

W PRACOWNI KOMPETENCJI CYFROWYCH – ŚWIADOME UCZENIE SIĘ KODOWANIA I PROGRAMOWANIA



Wzorcowy materiał szkoleniowy w zakresie innowacyjnych rozwiązań organizacyjno-dydaktycznych dla nauczycieli i studentów studiów pedagogicznych

Edukacja informatyczna w klasach I-III szkoły podstawowej

Łukasz Dworski

Autor:

Łukasz Dworski

Recenzentka:

Natalia Walter

Wydawca:

Euro Innowacje sp. z o.o.

Publikacja została opracowana w ramach projektu pt. „Szkoła Ćwiczeń w Gminie Rawicz”, realizowanego w partnerstwie przez Gminę Rawicz (Beneficjent projektu) oraz Euro Innowacje sp. z o.o. (Partner projektu).

Projekt jest finansowany ze środków budżetu państwa oraz Unii Europejskiej, w ramach Europejskiego Funduszu Społecznego, Programu Operacyjnego Wiedza Edukacja Rozwój (PO WER), II Osi Priorytetowej „*Efektywne polityki publiczne dla rynku pracy, gospodarki i edukacji*”, Działania 2.10 „*Wysokiej jakości system oświaty*”.

Publikacja jest udostępniona na zasadach wolnej licencji Creative Commons – Uznanie autorstwa 3.0 Polska (CC BY 3.0 PL)



Fundusze Europejskie
Wiedza Edukacja Rozwój



**Rzeczpospolita
Polska**

Unia Europejska
Europejski Fundusz Społeczny



SPIS TREŚCI

WSTĘP	4
CEL PUBLIKACJI	6
1. Koncepcje edukacyjne	8
<i>1.1 Koncepcje edukacyjne w korelacji z podstawą programową</i>	<i>8</i>
2. Kodowanie a programowanie	10
<i>2.1 Różnice pomiędzy kodowaniem a programowaniem</i>	<i>11</i>
<i>2.2 Wpływ i rola programowania na rozwój myślenia komputacyjnego</i>	<i>13</i>
3. Uczenie się poprzez działanie	16
<i>3.1 Z komputerem i bez komputera</i>	<i>18</i>
<i>3.2 W labiryncie programowania</i>	<i>23</i>
<i>3.3 Mierniki efektów kształcenia</i>	<i>26</i>
4. Zaprogramowane klocki	29
PODSUMOWANIE	31
BIBLIOGRAFIA Z UWZGLĘDNIENIEM NETOGRAFII.....	32
WYKAZ ILUSTRACJI.....	35
WYKAZ TABEL	35



WSTĘP

Niniejsza publikacja powstała z myślą o nauczycielach szkół podstawowych I etapu edukacyjnego oraz studentach studiów pedagogicznych. Adresatem materiału szkoleniowego są nauczyciele edukacji wczesnoszkolnej, pedagodzy, psychologowie oraz inne osoby pracujące lub rozpoczynające pracę z dziećmi w wieku 6-9 lat. Kierowana jest ponadto do wszystkich nauczycieli dla których rozwijanie zarówno kompetencji cyfrowych u uczniów metodami praktycznego doświadczania, jak i praktycznych umiejętności wykorzystania narzędzi cyfrowych stanowią kluczowy element ich pracy lub chcą stawiać sobie nowe cele i wyzwania w oparciu o prezentowane rozwiązania. Projekt ten zakłada realizację zajęć dla uczniów klas 1-3 szkoły podstawowej z czterech modułów tematycznych:

- a) twórczości,
- b) technologii informacyjnej (algorytmika i programowanie),
- c) rozwijanie kompetencji z zakresu języka angielskiego,
- d) współpracy.

Wszystkie moduły są ze sobą połączone w taki sposób, by nabywaną wiedzę i umiejętności wzmacniać a realizowane ćwiczenia pokazywać i omawiać w różnych perspektywach. Szczegółowy opis celów i treści poszczególnych modułów umieszczony jest treści prezentowanych scenariuszy i ćwiczeń do realizacji przez uczniów.

