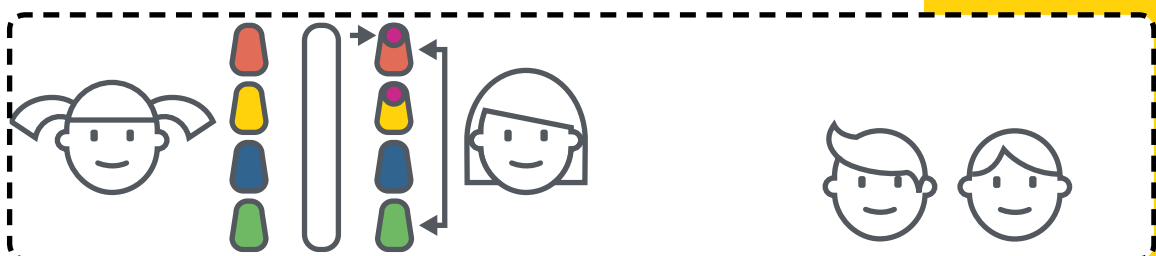
W trzeciej turze kolejny zawodnik przynosi zielony kubek. Nie jest to jednak prawidłowy kolor na trzeciej pozycji, a strażnik nie oznacza kubka.



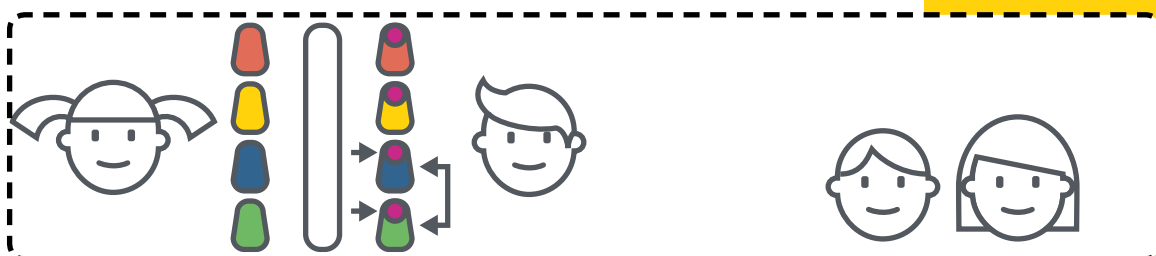
Ostatni bieg to przeniesienie czerwonego kubka.



Drugi etap gry to zamiany miejsc między kubkami. Każdorazowo zawodnik może zamienić miejscami dwa kubki. Zawodnik decyduje się na zamianę miejscami kubków czerwonego i niebieskiego. Strażnik kładzie znacznik na czerwonym kubku.



Kolejny zawodnik ma szansę doprowadzić układ do finalnego, dobrego ustawienia, zamieniając miejscami kubki niebieskiego i zielonego koloru.



BINARNY MASTERMIND

Tym razem uczniowie zamiast układu kolorowych kubków będą odszyfrowywali liczby. Grę można przeprowadzać w parach, w których dzieci na zmianę wykonują zadania lub są strażnikiem kodu.

Dziecko losuje liczbę zapisaną w systemie dziesiętnym, a następnie układa sekwencję kubków, kodując wartość w systemie binarnym. Kubek postawiony może reprezentować wartość 1, a kubek leżący – 0. Dziecko będące strażnikiem kodu musi pamiętać, aby za każdym razem, gdy jego kolega ustawi poprawnie dany element, zaznaczyć to przy pomocy klocka lub nakrętki, którą umieści na pudełku.

Przykładowa rozgrywka. Uczeń losuje liczbę 11, która w systemie binarnym ma zapis 001011. Dziecko układa kubeczki według poniższego wzoru, ale popełnia błąd w ustawieniu trzeciego i czwartego kubka:



Strażnik kodu oznacza prawidłowo ustawione kubki, udzielając informacji zwrotnej, na podstawie której gracz może skorygować kod:

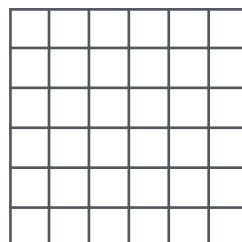


BINARNE OBRAZKI

Zabawa ta jest wstępem do działań związanych z zagadnieniem grafiki komputerowej i sposobu powstawania obrazów w komputerze. Co sprawia, że na monitorze wyświetla się grafika? Jak zera i jedynki zmieniają się w barwne obrazy? Co różni rysunki wykonywane przez ludzi na kartkach od tych wyświetlanych na ekranach komputerów? Co mają wspólnego?

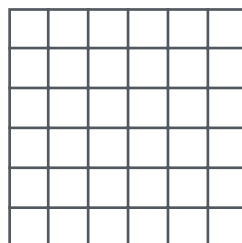
Binarne kody zostaną wykorzystane do tworzenia obrazków. Zero będzie oznaczało pustą kratkę, a jeden pełną. Uczeń otrzymuje kod zapisany w systemie dziesiętnym, np.:

0
18
0
18
30
0



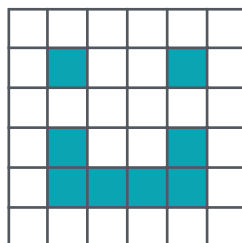
Następnie zapisuje liczbę przy pomocy systemu binarnego:

0 0 0 0 0 0 0 0
18 0 1 0 0 1 0 0 0
0 0 0 0 0 0 0 0 0
18 0 1 0 0 1 0 0 0
30 0 1 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0



Na koniec według tego binarnego układu wypełnia kolorem odpowiednie kratki:

0 0 0 0 0 0 0 0 0
18 0 1 0 0 1 0 0 0
0 0 0 0 0 0 0 0 0
18 0 1 0 0 1 0 0 0
30 0 1 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0



Uczniowie mogą teraz spróbować swoich sił w tworzeniu prostych obrazków, tworząc własne przykłady lub wykorzystując platformę code.org.

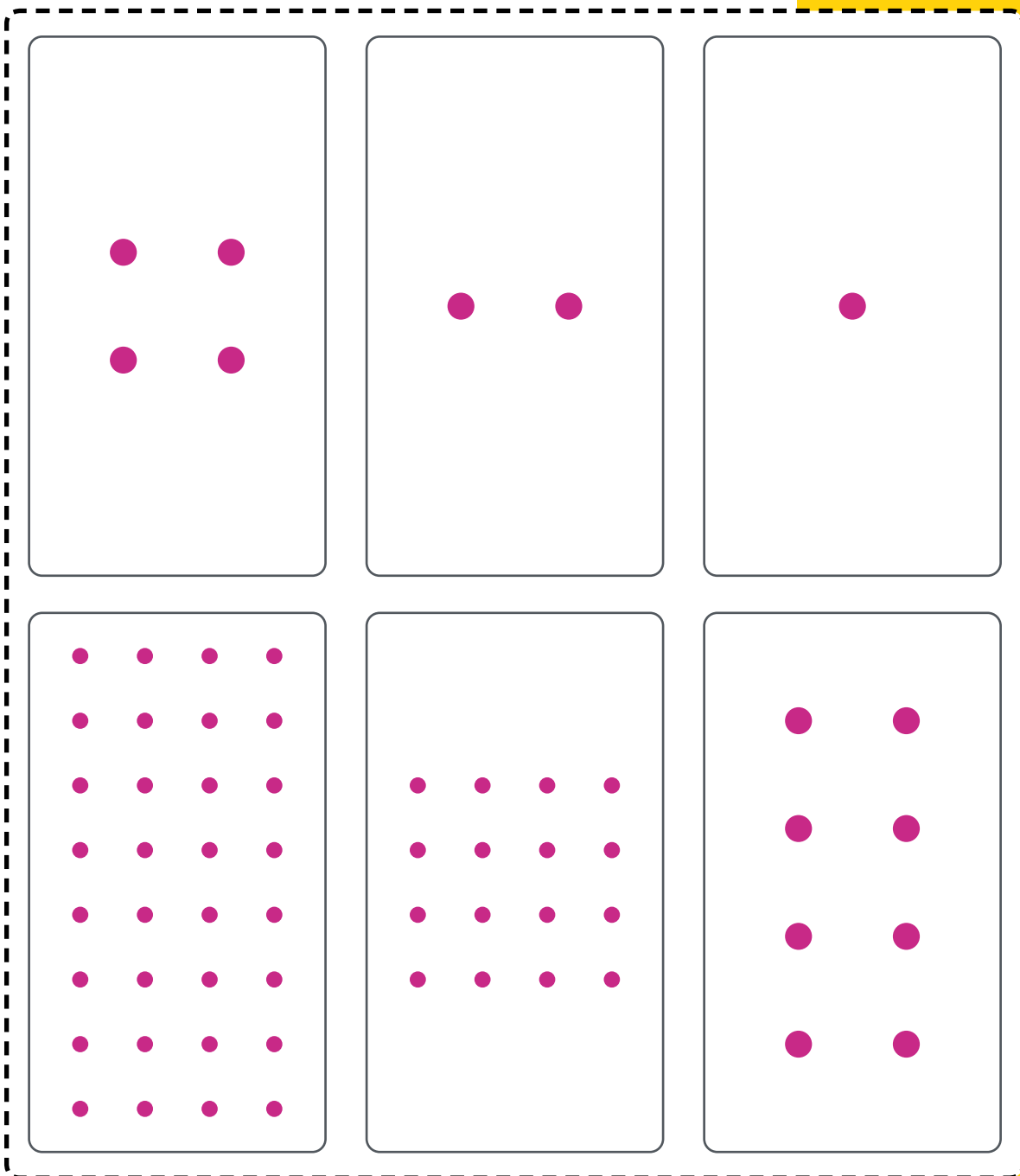
ZAGADKA DLA UCZNIÓW

0, 0												
3, 48												
2, 16												
2, 16												
31, 62												
31, 62												
8, 4												
28, 14												
20, 10												
0, 0												
0, 0												
0, 0												

Rozwiązanie:

0, 0												
3, 48				■	■	■	■					
2, 16				■			■					
2, 16				■			■					
31, 62	■	■	■	■	■	■	■	■	■	■	■	■
31, 62	■	■	■	■	■	■	■	■	■	■	■	■
8, 4		■							■			
28, 14	■	■	■					■	■	■		
20, 10	■		■					■		■		
0, 0												
0, 0												
0, 0												

KARTY DO ZAJĘĆ: BITY



KARTY DO ZAJĘĆ: ZAGADKA DLA UCZNIÓW

0
18
0
18
30
0

KARTY DO ZAJĘĆ: BINARNE OBRAZKI. PRZYKŁAD 1.