Wiek wczesnoszkolny to istotny etap rozwoju dziecka. W tym czasie zachodzi wiele procesów, które warunkują przyszłe funkcjonowanie w obszarze poznawczym, fizycznym, emocjonalnym i społecznym. Najważniejsze zmiany dotyczą rozwoju psychomotorycznego, kompetencji społecznych, rozwoju pamięci, uwagi, myślenia oraz komunikacji, jak również poszerzania wiedzy o sobie i otaczającym świecie.

Biorąc to wszystko pod uwagę, przygotowałem propozycje dla uczniów, w którym dostosowuję zadania i treści do psychofizycznego rozwoju dziecka



poprzez stopniowanie trudności, ciekawy dobór tematów oraz narzędzi, metody polegające na poszukiwaniu, odkrywaniu, a także współpracy oraz modelu oceniania kształtującego wspierającego uczenie się.

Dlatego też celem realizacji programu jest nie tylko kształtowanie technicznych umiejętności obsługi różnorodnych narzędzi, ale wykorzystanie ich do poznawania świata, rozwijania kompetencji społecznych, poznawczych czy pielęgnowania zainteresowań.

Zgodnie z podstawą programową kształcenia ogólnego zadaniem szkoły jest łagodne wprowadzenie dziecka w świat wiedzy, przygotowanie do wykonywania obowiązków ucznia oraz wdrażanie do samorozwoju. Kształcenie ogólne w szkole podstawowej ma na celu m.in. wprowadzanie uczniów w świat współpracy, rozwijanie kompetencji takich jak kreatywność czy innowacyjność, rozwijanie umiejętności krytycznego i logicznego myślenia, rozumowania, rozbudzanie ciekawości poznawczej, kształtowanie umiejętności, które pozwalają w sposób bardziej dojrzały zrozumieć świat. Cele edukacji informatycznej dotyczą głównie:

- a) rozwijania umiejętności rozwiązywania problemów z wykorzystaniem komputera i innych urządzeń cyfrowych,
- b) rozwijania umiejętności programowania oraz tworzenia grafiki i tekstu,
- c) rozwijania umiejętności bezpiecznego i zgodnego z prawem korzystania z komputera i zasobów internetu,
- d) rozwijania kompetencji społecznych i umiejętności komunikowania się z wykorzystaniem technologii cyfrowej.

Zmiany zachodzące we współczesnym świecie powodują, że szkoła potrzebuje nowatorskich narzędzi edukacyjnych, form i metod pracy wykraczających poza utarty sposób myślenia. Wiąże się to z koniecznością opracowania programów nauczania i scenariuszy uwzględniających twórcze podejście do edukacji, pozwalających na indywidualizację procesu nauczania i wszechstronny rozwój umiejętności uczniów.



CEL PUBLIKACJI

Celem niniejszego opracowania jest wspieranie nauczycieli poprzez przedstawienie różnych aspektów rozwijania i wspierania kompetencji kluczowych u uczniów ze szczególnym uwzględnieniem innowacyjnych metod nauczania. Umożliwienie swoim uczniom podwyższenia poziomu ich wiedzy i umiejętności poprzez pomoc w procesie uczenia się, zmierzającym do obiektywnego postrzegania rzeczywistości i kreatywnego myślenia i umiejętności rozwiązywania problemów. Rozwój tych umiejętności jest niezbędny do efektywnego funkcjonowania we współczesnym społeczeństwie – do dostępu do satysfakcjonującej pracy i spełnienia w życiu osobistym.

Nowa podstawa programowa zawiera szereg wytycznych, do których należy między innymi kształtowanie i rozwijanie kompetencji kluczowych. Jest to połączenie wiedzy, umiejętności i postaw uważanych za niezbędne dla potrzeb samorealizacji i rozwoju osobistego, aktywnego obywatelstwa, integracji społecznej oraz zatrudnienia.

Kompetencje kluczowe można zdefiniować w bardzo prosty sposób jako połączenie wiedzy, umiejętności i postaw odpowiednich do odnalezienia się w napotkanej na swej drodze sytuacji uczenia się.