0, 0

3, 48

2, 16

2, 16

31, 62

31, 62

8, 4

28, 14

20, 10

0, 0

0, 0

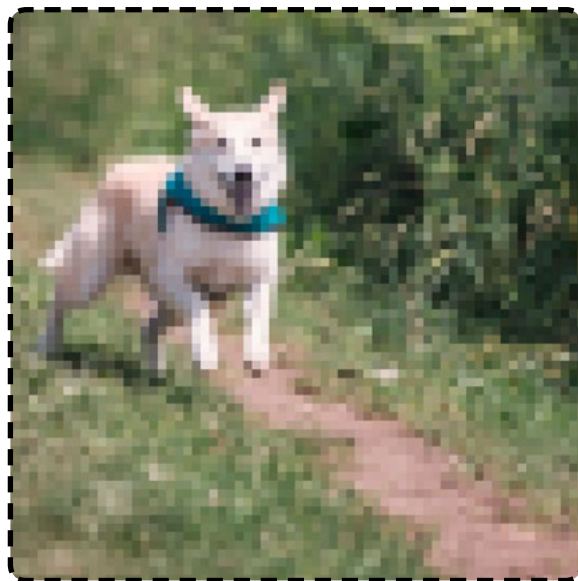
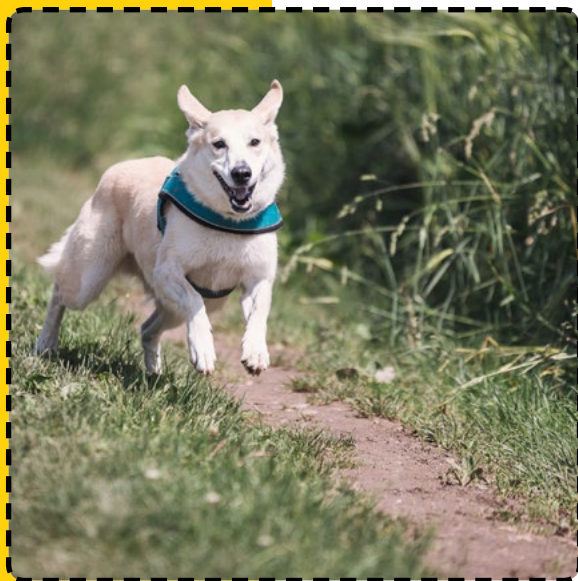
0, 0

PX, CZYLI PIKSELE

Wcześniejsze aktywności stanowią dobre wprowadzenie do zagadnień tej części poradnika. Z czego „zbudowane” jest zdjęcie lub rysunek, który jest wyświetlany na ekranie komputera? Jak to możliwe, że komputery, posługując się wyłącznie ciągami zer i jedynek, umożliwiają nam podziwianie filmów i barwnych grafik?

Przygotuj zdjęcie – możesz je pobrać ze źródeł zawierających fotografie na otwartych licencjach np. <https://commons.wikimedia.org> czy <https://pixabay.com/pl/>.

Wyświetl fotografię na ekranie lub z wykorzystaniem rzutnika. Przyjrzyjcie się zdjęciu w dużej rozdzielczości np. 3072x2304 px. Czy uczniowie coś dostrzegą? Spójrzcie na to samo zdjęcie, ale zapisane w mniejszej rozdzielczości np. 640x480 px. Czy coś się zmieniło? Powiększcie obraz do momentu, gdy będą wyraźnie widoczne piksele.



Jeśli nie ma możliwości wyświetlania zdjęć, aby zademonstrować, czym są piksele i jak są postrzegane, można wykorzystać edytor grafiki, np. Paint, lub wbudowany w programie Scratch tryb bitmapy.

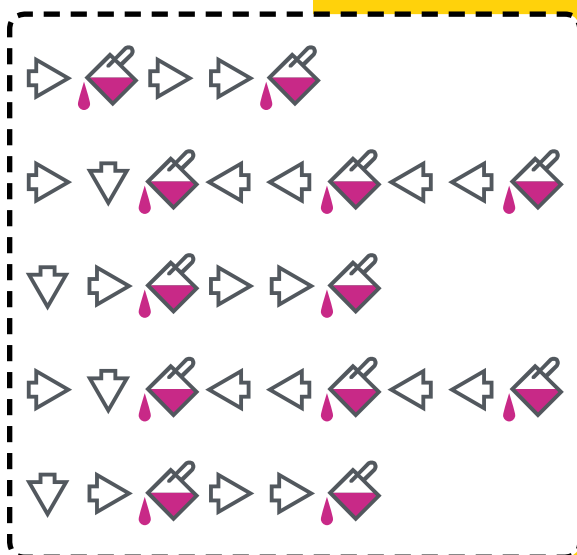
Piksele (ang. *pics*, skrót od *pictures* i *ele*, skrót od *element*) to najmniejsze jednolite, jednokolorowe elementy obrazu wyświetlanego na ekranie. Jeden piksel w przypadku monitorów komputerowych to bardzo mały kwadrat (choć pojawiają się niekiedy podziały obrazu, przy którym każdy piksel jest np. heksagonalny). Rozdzielczość monitora to liczba pikseli, jakie w pionie i poziomie ma matryca.

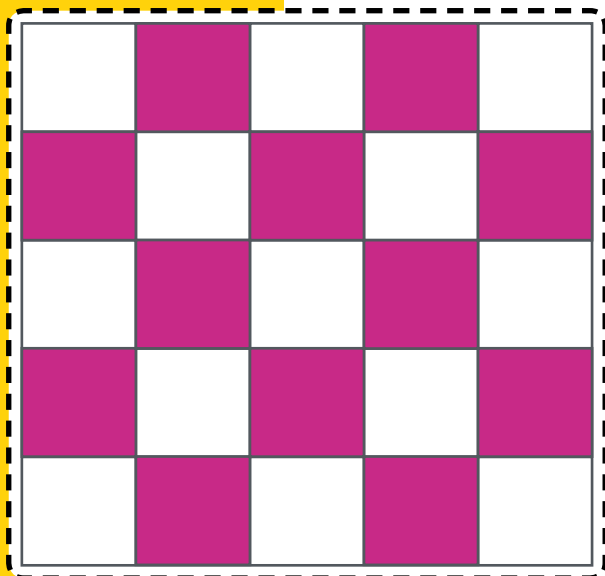
Zadbajmy o czas i przestrzeń na rozmowę o tym, jak powstają takie obrazy. Dlaczego tak się dzieje, że widzimy zdjęcie ułożone z drobnych kwadratów, a widoczne przedmioty wydają się mieć gładkie krawędzie? Jak można byłoby zaprogramować maszynę, aby wykonała jakiś rysunek na kratownicy? Propozycje podawane przez uczniów warto zapisać np. na arkuszu papieru lub tablicy. Być może pojawi się pomysł zamalowywania przez maszynę odpowiednich pól. Jeśli nie, możemy zaproponować dzieciom historię robota, która to będzie wprowadzeniem do kolejnej aktywności:

Mały robot poruszał się po planszy zgodnie z kierunkiem wyznaczonym przez strzałki. W wyznaczonych miejscach wylewał farbę, wypełniając w ten sposób pola...

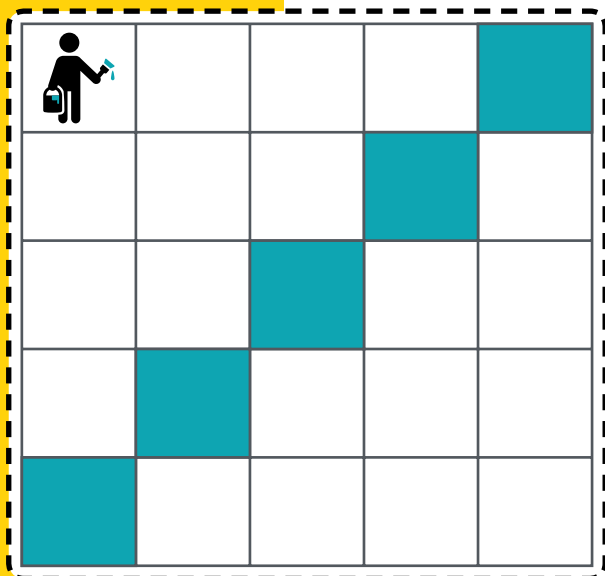
Stemplowanie

Przygotuj małego robota (zabawka lub model wykonany np. z papieru), kratownicę o wymiarach 5x5 krutek, kwadraty z kolorowego papieru w rozmiarze jednej kratki z kratownicy. Ustaw robota w lewym górnym narożniku. Zaprezentuj uczniom kod zadania:





Zrealizujcie wspólnie skrypt zadania, aby przećwiczyć i zaprezentować mechanizm aktywności, zanim uczniowie będą pracowali w parach lub indywidualnie. Przykładowo jeden uczeń będzie przesuwiał robota po planszy zgodnie z kierunkami wyznaczonymi przez strzałki. Drugi będzie „wylewał” farbę, czyli układał kolorowe kartki w wyznaczonym miejscu – tam, gdzie znajduje się robot. Pozostali uczniowie mogą kontrolować poprawność wykonania zadania. Rezultat widzicie po lewej stronie.