Kompetencje kluczowe to:

- 1) kompetencje w zakresie rozumienia i tworzenia informacji;
- 2) kompetencje w zakresie wielojęzyczności
- 3) kompetencje matematyczne oraz kompetencje w zakresie nauk przyrodniczych, technologii i inżynierii
- 4) kompetencje cyfrowe

Program ma na celu:

1. Poznanie nieformalnego znaczenia wybranych pojęć związanych z informatyką, często w formie zabawy:
 - a) liniowa kolejność, powtarzanie czynności,
 - b) instrukcja, sekwencje instrukcji,



- c) algorytm (plan działania).
2. Początek myślenia algorytmicznego, wspomagany wizualizacją lub symulacją działań algorytmicznych – uczniowie stawiają w ten sposób pierwsze kroki w wizualnym języku programowania.
3. Proste programy sterujące robotem lub inną istotą na ekranie.
4. Posługiwanie się komputerem i aplikacjami użytkowymi – opracowywanie kompozycji graficznych i dokumentów tekstowych.
5. Korzystanie bezpiecznie i zgodnie z prawem ze wskazanych zasobów w internecie.

Niniejsza publikacja jest więc sumą doświadczeń wynikających z podejmowanych przez autora działań w zakresie wspierania procesów edukacyjnych w szkołach, zwłaszcza w zakresie rozwijania kompetencji kluczowych uczniów i uczennic.



1. Koncepcje edukacyjne

Rozwijanie kompetencji informatycznych dzieci i młodzieży jest jednym z priorytetowych kierunków realizacji polityki oświatowej państwa. Informatyka jako dziedzina nauki (*ang. computer science*) – powiązana z jej praktyczną stroną wywodzącą się z dynamicznie rozwijających się technologii, wspiera i integruje się ze wszystkimi dziedzinami życia oraz wyposaża je w podstawowe metody i narzędzia. Bardzo ważne jest świadome i bezpieczne korzystanie z nich oraz umiejętność rozwiązywania problemów z różnych dziedzin z ich zastosowaniem. Dla współczesnych dzieci i młodzieży naturalne jest środowisko nasycone nowoczesną technologią – komputery, smartfony, tablety towarzyszą im od najmłodszych lat. W zakresie wykorzystania nowości technicznych i w pracy z dostępnymi aplikacjami uczniowie swoimi umiejętnościami często przewyższają dorosłych – nauczycieli.

W klasach 1-3 szkół podstawowych wyzwaniem mającym wpływ na aktywność cyfrową najmłodszych jest respektowanie zasad ergonomii, ich aktywności edukacyjnej i zabawy z wykorzystaniem nowych technologii. Zazwyczaj jednak ich doświadczenia nie są związane z działaniami edukacyjnymi.

1.1 Koncepcje edukacyjne w korelacji z podstawą programową

Celem zmian w edukacji informatycznej jest przekazanie uczniom wartości ponadczasowych, jakie niesie technologia – wyphywających z informatyki jako nauki i jej uzasadnionych zastosowań w innych dziedzinach, przedmiotach. Realizacja tego wyzwania wiąże się z nowym podejściem metodycznym zakładającym wykorzystanie technologii tylko w uzasadnionych przypadkach, gdy jej obecność niesie ze sobą wzmocnienie lub podniesienie efektów kształcenia, oraz zwięźczenie rozpatrywania różnorodnych problemów w postaci zaprogramowania ich rozwiązania. Przy czym zaprogramowanie rozwiązania nie wiąże się jedynie z napisaniem programu, ale może mieć



dowolną formę prowadzącą do uzyskania rozwiązania z pomocą technologii i dostępnych aplikacji.

Konieczność zmian w kształceniu informatycznym szybko dostrzegły niektóre kraje, wprowadzając nowe podstawy programowe kładące nacisk na powszechną naukę myślenia algorytmicznego i programowania oraz kształcenie umiejętności ich stosowania do rozwiązywania problemów z różnych dziedzin – umiejętność ta nazywana jest myśleniem komputacyjnym (*ang. computational thinking*).