Zaproponuj uczniom wykonanie podobnego zadania, polegającego na zakodowaniu drogi malarza, który musi wylać farbę na określone pola. Przykładowo:

Zaproś uczniów do pracy w parach. Po skończonym zadaniu dzieci odczytują swoje kody. Możemy je zapisać na tablicy lub przygotować wystawę kodów, wywieszając je w miejscu umożliwiającym zapoznanie się ze wszystkimi pracami. Czy są różnice?

Możliwe, że pojawią się kody, w których postać malarza będzie przechodziła przez wszystkie pola. A może pojawią się również kody zoptymalizowane, czyli prowadzące ludzika przez minimalną liczbę pól w celu zamalowania tych wskazanych?

Różnorodność form zapisu ukazuje też poziom zróżnicowania grupy. Przeanalizujcie kody, omówcie optymalizację kodu zawierającego powtórzenia.

Na przykład:

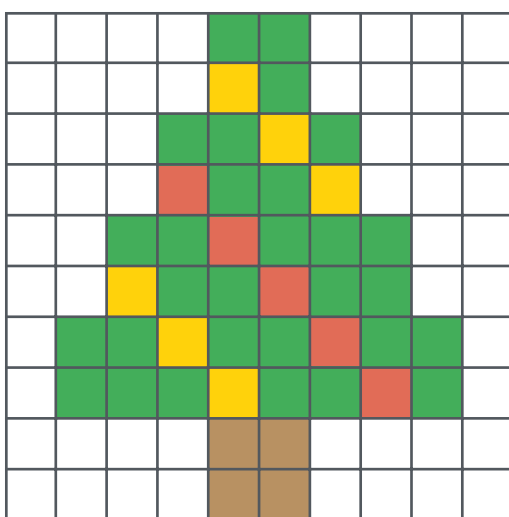


Modyfikacją zadania może być samodzielne tworzenie zakodowanych w ten sposób obrazków. Ciekawym wyzwaniem będzie taka zmiana kodu przez uczniów, aby w skrypcie pojawił się błąd. Zadaniem pozostałych jest wówczas wyszukanie tego błędu, czyli debugowanie... o czym szerzej w dziale **D jak debugowanie**.

PUSTE, PEŁNE

Zaprezentuj uczniom kratownicę o wymiarach 10x10 pól. Po lewej stronie ułóż według poniższej instrukcji liczby oraz oznaczenia kolorów. Liczby na białym tle oznaczają liczbę pustych pól, a liczby na kolorowych kartkach – liczbę pól, które należy wypełnić odpowiednim kolorem. Uzupełnijcie po kolei pola. Sprawdźcie, jaki rysunek został ukryty w kodzie.

			4	2	4		
		4	1	1	4		
	3	2	1	1	3		
	3	1	2	1	3		
	2	2	1	3	2		
	2	1	2	1	2	2	
1	2	1	2	1	2	1	
1	3	1	2	1	1	1	
			4	2	4		
			4	2	4		



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A															
B															
C															
D															
E															
F															
G															
H															
I															
J															
K															
L															
Ł															
M															
N															

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A															
B															
C															
D															
E															
F															
G															
H															
I															
J															
K															
L															
Ł															
M															
N															

Rozwiązanie widzisz po lewej stronie.

Zaproś uczniów do przygotowywania i kodowania własnych rysunków.

Zakodowane obrazki to pomysł, który można wykorzystać np. na korytarzach szkolnych. Można zakodować obrazki lub litery układające się w jakieś hasło. Wystarczy wydrukować planszę z kratownicą z zapisanymi obok liczbami oznaczającymi pola puste i pełne oraz poprosić uczniów, aby na przerwie zapełniali kratki – aż do momentu, gdy ujawni się ukryta zawartość.

LOSOWANE KOLOROWANKI

Innym pomysłem jest zabawa w kolorowanie według wylosowanych warunków. Wydrukuj planszę z kratownic z liczbami i literami taką, jak ta po lewej.

Obok kratownicy umieść wydrukowane kartoniki ze współrzędnymi i kolorami, jakimi należy zamalować te współrzędne. Zadaniem uczniów jest losowanie karteczek i zamalowywanie krerek odpowiednim kolorem. Efektem będzie kolorowy rysunek.

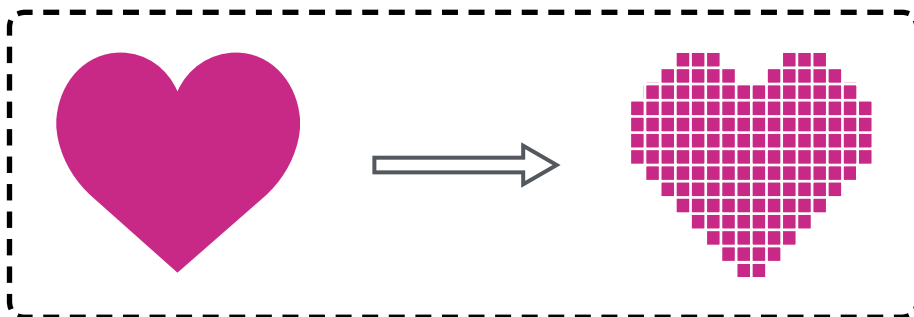
Link do przykładowych zakodowanych obrazów znajduje się tu:

<https://goo.gl/6eG5Un>

CYFROWY OBRAZ, CZYLI JAK „MALUJĄ” KOMPUTERY

Mając za sobą różne ćwiczenia i zabawy, powróćmy do pytania o to, w jaki sposób powstaje obraz na ekranie komputera. Jak aktywności z kolorowaniem pól łączą się z systemem binarnym? Jak to możliwe, że mając do dyspozycji tylko dwie wartości (0, 1) możemy podziwiać piękne fotografie, zaawansowane grafiki czy filmy?

Prześledź z uczniami dotychczasowe zabawy, spróbujcie omówić, czego się dowiedzieliście i czego doświadczyliście, poszukajcie w toku rozmowy odpowiedzi na pytanie, jak powstają barwne obrazy w komputerze.



Dane, które nie są liczbami (obrazy, dźwięki), muszą zostać zapisane w systemie binarnym, ponieważ komputery posługują się wyłącznie taką reprezentacją liczbową (0, 1). Obraz widoczny na monitorze komputera zbudowany jest z pikseli, z których każdy ma określony kolor oraz współrzędne.

Na przykład zapis:

000000 010010 000000 010010 011110 000000

umożliwia wygenerowanie prostego rysunku widocznego po prawej stronie.

0	0	0	0	0	0
0	1	0	0	1	0
0	0	0	0	0	0
0	1	0	0	1	0
0	1	1	1	1	0
0	0	0	0	0	0

01	01	01	01	01	01
00	00	00	00	00	00
00	11	00	00	11	00
00	00	00	00	00	00
00	10	00	00	10	00
00	10	10	10	10	00

Łącznie bitów w grupy umożliwia efektywniejsze kodowanie informacji. Dwubitowe grupy mogą kodować cztery kolory, np.: biały (00), żółty (01), niebieski (11) oraz czerwony (10). Zapis:







```
010101010101 000000000000 001100001100
000000000000 001000001000 001010101000
```

umożliwia wygenerowanie prostego rysunku widocznego po lewej stronie. Ilubitowy kod umożliwiłby nam zakodowanie sześciu lub ośmiu różnych kolorów?

(trzybitowy: 000, 001, 010, 011, 100, 101, 110, 111)

Kolory są najczęściej reprezentowane przy użyciu 8-bitowych pakietów, co oznacza zestaw 256 możliwych kombinacji zer i jedynek (od 0 do 255). Najpopularniejszym sposobem na reprezentowanie kolorów jest zastosowanie połączenia trzech barw światła: czerwonego, zielonego i niebieskiego (z ang. RGB, red, green, blue). Używając 8-bitów, otrzymujemy bogatą paletę barw – 256 kombinacji dla każdego z bazowych trzech kolorów, czyli prawie 16,77 milionów kolorów!

Przykładowe barwy:

Kolor	Wartości RGB	Wartości binarne RGB
 czarny	0, 0, 0	00000000, 00000000, 00000000
 biały	255, 255, 255	11111111, 11111111, 11111111
 czerwony	255, 0, 0	11111111, 00000000, 00000000
 zielony	0, 255, 0	00000000, 11111111, 00000000
 niebieski	0, 0, 255	00000000, 00000000, 11111111
 żółty	255, 255, 0	11111111, 11111111, 00000000

ZAŁĄCZNIK 2.: DYKTANDO GRAFICZNE – KARTA PRACY

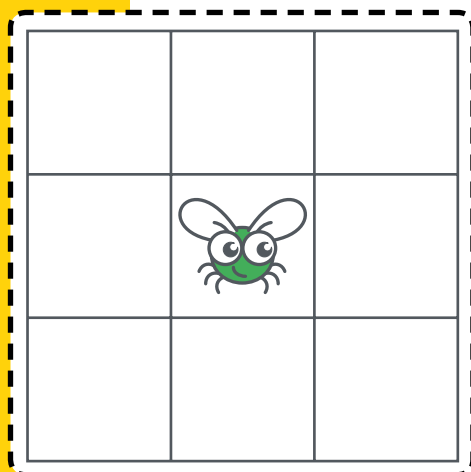
{ C JAK CIĄGI TRAS

{ 8 } *Mucha. Skoncentruj się!*, zamiastkserowki.edu.pl, CC BY-SA, online: <http://bit.ly/2xHdEOH> [dostęp 20.09.2018].

Prezentowane w tej części zabawy stanowią uniwersalną bazę, umożliwiającą spleatanie różnych treści edukacyjnych z podstawami programowania. W toku zabaw uczniowie utrwalą m.in. określanie kierunków poruszania się przedmiotów w dwuwymiarowej płaszczyźnie. Tworzenie zakodowanych ścieżek ruchu staje się okazją do ćwiczeń prostych pętli warunkowych, sekwencji, powtórzeń, procedur oraz kompresji danych, a więc optymalizacji zapisu. Aktywności stanowią też doskonały wstęp do debugowania, czyli pierwszych prób poprawiania prostych kodów.

Na początek zabawa „Mucha. Skoncentruj się!” { 8 } – prosta w przygotowaniu aktywność umożliwiająca poprzez dostosowanie szerokie wykorzystanie.

MUCHA – ZABAWA Z KIERUNKAMI



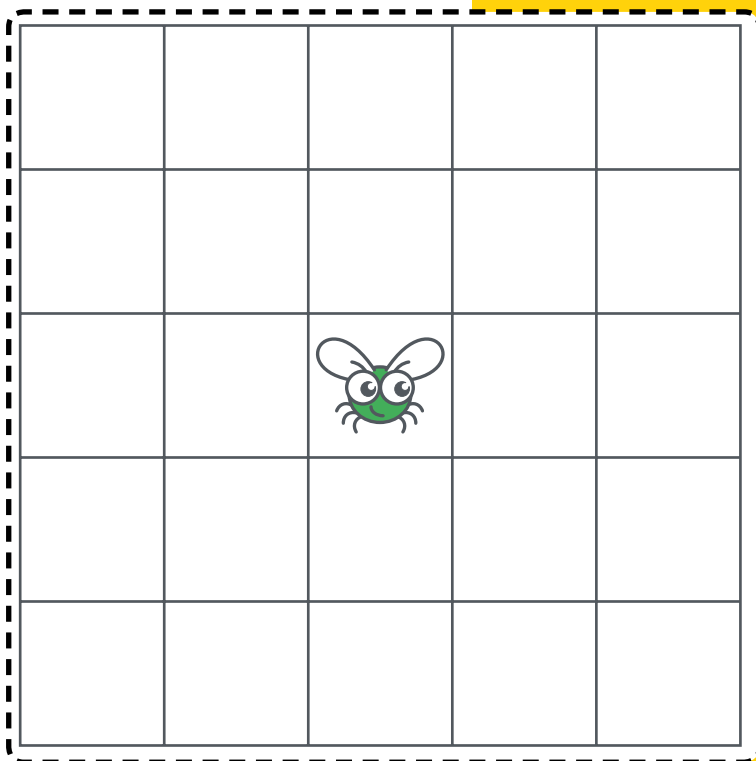
Wyznacz na kratownicy powierzchnię o wymiarach trzy na trzy pola. Na środkowym polu znajduje się postać, np. mucha.

Zabawa polega na wsłuchiwaniu się w polecenia sterujące muchą i poruszaniu nią w wyobraźni. Podczas całej rozgrywki mucha znajduje się na środku kratownicy, a uczestnicy kontrolują jej teoretyczną zmianę położenia jedynie w myśli. Nawigator wypowiadający polecenia ruchu używa określeń:

- mucha idzie w lewo,
- mucha idzie w prawo,
- mucha idzie w górę,
- mucha idzie w dół itd.

Pozostali uczestnicy zabawy, słuchając uważnie poleceń, wyobrażają sobie położenie muchy na planszy. Gdy mucha „wyjdzie” poza krawędzie planszy, dzieci sygnalizują to, wołając: „Wyszła!”.

Polecenia należy początkowo podawać w wolnym tempie tak, by uczestnicy zdążyli śledzić w wyobraźni ruch owada. Kolejne próby mogą odbywać się już przy narastającym tempie.



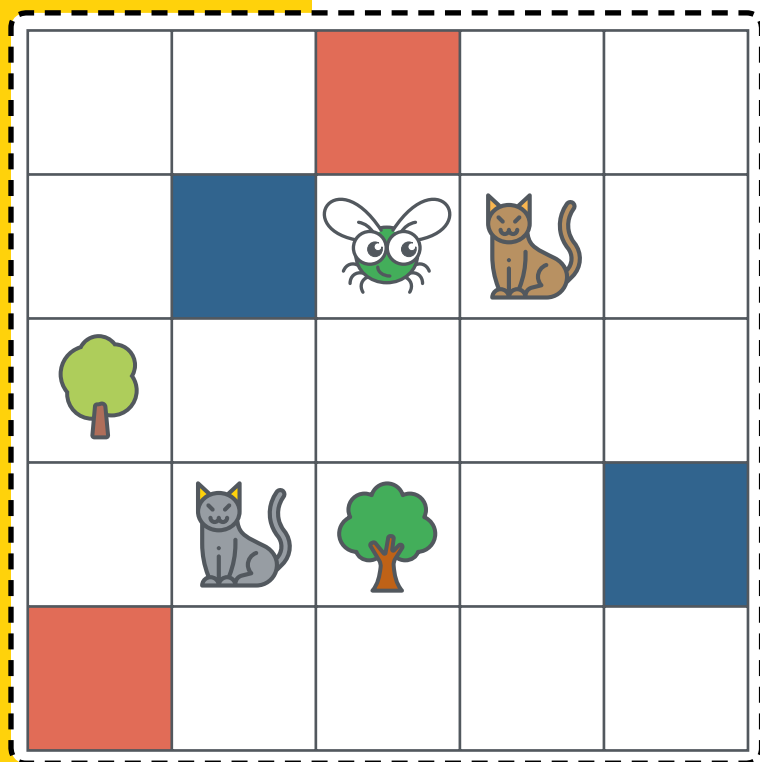
Utrudnieniem dla uczniów, poza przyspieszaniem tempa wypowiedzania komend, jest również powiększenie planszy, np. do wymiarów 5x5 pól. Warto z uwagą dokonywać zmian poziomu trudności tak, by nie wyeliminować z zabawy uczniów z zaburzoną percepcją wzrokową lub niewykształconymi dobrze umiejętnościami określania kierunku.

MUCHA. ZAREAGUJ!

Zabawę najlepiej przeprowadzić na powiększonej kratownicy, o wymiarach nie mniejszych niż 5x5 pól. Do przygotowania aktywności potrzebne będą różne obrazki lub kolorowe kartki, którymi zakodujemy zadania do wykonywania przez dzieci. Równie istotnym etapem przygotowującym jest rozmowa pozwalająca na odwołanie się do doświadczeń uczniów. Jakie mechanizmy są wykorzystywane w grach komputerowych? Czy dzieci znają gry, w których bohater, stając na jakimś polu, wywołuje

pewną reakcją? Przykładowo wskakując na kamień – słyszymy odgłos uderzenia, wchodząc na kładkę – słyszymy trzeszczenie, biegnąc po specjalnie wyznaczonych polach – uruchamiamy otwarcie jakiegoś tajemnego przejścia itd.

Zaproponuj uczniom zabawę z podobnymi mechanizmami. Postać muchy będzie poruszała się analogicznie do zasad ćwiczonych we wcześniejszym wariancie, jednak tym razem dodatkowo położenie muchy ma wywoływać odpowiednie reakcje u uczestników. Rozłóż na kratownicy kilka różnokolorowych kafelków oraz ilustracji. Na początek przygotuj tylko kilka pól funkcyjnych, trudność zadania zwiększysz wraz z nabywaniem przez dzieci sprawności w zapamiętywaniu i wykonywaniu zakodowanych znaczeń.



Ustal zasady pierwszej rozgrywki. Przykładowo, gdy mucha stanie na polu:

- w kolorze niebieskim, uczestnicy zabawy wydają dźwięk „plum”;
- w kolorze czerwonym, uczestnicy klaszczą raz w dłonie;
- z ilustracją drzewa, uczestnicy unoszą ręce w górę i wykonują gest machania, naśladując ruch gałęzi drzew na wietrze;
- z ilustracją kota, uczestnicy wydają dźwięk „miau”.

Grę można rozwijać i modyfikować. W kolejnej odsłonie zabawy na polach planszy ułóż liczby. Tym razem mucha będzie nie

wychodzić za planszę, ale zatrzymywać się na kolejnych polach. Uczniowie będą mieli za zadanie zapisać np. liczby, na których stanęła mucha, a następnie wykonać jakieś działanie z daną liczbą (pomnożyć przez 5, dodać dziesiątkę itd.). Na zajęciach z języka polskiego możemy na polach rozłożyć kartki z wyrazami np. zawierającymi ćwiczone trudności ortograficzne. W ramach edukacji przyrodniczej na macie mogą pojawić się np. ilustracje przedstawiające rośliny lub zwierzęta. Poruszanie po planszy można również zmodyfikować, wprowadzając kierunki czterech stron świata.