W nowej podstawie programowej opracowanej przez Zespół MEN (Ministerstwo Edukacji Narodowej) ds. podstawy programowej z informatyki nastąpiła zmiana organizacyjna polegająca na ujednoczeniu nomenklatury przedmiotu przez wprowadzenie w edukacji wczesnoszkolnej nazwy „edukacja informatyczna” zaś nazwy „informatyka” dla zajęć prowadzonych na kolejnych etapach edukacyjnych.

Nowa metodyka pracy przyjęta w momencie wprowadzenia programowania ma sprzyjać prawidłowemu rozwojowi dziecka od najmłodszych lat i w tym kontekście wpływać na poszerzenie pola jego doświadczeń. Pozwala ona na realizację zapisów podstawy dotyczących osiągnięć w zakresie rozumienia, analizowania i rozwiązywania problemów przez wykorzystanie przestrzeni klasy do zabaw – również ruchowych.

Na etapie nauczania wczesnoszkolnego dziecko podejmuje pierwsze próby wizualnego programowania. Formułuje instrukcje lub sekwencje instrukcji dla wybranego obiektu, obserwując jednocześnie efekty swojej pracy na ekranie. Poprawia rozwiązania, aż do osiągnięcia wyznaczonego celu, tworząc w ten sposób pierwsze programy. Poczynając od sytuacji najprostszych, przechodzi do coraz trudniejszych, jednocześnie doskonaląc koordynację ruchową poprzez wykorzystanie myszy czy ekranu dotykowego, poznaje też funkcjonalności klawiatury. Podstawa programowa nie narzuca konkretnego rodzaju wizualnego języka programowania wykorzystywanego podczas lekcji, pozostawia ten wybór nauczycielowi.



2. Kodowanie a programowanie

Naukę kodowania dzieci rozpoczynają już w wieku przedszkolnym lub wczesnoszkolnym. Na początku odbywa się ona w formie zabawy, jednak już wtedy uczy logicznego, kreatywnego i analitycznego myślenia. Dzieci poznają wówczas także kompetencje w postaci sprawczości, którą wykorzystuje się w ciągu całego życia. Dla wszystkich osób, które chcą rozpocząć lub kontynuować przygodę z kodowaniem powstała coroczna, europejska inicjatywa społeczna CodeWeek, w której Polska jest liderem.

Kodowanie opiera się na logice. Kod ma swoją własną strukturę, a poszczególne elementy są tutaj powiązane ze sobą. Oznacza to, że dobry programista powinien błyskawicznie dostrzegać i rozumieć zależności, aby napisać kod, który rzeczywiście będzie dobrze działał. Analityczne myślenie potrzebne jest, aby móc zrozumieć struktury logiczne oraz składnię poszczególnych języków. Dzięki temu stworzony kod nie będzie wymagał wielu poprawek, a program czy aplikacja będą dobrze działać. Nauka kodowania dla dzieci bardzo często skupia się właśnie na ćwiczeniu logiki i analitycznego powiązywania faktów.

Jedną z ciekawszych form wykorzystania narzędzi komputerowych jest nauka programowania. Umiejętność ta zaczyna być obecnie traktowana na równi z umiejętnością pisania oraz czytania (Wing Konser, 2012). Nabyte sprawności wspierają procesy myślowe oraz pobudzają dalszy rozwój uczniów.

Formułowanie poleceń, konstruowanie algorytmów oraz przewidywanie rezultatów podjętych działań ćwiczy umiejętność logicznego myślenia, rozwiązywania problemów, kreatywnego działania. Nauka programowania wspomaga również kształtowanie wytrwałości w dążeniu do celu. Rozwija zainteresowanie matematyką, rozumienie otaczającego świata, jak też interpretowanie zjawisk przyrodniczych. Filiz Kalelio lu oraz Yasemin Gülbahar na podstawie zrealizowanych badań odnotowali, że nauka programowania podnosi samoocenę uczniów w kontekście umiejętności pokonywania trudności i rozwiązywania zadań.