LABIRYNTY

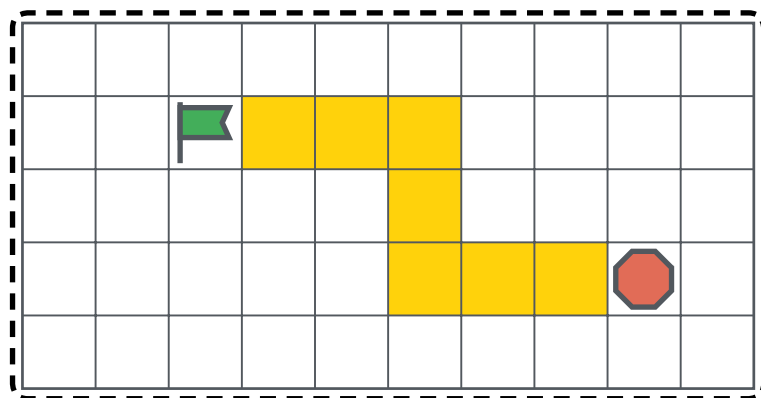
Zabawa labiryntami to doskonała okazja do nauki i rozwijania umiejętności związanych z budowaniem sekwencji skryptów pozwalających na precyzyjne poruszanie. To również ćwiczenie koncentracji uwagi, okazja do wspierania rozwoju logicznego myślenia. Dzięki zabawom labiryntami można wprowadzić i ćwiczyć proste instrukcje warunkowe, czyli instrukcje sterujące działaniami, przebiegiem programu.

Dobrze, jeśli instrukcja warunkowa pojawi się w formie zabawy, tak by umożliwić najmłodszym jej intuicyjne poznanie, choć osadzone na dobrze przygotowanym gruncie i przy wsparciu dorosłego, znającego teoretyczne podstawy.

Instrukcja warunkowa to realizacją programu w zależności od tego, czy wartość konkretnego wyrażenia jest prawdą czy fałszem. Podstawowym rodzajem instrukcji warunkowej jest **jeżeli... to...** Instrukcje warunkowe, obok pętli, stanowią konstrukcje programistyczne umożliwiające programom podejmowanie decyzji. To ważne zagadnienie z perspektywy pracy programisty... ale również istotne dla młodych użytkowników. Umożliwia bowiem doprecyzowanie hasła, że „komputery i roboty podejmują decyzję” poprzez podkreślenie aktywności człowieka, ponieważ decyzja maszyn odbywa się na bazie programu stworzonego przez ludzi.

Ćwiczenia z labiryntami rozpocznij prostą zabawą. Na kratownicy ułóż trasę z kilku kolorowych kafelków, zaznaczając jednocześnie start oraz metę. Na polu oznaczonym hasłem start umieść postać (np. maskotkę). Wyjaśnij, że to jest nasz bohater poruszający się po labiryncie krokami równymi polom, z których ułożona jest trasa (czyli jeden krok – jedno pole).

Poproś uczniów, aby spróbowali podać instrukcję przejścia postaci od startu do mety. Młodsze i mniej doświadczone dzieci może będą potrzebowały samodzielnego przejścia trasy, tak by ustalać i sprawdzać kierunek ruchu, położenie własnego ciała. Następnie zaproś do ułożenia instrukcji z wykorzystaniem symboli, np. strzałek, oraz oznaczeń startu i zakończenia.



Skrypt:

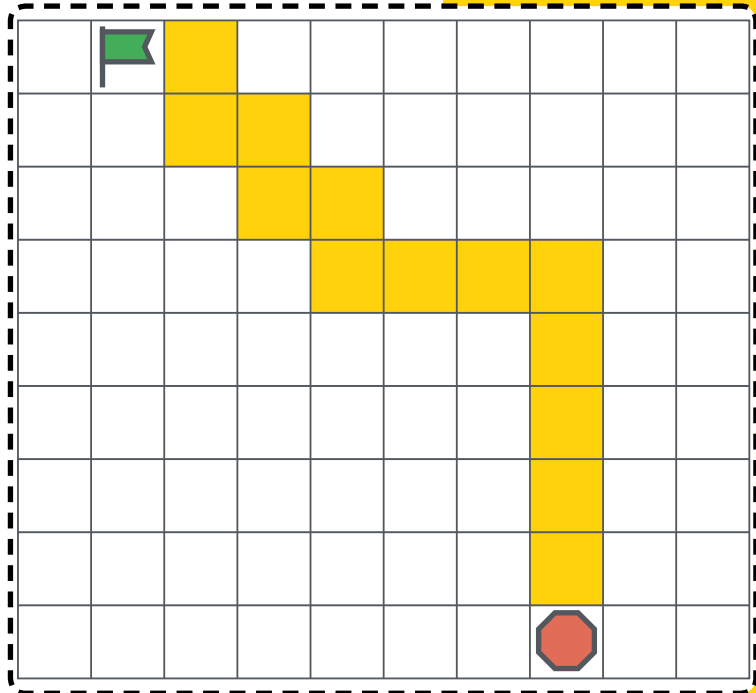


Umieszczanie symboli rozpoczynających i kończących skrypt jest istotnym elementem (patrz rozdział A jak algorytm), powtórzenie tych zagadnień będzie przydatne na późniejszym etapie zajęć z zakresu programowania, np. z wykorzystaniem aplikacji Scratch Jr lub Scratch.

Warto przetestować gotowy skrypt. Jedna osoba przyjmuje rolę nawigatora odczytującego kod. Druga może być beta-teste-

rem, przesuwanym po polach labiryntu postać sterowaną poleceniami kodu.

Jeśli zapis jest prawidłowy, zaproś dzieci to tworzenia innych labiryntów, także tych bardziej skomplikowanych. Przygotowanie rozbudowanych labiryntów może zaowocować stworzeniem długich skryptów. Jest to doskonały pretekst do tego, by w naturalny sposób wprowadzić drugi rodzaj spośród instrukcji sterujących, czyli **pętli**, pozwalających na wielokrotne wykonanie wybranego fragmentu instrukcji. Zastanówcie się razem, czy można skrócić tak długi skrypt. W jaki sposób? Czy w skrypcie pojawiają się fragmenty powtarzalne?



Przykładowe zadanie z optymalizacją znajduje się u góry po prawej stronie.

Przykładowy zapis skryptu:



Przykładowy zapis skryptu z pętlami:



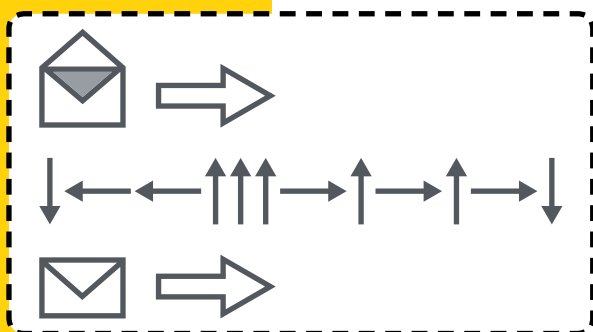
DYKTANDO GRAFICZNE

Nietypową formą tworzenia tras może być zabawa graficznymi dyktandami. Jej celem będzie ułożenie konturu zgodnie z instrukcjami, jednak realizowanymi z elementem instrukcji warunkowej.

Rozpocznij pracę od podziału uczniów na cztery zespoły. W każdym wyznacz jedną osobę – nawigatora, który będzie odczytywał kod. Pozostali członkowie grupy będą tzw. pilotami realizującymi zadanie.

Wręcz każdej grupie karteczki z zapisanym kodem, który jednak stanowi tylko fragment całego rysunku. Pierwszym zadaniem uczniów będzie odkodowanie instrukcji tak, aby rozpoznać moment inicjujący realizację otrzymanego fragmentu kodu.

Przykładowo w poniższej instrukcji realizację kodu rozpoczyna instrukcja warunkowa – jeżeli drużyna usłyszy klaśnięcie, wówczas wykonuje polecenia skryptu. Dodatkowo zespół będzie miał za zadanie zasygnalizować zakończenie realizacji swojego skryptu pukaniem:

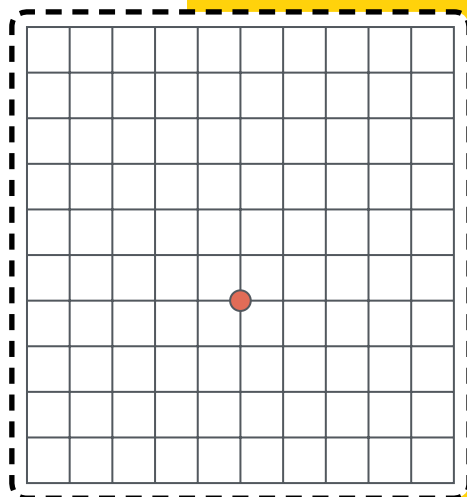


Kod zawiera symbole strzałek, które wskazują kierunek tworzenia konturu (np. rysowanie na kartce w kratkę, układanie na kratownicy pasków papieru wzdłuż krawędzi krutek).

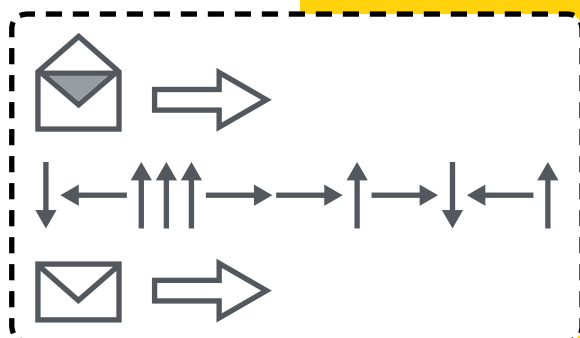
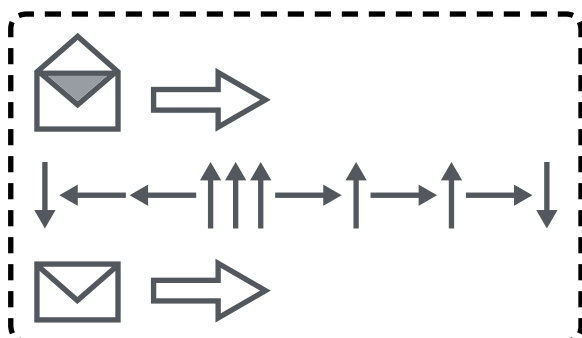
Drużyna, która jako pierwsza otrzyma kod inicjujący realizację kodu, rozpoczyna tworzenie konturu począwszy od wyznaczonego miejsca startowego.

Przykładowe położenie miejsca startowego na kratownicy możesz zobaczyć na kolejnej stronie.

Rozpocznij aktywność, nadając sygnał inicjujący działanie tej drużyny, która ma pierwszy fragment instrukcji. Zespół odbiera informację i zaczyna „rysowanie” konturu. Nawigator odczytuje kod zapisany na kartce wyrażony strzałkami (w górę, w prawo, w dół, w dół itd.), a na końcu „wysyła wiadomość” inicjującą działanie kolejnego zespołu.



Przykładowe instrukcje dla dwóch zespołów:



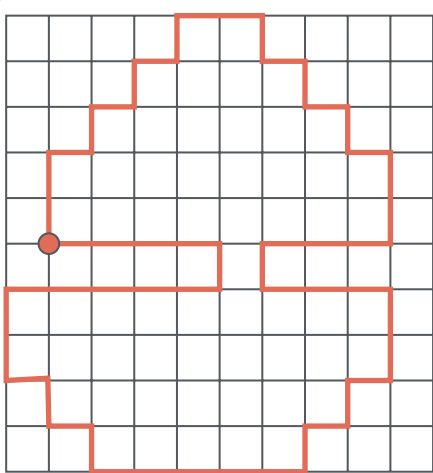
Na każdym etapie zabawy można pytać dzieci o ich przypuszczenia. Co będzie przedstawiał kontur?

Zadanie można rozwinąć: uczniowie wykonują rysunki, kodują je przy pomocy strzałek i wręczają koledze z ławki, aby ten odczytał kod i narysował według niego kontur.

Dopełnieniem tej części zajęć może być zadanie interaktywne „Artysta Kształty” będące częścią Kursu 1. na platformie code.org: <https://studio.code.org/s/course1/stage/10/puzzle/1>

Uczniom, którzy szybciej ukończą cały etap, można zaproponować trudniejszy, zawierający pętle <https://studio.code.org/s/course1/stage/18/puzzle/1>

Załącznik 1



Dyktando graficzne

Rozwiązanie obrazu
(rysowanie zaczyna się od kropki)

{ D JAK DEBUGO- WANIE

Przy wcześniejszych zabawach tu i ówdzie pojawiało się hasło debugowanie, czyli wyszukiwanie nieprawidłowości i porządkowanie skryptów. Nadszedł czas, by przyrzeć się bliżej pojęciu, które uznawane jest za ważny element programowania.

Debugowanie umożliwia znalezienie błędów w kodzie programu i następnie wyeliminowanie ich. Jest to jedna z bazowych czynności programistycznych, ponieważ w większych programach trudno uniknąć pojawienia się błędów.

Przydatne są przy tym cierpliwość i wytrwałość. Warto je rozwijać, ponieważ stanowią dobrą bazę przy realizacji innych projektów, nie tylko tych związanych z programowaniem.

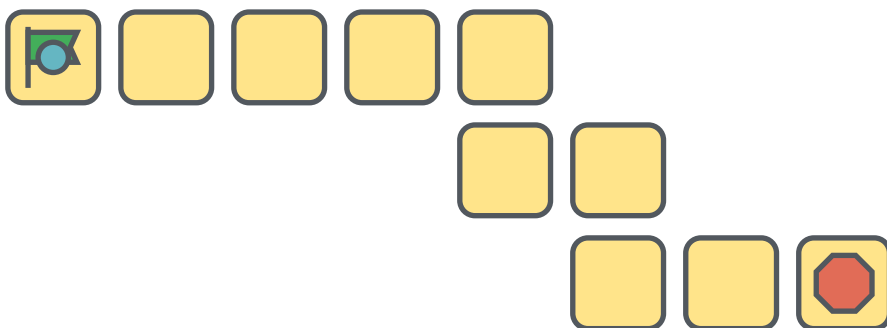
ZRÓB BŁĄD, ZNAJDŹ BŁĄD

Podziel uczniów na kilkusobowe zespoły. Praca będzie przebiegała w nietypowy sposób, bowiem z podziałem na etapy.

Etap 1.

Zadaniem zespołów jest ułożenie trasy labiryntu z karteczek post-it, z oznaczonym początkiem i końcem trasy. Na polu startowym należy położyć znacznik (np. maskotka, nakrętka).

Przykładowy układ:



Etap 2.

Grupy opuszczają swoje stanowisko, przesuając się o jedno miejsce, np. zgodnie z kierunkiem ruchu zegara. Zespoły mają teraz przed sobą układ labiryntu przygotowany przez inną grupę. Zadaniem uczniów jest zakodowanie drogi znacznika przy pomocy strzałek oraz symboli oznaczających początek i zakończenie skryptu. Po skończonej pracy grupa pozostawia swoją instrukcję przy labiryncie i ponownie zmienia stanowisko pracy, przesuując się ponownie w prawą stronę.

Przykładowy kod:



Etap 3.

Przy kolejnym stanowisku na grupę czeka zadanie polegające na sprawdzeniu poprawności znajdującego się tam kodu przygotowanego przez inną drużynę. Następnie na nowej karteczce grupa musi zapisać kod, celowo zmieniając kolejność niektórych strzałek tak, aby finalnie skrypt zawierał błąd. Kartkę z poprawnym kodem należy ukryć.

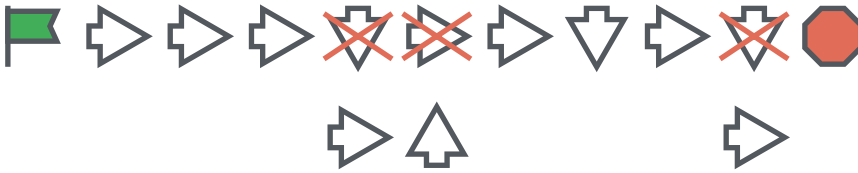
Przykładowy kod z błędem:



Etap 4.

Po kolejnej zmianie stanowiska grupa przystępuje do pracy polegającej na weryfikacji nowego kodu. Uczniowie znajdują błąd, a następnie go poprawiają.

Przykładowy kod z poprawionym błędem:

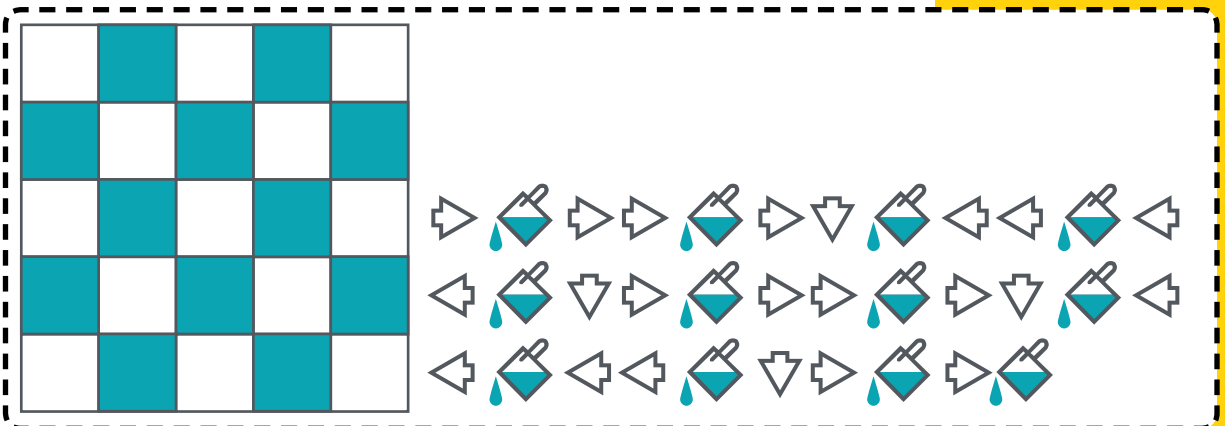


Zadania z labiryntami to doskonała okazja do bieżącego weryfikowania poziomu wykorzystywania przez uczniów umiejętności związanych z orientacją przestrzenną. Aktywności tego rodzaju można kontynuować z wykorzystaniem aplikacji, np.:

- **Kodable**
<https://goo.gl/tC8agF>
- **Labirynt, sekwencje**
<https://studio.code.org/s/course1/stage/4/puzzle/1>
- **Labirynt debugowanie**
<https://studio.code.org/s/course1/stage/5/puzzle/1>

Innym wariantem jest prosta zabawa w stemplowanie z błędami. Uczniowie, którzy mieli okazję ćwiczyć tego typu zadania (rozdział **B jak binarny system**), otrzymują ilustrację z zamalowanymi polami i kod, który zawiera błąd. Zadaniem dzieci jest wskazanie błędu i poprawienie kodu.

Przykładowy układ oraz błędny kod:



Tego typu zadania w wersji offline stanowią dobry wstęp ćwiczeń dotyczących debugowania z wykorzystaniem technologii, np. ze strony code.org

Ciekawym rozwiązaniem jest także przygotowanie projektu w Scratchu z ukrytymi błędami. Po prezentacji oczekiwanego efektu końcowego można zmienić kod tak, by wprowadzić do niego błąd. Zadaniem dzieci jest przetestowanie działania skryptu oraz wskazanie i naprawa błędu.

Znajdowanie i naprawa błędów w formie zabaw nie wyczerpują jednak zagadnienia. Warto poświęcić czas na rozmowę o tym, czego uczy nas debugowanie. Jakie umiejętności rozwija? W jakich sytuacjach życiowych mogą nam się przydać umiejętności ćwiczone w trakcie tej aktywności?