Kodowanie wykorzystano m.in. w alfabecie Braille'a czy kodzie Morse'a. Obecnie kody towarzyszą nam na każdym kroku, np. kody kreskowe na cenach/metkach, kody korekcyjne na biletach, w książkach i podręcznikach, kody na kartach płatniczych, podarunkowych czy rabatowych. To tylko przykłady, gdyż nie sposób wymienić wszystkich możliwych sposobów wykorzystania kodów we współczesnym świecie. W edukacji także pojawiło się kodowanie, które zostało wpisane do wymagań szczegółowych w nowej podstawie programowej (rozporządzenie z 14 lutego 2017 r.). Myślenie komputacyjne poniekąd zastąpiło dawne myślenie matematyczne, poszerzając tym samym zakres rozwijanych umiejętności i postaw oraz obszar działania.

Kodowanie w szkole bardzo często kojarzone jest z programowaniem, a co za tym idzie – z komputerem, z nowoczesną technologią informacyjną. Proces kodowania nie musi jednak oznaczać programowania z wykorzystaniem technologii IT (technologia informacyjna). Nie musi też ograniczać się do informatyki czy robotyki. Wystarczy, że w procesie przetwarzania informacji pojawia się myślenie logiczne lub jego komponenty, które składają się do zastąpienia, zamiany, odszukania, skojarzenia, przyporządkowania, klasyfikowania, uogólniania, zapamiętania czy zastosowania/użycia inaczej, a już mówimy o kodowaniu lub programowaniu offline.

2.1 Różnice pomiędzy kodowaniem a programowaniem

W ostatnich latach w Polsce, jak i na całym świecie zaczyna być dotkliwie odczuwalny brak dobrych programistów. Niektóre państwa, uczelnie czy inne instytucje próbują zachęcać ludzi do nauki programowania. Rodzą się projekty ministrów edukacji (Wielkiej Brytanii, Estonii), uniwersytetów (MIT, Carnegie Melon), czy firm prywatnych (Google, Microsoft, Lego) itp. Niestety, zamiast nauki programowania chodzi w nich w większości o naukę kodowania – kodowania dla komputerów. Ta metoda w ciągu ostatnich 20 lat nie doprowadziła do wykształcenia wystarczającej liczby programistów. Dlaczego?



Dlatego, że potrzebujemy programistów, a nie „koderów”. Różnica między programistą a „koderem” jest taka, jak między pisarzem i tłumaczem.

Programowanie to działanie, którego wynikiem jest program. Dlatego też programem nie jest wyłącznie program komputerowy. Program to prostu przepis na coś, czyli kolejności poleceń, które należy wykonać, aby osiągnąć zamierzony efekt. Różnica między przepisem i programem polega na tym, że program jest już bardzo szczegółowym (doprecyzowanym) przepisem, w którym są uwzględnione wszystkie przewidziane, jak i nieprzewidziane sytuacje. Instrukcja warunkowa to po prostu rozgałęzienie w działaniu programu (ang. if = jeżeli). W zależności od tego, czy warunek zawarty w instrukcji jest prawdziwy lub fałszywy, wykonane zostają inne instrukcje.

Poniższy przykład – przepis na przygotowanie kawy:

1. *Nalej wodę do czajnika.*
2. *Wstaw czajnik na zapalony gaz.*
3. *Zagotuj wodę.*
4. *Zalej wrzątkiem kawę w filiżance.*

Program (w dużym uproszczeniu) wyglądałby tak:

1. *Weź czajnik.*
2. *Odkręć kran. Czy woda leci?*
 - a) *Jeżeli tak, to nalej wody do czajnika.*
 - b) *Jeżeli nie, to zakręć kran i idź sprawdzić czy nie jest zakręcony główny zawór. Jeżeli nie, to zadzwoń do wodociągów i zgłoś awarię. Poczekaj aż zacznie lecieć woda, albo zdobądź wodę z innego źródła (sklep, studnia itp.).*
4. *Wstaw czajnik na gaz i zapal gaz. Czy jest gaz?*

i tak dalej.

Jak widać z powyższego, prostego przykładu, tworzenie programu (programowanie) nie jest tylko tworzeniem przepisu (który też jest algorytmem), ale chodzi w nim już o rozwiązywanie problemów (przewidzianych i nieprzewidzianych).