{ E JAK EKSTRA DODATEK

Na kanwie kilkuletniego doświadczenia grupy nauczycieli i edukatorów, działających w ogólnopolskim programie Mistrzowie Kodowania, przygotowany został swojego rodzaju niezbędnik. To zestaw materiałów, które sprawdziły się podczas zajęć z podstaw programowania offline.

W dużej mierze są to materiały, które często znajdują się w salach lekcyjnych lub świetlicach. Część można zastąpić tym, co akurat udało się nam zgromadzić. Niektóre „gadżety” łatwo wykonasz na zajęciach razem z dziećmi.

GADŻETY

W zabawach wprowadzających podstawy programowania mogą się przydać m.in.:

- Plastikowe piłki do zabawy, tzw. kulki do basenu w kilku kolorach.
- Różnokolorowe kubeczki plastikowe.
- Taśma malarska do wyklejania kratownicy.
- Taśma izolacyjna do wyznaczania labiryntów i tras.

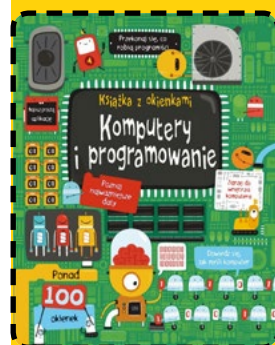
- Notes w bloczku z karteczkami białymi i kolorowymi.
- Karteczki typu post-it.
- Sześciennie drewniane kostki, klocki multiklocki lub klocki typu Lego.
- Małe figurki (np. Lego, z jajek niespodzianek, Duplo, Playmobil itd.).
- Kostki do gry (sześciennie, wielościenne, z kolorowymi ściankami itp.).
- Plastikowe nakrętki w różnych kolorach.
- Plastikowe packi na muchy.
- Sznurek, nitka, tasiemka.
- Figury do tangramów.
- Papier do origami.
- Instrumenty muzyczne itd.

Listę można stale rozszerzać, w zależności od naszych potrzeb i możliwości.

KSIĄŻKI

Aktywności z zakresu podstaw programowania można urozmaicać, wykorzystując książki dostępne na rynku wydawniczym, m.in.:

Komputery i programowanie. Książka z okienkami. Jedynie 16 grubych, kartonowych stron. Na każdej z nich moc sympatycznych minirobotów zaprasza do zaglądania do ukrytych okienek, gdzie znajdują się informacje na temat komputera (co ważne, rozumianego nie tylko jako klasyczne pudło z osobnym monitorem i klawiaturą) i tego, co kryje się w jego wnętrzu. W przystępny sposób podane są wiadomości o tym, czym jest programowanie oraz jak „myśli” komputer, jak realizuje nasze polecenia, czym są schematy blokowe, pętle i algorytm. Odkrywając języki programowania, najwięcej możemy dowiedzieć się o tym, co bliskie najmłodszym, czyli o Scratchu. Pod koniec wędrówki przez barwne karty trafimy na ciekawe informacje na temat list, danych, zmiennych, funkcji... czyli słów kilka o sprawnym pisa-



niu kodu. Na koniec jeszcze plansza o internecie oraz o historii komputerów.



Hello Ruby. Programowanie dla dzieci. To pierwsza z serii książek o przygodach sympatycznej i rezolutnej dziewczynki. Lektura przygód Ruby poprzez zabawę dostarcza podstawowych pojęć z programowania, technologii i komputerów.

Linda Liukas, tworząc opowieść o Ruby, wprowadza czytelników w takie zagadnienia jak tworzenie algorytmu (mycia zębów), rozpoznawanie wzoru (obrazków na tapecie), układanie pętli (figur tanecznych) czy też szukanie błędów w instrukcji (przygotowania kąpieli).

<http://www.helloruby.com/pl>



Oficjalna książka Scratch Jr. Tę książkę koniecznie trzeba mieć w swojej bibliotece, zwłaszcza jeśli pracujemy z aplikacją tabletową Scratch Jr. W bardzo prosty i przystępny sposób, czytelny nawet dla uczniów, zostały opisane podstawowe funkcje programu i jego interfejs. Co najważniejsze, książka zawiera gotowe pomysły do wykorzystania od zaraz. Tworząc miniprojekty, razem z uczniami poznasz środowisko aplikacji, a tym samym zgłębisz tajniki kodowania. Przyjazny kotek, piękna grafika oraz ciekawe rozwiązania to główne walory tej pozycji, które przyciągną niejednego małego programistę.



Scratch bez tajemnic. To minipodręcznik opowiadający, jak stworzyć proste gry w środowisku Scratch. Począwszy od pierwszych kroków, zapoznania się z interfejsem, funkcjami bloków i efektami, jakie daje ich wykorzystanie w danym momencie, poprzez proste projekty poznajemy tajniki tworzenia gier i animacji. W książce jest wiele ciekawych pomysłów na proste projekty, które możesz wykonać wraz z uczniami.

Kodowanie dla dzieci. Baw się i ucz ze Scratchem i Pythonem. Książka dedykowana najmłodszym programistom. Ciekawa grafika oraz projekty przyciągają oko. Już od pierwszych lekcji proponowane jest tworzenie swoich własnych bloków w Scratchu, co pokazuje nieco inny model pracy niż dotychczas. Oprócz ciekawych projektów znajdziemy słowniczek pojęć, propozycje innych rozwiązań programistycznych, takich jak Raspberry Pi czy Snap!

Programowanie dla dzieci. Obecnie na rynku wydawniczym dostępne są dwie części z tej serii. W pierwszej znajdują się propozycje prostych projektów w Scratchu i pomysły na minigry typu „Rozgnieć komara”, „Tor wyścigowy”, „Spróbuj mnie złapać” itd. Na uwagę zasługuje fakt, że w książce jest podana strona, z której można pobrać grafiki i wykorzystać je w projektach. Dzięki tym grafikom Scratch nabiera innej, trochę świeższej barwy. W drugiej części znajdziemy bardziej zaawansowane animacje ze Scratchem.

Moje roboty z serii Pizamorama to książka, do której można zajrzeć jako do ciekawostki. Na dwóch tuzinach stron ukryte są „mobilne” ilustracje, przedstawiające artystyczną wizję działających robotów, migających diod, poruszających się trybików.

Jak to możliwe, że rysunki się poruszają? Efekt uzyskujemy dzięki „prążkowanej” folii przykładanej do „prążkowanych” ilustracji i de-li-kat-nie, powoli przesuwanej na bok.

Zabawa tą książką pozwala nam poczuć, że pozornie magiczne rzeczy to często tylko świetny pomysł i dopracowana realizacja autora. Działanie wymaga skompletowania wszystkich elementów i właściwego ich użycia. Oraz... że czasem lepszy finalny efekt jest skutkiem spokojniejszego tempa pracy.



TECHNOLOGIA

Sztafetę tego metodnika rozpoczęło omówienie podstawowych treści z zakresu programowania oraz propozycje zabaw offline, czyli aktywności niewymagających użycia komputerów. Zbliżając się do finału tego opracowania, sięgamy nieco po technologię. W poszukiwaniu przydatnych materiałów i narzędzi warto bowiem korzystać z zasobów sieci. Wśród przydatnych stron można znaleźć m.in.:



wiki.mistrzowiekodowania.pl

Udostępnione bezpłatnie materiały przygotowane przy wsparciu firmy Samsung, w ramach programu Mistrzowie Kodowania; strona zawiera szereg scenariuszy oraz webinarów dotyczących programowania w środowisku Scratch, Scratch Jr, GreenFoot (Java), PocketCode, AppInventor oraz robotów Dash&Dot, Beebot, mBot.



zaprogramujprzyszlosc.edu.pl

Strona projektu realizowanego przez Fundację Orange i Stowarzyszenie Mistrzowie Kodowania. Organizacje prowadzą ją wspólnie w ramach 9 projektów regionalnych, które uzyskały dofinansowanie ze środków Programu Operacyjnego Polska Cyfrowa.



csunplugged.org

Zbiór bezpłatnych materiałów dydaktycznych, które uczą informatyki poprzez angażowanie gier i łamigłówek, przy wykorzystaniu kart, sznurków, kredek i ruchu.



oswajamyprogramowanie.edu.pl

Największą siłą internetu są ludzie. Nauczyciele, edukatorzy i pasjonaci dzielą się swoimi pomysłami i doświadczeniem, tworząc blogi. Jednym z takich miejsc jest blog prowadzony przez grupę nauczycieli i edukatorów, których połączył program Mistrzowie Kodowania.

Na blogu znajdziesz wiele pomysłów na zajęcia, praktycznych rozwiązań, gier i zabaw... oraz informacje o innych miejscach w internecie, warty odwiedzenia w poszukiwaniu materiałów na temat podstaw programowania w edukacji.

Zajęcia z podstaw programowania to doskonała okazja także dla nas, dorosłych, aby uczyć się i stale rozwijać. Technologia może urozmaicić zajęcia, wspierać pogłębianie przydatnych umiejętności – nie tylko tych ściśle związanych z programowaniem i nie tylko u dzieci. Niezależnie od wybranej aktywności, zabawy czy wykorzystywanej aplikacji, ważne jest, aby także samemu ją poznać. A to doskonała okazja do tego, by w trakcie przygotowywania się do zajęć również dorośli mogli dobrze się bawić, grając, testując i sprawdzając.