Możemy więc zdefiniować, że programowanie = rozwiązywanie problemów.

Programistą jest osoba rozwiązująca problemy (np. kierownik, prawnik, programista komputerów). Kodowaniem nazywamy zapis rozwiązanego problemu do konkretnego języka lub innej formy, w zależności od wykonawcy i danej sytuacji.

W związku z powyższym uważam, że błędem jest prowadzenie przez nauczycieli zajęć tylko z obszaru kodowania, bez nauki umiejętności utożsamianych z programowaniem. Traktowanie kodowania tylko jako etapu przygotowującego do następnego poziomu jakim jest nauka programowania zarezerwowana dla informatyków i programistów.

Takie podejście w nauczaniu uczy dzieci wykorzystania i powielania schematów oraz szablonów prowadzących nas do jedynego słusznego rozwiązania zadania. Kodowanie w korelacji z programowaniem stawia przed uczniami możliwości rzeczywistego rozwiązywania problemów i zachęca do poszukania niestandardowych rozwiązań.

2.2 Wpływ i rola programowania na rozwój myślenia komputacyjnego

Intensywny rozwój nowych zastosowań technologii informacyjno-komunikacyjnych i ich wpływ na funkcjonowanie współczesnego człowieka spowodowały, iż z biegiem czasu dotychczasowe rozumienie kompetencji informatycznych musiało ulec ewolucji. Wielość i zakres zastosowań spowodowały zmiany w stylach i strategiach działania współczesnego człowieka. Szczególnie jest to widoczne u pokolenia millenialsów, dla których narzędzia i środki technologii informacyjnej stanowią naturalne środowisko życia. Wraz z postępem cywilizacyjnym oczywiste zatem stało się poszerzenie kompetencji informatycznych o myślenie komputacyjne (computational thinking), pozwalające na rozwiązywanie problemów z różnych dziedzin przy użyciu metod komputerowych. Zdaniem Sysło rozwijanie myślenia algorytmicznego przestało już wystarczać. Znaleźliśmy się w momencie, w którym



należy rozwijać myślenie „obejmujące myślenie na wielu poziomach abstrakcji nierozzerwalnie związane z informatyką i jej metodami”.

Myślenie komputacyjne to proces znajdowania rozwiązań do skomplikowanych otwartych problemów. Niekiedy w myśleniu komputacyjnym podkreśla się rolę jaką odgrywają komputery (wówczas myślenie komputacyjne jest rodzajem porządkowania danych i formułowania problemów oraz rozwiązań w taki sposób, by były zrozumiałe również dla komputerów) i dodaje etapy związane z tłumaczeniem rozwiązania na język zrozumiały dla komputera (programowanie, w tym kodowanie), a także znajdowania innych zastosowań danego algorytmu/rozwiązania (adaptacja do innych kontekstów). Choć myślenie komputacyjne brzmi abstrakcyjnie, jego elementy włączane są do programów związanych z nauką kodowania. Myślenie komputacyjne jest niezbędne nie tylko do tworzenia programów komputerowych i zrozumienia języków programowania, ale może być również wykorzystywane do wspierania rozwiązywania problemów we wszystkich dyscyplinach i sektorach. Rozwijanie myślenia komputacyjnego ma miejsce, gdy problemy są formułowane tak, aby można je było rozwiązać metodami komputerowymi za pomocą narzędzi informatycznych:

- rozwiązanie problemu wymaga określenia związków logicznych pomiędzy danymi i wnioskowania na podstawie analizy wzajemnych powiązań; problem można przedstawić w postaci algorytmu poprzez projektowanie i analizę możliwych rozwiązań,
- możliwe jest tworzenie modeli sytuacji i symulacji zjawisk oraz procesów za pomocą narzędzi i metod komputerowych,
- uzyskuje się możliwie najbardziej efektywne rozwiązanie przy najlepszym wykorzystaniu narzędzi i środków informatycznych,
- stosuje się uzyskane rozwiązanie do rozwiązywania innych problemów. Przy stosowaniu metod komputerowych do rozwiązywania problemów zaleca się, aby uzyskane rozwiązania:



- zostały przedstawione w sposób czytelny i przejrzysty (komunikatywny),
- zachowały poprawność merytoryczną, czyli były realizowane w zgodzie z koniecznymi założeniami,
- wykazywały wysoką efektywność uwzględniającą wykorzystanie zasobów informatycznych.