Dostępnych jest wiele aplikacji wspomagających wprowadzanie uczniów w świat programowania. Większość z tych programów wykorzystuje system kodowania przy pomocy kolorowych bloczków, strzałek lub innych prostych piktogramów. Każda z tych aplikacji ma własny system kodowania informacji, w zależności od tego, czy jest to kodowanie na płaszczyźnie czy w przestrzeni imitującej trójwymiar. Jedne mają charakter gier platformowych, inne prezentują proste labirynty do przejścia. Tego typu aplikacje łączą ich atrakcyjność. Można by rzec: „Dzieci to lubią!”. Dobrze wybrana i wprowadzona aplikacja to coś więcej niż gra, to tak zwane „apki na mózg”.

Zbliżając się do mety naszej sztafety, nie będziemy jednak opisywać szczegółowo programów i aplikacji. Nie poruszymy również kwestii organizacji pracy z technologią ani nie wspomnimy o działaniach w parach czy w grupach. Pominiemy także kwestię tego, jak uczyć i jak nie uczyć korzystania z aplikacji... Zapraszamy jednak do poszukiwania naszych kolejnych publikacji, w tym Narzędziownika, w którym poruszone zostaną m.in. powyższe tematy.

Można rzec, że takie marginalizowanie technologii w opracowaniu dotyczącym podstaw programowania jest pomyłką lub

co najmniej przekorą... Takie pozycjonowanie jest jednak celowym zamierzeniem.

Jesteśmy głęboko przekonane, że najciekawszy program nie zastąpi relacji z drugim człowiekiem. A współpracę w grupie, kreatywność, umiejętność radzenia sobie z trudnościami można z powodzeniem rozwijać w toku aktywności offline czy działań projektowych. Technologię traktujemy jako narzędzie, uzupełnienie szeroko rozumianej edukacji, nie zaś cel działań. Działania prowadzone w toku edukacji wczesnoszkolnej, sprzyjające rozwijaniu kompetencji związanych z programowaniem nie mają przecież służyć wykształceniu rzeszy przyszłych programistów. Istotniejsze jest wzmacnianie i wspieranie rozwoju wszystkich dzieci, także tych, które będą w przyszłości wykonywały inne zawody.

Kształt i formuła spotkań z programowaniem zależą od nas i naszych uczniów. Czy damy sobie i uczniom możliwość wyzwolenia kreatywności? Czy zapewnimy przestrzeń do samodzielnego i zespołowego odnajdywania rozwiązań? Czy otworzymy się na poszukiwania wypełnione wieloma próbami? Czy uznamy popełnianie błędów za wartościowe i niezbędne?

Mając ugruntowane podstawy teoretyczne, możemy bezpiecznie uruchomić pokłady naszej twórczości. Naszą pomysłowością możemy też zachęcić innych do innowacyjności. Sztafeta inspiracji to zapraszanie do kreatywności nauczycieli oraz uczniów.

Krytyczna lektura scenariuszy i propozycji zabaw, otwartość na ich modyfikowanie i dostosowywanie do potrzeb grupy są niezwykle istotne. Wiążą się z wcześniej wspomnianą kreatywnością i innowacyjnością, ale jednocześnie zapraszają do refleksji nad... czasem. Samodzielne i zespołowe odnajdywanie rozwiązań wymaga bowiem przestrzeni na zastanowienie, rozważenie, rozmowy, a także wielokrotne próby. To jednocześnie splata się z hasłem „błędy”. Warto z nimi się oswoić, można je polubić. Są przecież nieodłącznym elementem naszego działania i roz-

woju. Są wartościowe, ponieważ jeśli zostaną przeanalizowane, mają szansę stać się elementem naszej wspólnej nauki.

Dbający o dobrą atmosferę, budujący bezpieczną przestrzeń, wspierający wysiłki dzieci nauczyciel staje się swoistym przewodnikiem. Osobą zapalającą iskierki i podsycającą je do zwiększania swoich płomieni. Jednocześnie sam może być pochodnią płonąca dzięki zaangażowaniu, pomysłom i działaniu uczniów. Bądźmy takimi nauczycielami – tego sobie i Czytelnikom życzymy ;-)

{ F JAK FINAŁ

W naszym poradniku symboliczne przekroczenie linii mety zostało umieszczone w przedostatnim rozdziale. Ale gdy sztafeta kończy swój bieg, na zawodników czekają jeszcze owacje publiczności i medale. Nagroda za wysiłek i trud sportowców. Wprawdzie nie możemy rozdać medali, ale w finale przekazujemy jeszcze trzy nagrody.

RAZ, DWA, TRZY!

Zabawa uczy koncentracji, tworzenia sekwencji, a także powtórzeń. Zaproś uczniów do wspólnego odliczania do trzech.

To łatwe! – odpowiedzą zapewne uczniowie. Zaczynają liczyć: raz, dwa, trzy, raz, dwa, trzy, raz, dwa... Zapytaj, co będzie, gdy zaczniecie liczyć na zmianę. Rozpocznij liczenie. Dzieci szybko nabiorą wprawy, wtedy możesz wprowadzić dodatkowe elementy.

Zaproponuj, by zamiast mówić „raz”, dzieci klaskały w dłonie, klaszcząc naprzemiennie, jak w poprzednim odliczaniu. „Dwa” zastąpcie tupnięciem. Zaczynajcie pierwszą sekwencję: Twoje klaśnięcie, tupnięcie dzieci i Twoje „trzy”. Później drugą: uczniowie klaszczą, Ty tupiesz, uczniowie odliczają „trzy”... Czym zastąpisz słowo „trzy”? Czy uda się bezbłędnie odliczyć do trzech bez słów? Jakie inne sekwencje uda się stworzyć?

PRZEKAŻ WIADOMOŚĆ

Do aktywności potrzebne będą: kratownica, kolorowe kartki wielkości kraterki na kratownicy, koc, kartoniki z liczbami. W zabawie wezmą udział dwie drużyny, a celem ich działań będzie ułożenie symetrycznego rysunku z wykorzystaniem opcji kodowania i dekodowania wiadomości o położeniu elementów na kratownicy.

	A	B	C	D	E	E	D	C	B	A	
1											1
2											2
3											3
4											4
5											5
6											6
7											7
8											8
9											9
10											10

Kratownicę przedziel kocem na dwie równe części tak, aby dzieci nie widziały części planszy należącej do drugiej drużyny. Następnie, według powyższego wzoru, rozłóż liczby wzdłuż pionowego boku oraz litery alfabetu wzdłuż górnej krawędzi planszy.

Jedna z drużyn tworzy z kolorowych kartek prosty obrazek tak, aby druga drużyna widziała układu na macie. Następnie członkowie pierwszej drużyny przekazują informacje na temat położenia kartek, kodując informację (np. czerwony na B7). Po zakończeniu nadawania wszystkich wiadomości odsoń całość maty i sprawdź, czy obie części obrazka są symetryczne.

Jeśli wszystko się zgadza, zamień drużyny rolami, aby stworzyć kolejną pracę.

PIEKŁO-NIEBO

W trakcie zabawy dzieci stworzą w zespołach animację, losując duszki i sprawdzając działanie skryptów oraz bloczków w programie ScratchJr. Piekło-Niebo może być formą urozmaicenia wcześniejszych aktywności z użyciem aplikacji, jak również instruktażem działania nowych skryptów w aplikacji. Praca w parach uatrakcyjni rozgrywkę, a dodatkowo jest doskonałą okazją do rozwijania umiejętności komunikacji.

Zaproponuj dzieciom dobranie się w pary i ustalenie, kto rozpoczyna rozgrywkę. Rozdaj parom po jednej sześcienną kostkę do gry oraz po jednym tablecie z aplikacją ScratchJr. Przygotujcie plansze do zabawy, wycinając kartę Piekło-Niebo i składając według instrukcji [9]. Omów zasady gry:

- Wymyślcie w parze historyjkę, wykorzystując duszki i skrypty przepisane z zabawki do aplikacji.
- Pierwszy gracz rzuca kostką, a następnie porusza zabawkę tyle razy, ile wskazuje liczba oczek na kostce.
- Po zatrzymaniu ruchu zabawki drugi gracz wybiera jednego duszka spośród widocznych elementów.
- Wybrana postać zostaje dodana do projektu w aplikacji ScratchJr.
- Pierwszy gracz rozkłada zabawkę, umożliwiając sprawdzenie, jaki skrypt kryje się pod duszkiem.
- Drugi gracz przepisuje kod w aplikacji.
- Gracze zamieniają się rolami i powtarzają procedurę.

{ 9 } Dawne gry i zabawy
– *piekło/niebo*, online:
<https://youtu.be/s0rnLmz9uaU>
[dostęp 20.09.2018].

{ 10 } *Mobizen Screen Recorder – Record, Capture, Edit, online:* <http://bit.ly/1lO3uSO> [dostęp 20.09.2018].

Zabawę można przeprowadzić w wielu wariantach, np.:

- Możecie dodać maksymalnie 5 klocków ewentualnie jednego duszka.
- Ustal wspólnie z uczniami liczbę dodatkowych bloków wykorzystanych do stworzenia historyjki. Praca może być tworzona w kilku zespołach. Dzieci mogą przy pomocy skrzynki pocztowej przesłać plik z przygotowaną pierwszą częścią historii innej parze tak, aby kolejny zespół dokończył animację według własnego pomysłu. W danej animacji jest możliwe dodanie czterech scen, więc nad programem mogą pracować kolejno cztery zespoły.
- Zainstaluj na tabletach aplikację Mobizen **{10}**, aby nagrać efekt pracy. Udostępnij projekty na szkolnej stronie, kanale YouTube lub prześlij e-mailem do rodziców, którzy mają aplikację Scratch Jr na swoim tablecie.



Publikacja jest współfinansowana przez Unię Europejską ze środków Europejskiego Funduszu Rozwoju Regionalnego w ramach Programu Operacyjnego Polska Cyfrowa na lata 2014–2020 w obrębie projektów „Zaprogramuj przyszłość”.