W rozwijaniu myślenia komputacyjnego chodzi zatem o nauczanie programowania jako sposobu na rozwiązywanie problemów za pomocą współczesnych narzędzi i metod informatycznych. Narzędzie stanowi tutaj środek do realizacji rozwiązania problemu. Praca koncepcyjna jako element pracy intelektualnej wyprzedza wybór narzędzia i realizację rozwiązania przy zachowaniu wszystkich elementów metody projektu.

Istotne w myśleniu komputacyjnym są również postawy i umiejętności, chociażby takie, jak: eksperymentowanie, wyobrażanie, dążenie do celu, współpracowanie, klasyfikowanie, wyszukiwanie, uogólnianie, analizowanie i kodowanie, bez których rozwiązywanie problemów jest niemożliwe. Podobnie jak myślenie komputacyjne, programowanie coraz częściej jest definiowane jako etapowy proces obejmujący myślenie na wielu poziomach abstrakcji z wykorzystaniem narzędzi TIK (technologie informacyjno-komunikacyjne) w rozwiązywaniu problemów z różnych dziedzin życia (Podstawa programowa 2017: 27). W takim ujęciu programowanie jest bliższe definicji myślenia komputacyjnego, natomiast kodowanie lub programowanie offline (bez komputera) zdaje się procesem wstępnym, przygotowującym do programowania właściwego, a jednocześnie niezbędnym dla prawidłowego rozwoju procesów poznawczych u dziecka w wieku przedszkolnym i wczesnoszkolnym.



3. Uczenie się poprzez działanie

W edukacji praktyczne metody nauczania - uczenia się na poziomie wczesnej edukacji, polegające na samodzielnej realizacji przez uczniów zadań stanowią najefektywniejszą formę nauki przez doświadczenie oraz wzajemne uczenie się, które mają miejsce w sytuacji realnego działania uczniów. Projekty edukacyjne mają długą historię. Ich początków można szukać w koncepcji *learning by doing* (nauki przez działanie) sformułowanej przez Johna Deweya jeszcze pod koniec XIX wieku, w myśl której rolą nauczyciela nie jest przekazywanie konkretnej wiedzy, a towarzyszenie uczniowi w zdobywaniu określonych doświadczeń. Założenie, że podstawą nauki powinno być doświadczenie i oparte na zainteresowaniu uczniów rozwiązywanie konkretnych wyzwań czy problemów, pojawiało się w większości nurtów pedagogiki progresywnej, a pod koniec XX wieku stało się coraz szerszej wprowadzaną metodą w głównym nurcie edukacji – pod nazwą metody projektu (*project-based learning*).

Jeżeli wiedza człowieka tworzona jest w toku indywidualnego nadawania znaczeń i kształtowana jest poprzez doświadczenia jednostki, to zmianie ulega cała filozofia myślenia o procesie kształcenia. Uczenie bowiem dotyczy nie tyle przyswajania przez dziecko wiedzy, ile odnosi się do rozwoju jego jako człowieka. Uczenie się w tej perspektywie będzie dotyczyć każdego doświadczenia dziecka, które wpływa na kształtowanie jego jako człowieka i jego sposobu rozumienia rzeczy wistości społeczno-kulturowej, w której funkcjonuje. W związku z tym doświadczenie jest czynnikiem niezbędnym w procesie uczenia się. Te założenie wysunięte przez konstruktywistów znajduje odzwierciedlenie w neuronauce, która podkreśla, że eksploracja jest podstawą procesu uczenia się. Dziecko „w wyniku powtarzania doświadczeń [...] kumuluje i koduje wielką liczbę informacji. Na podstawie tych danych, [...] ustala prawdopodobieństwo, dzięki czemu [...] potrafi nieświadomie przewidzieć, czy do takiego, czy innego zdarzenia społecznego, lingwistycznego bądź fizycznego dojdzie, czy nie”



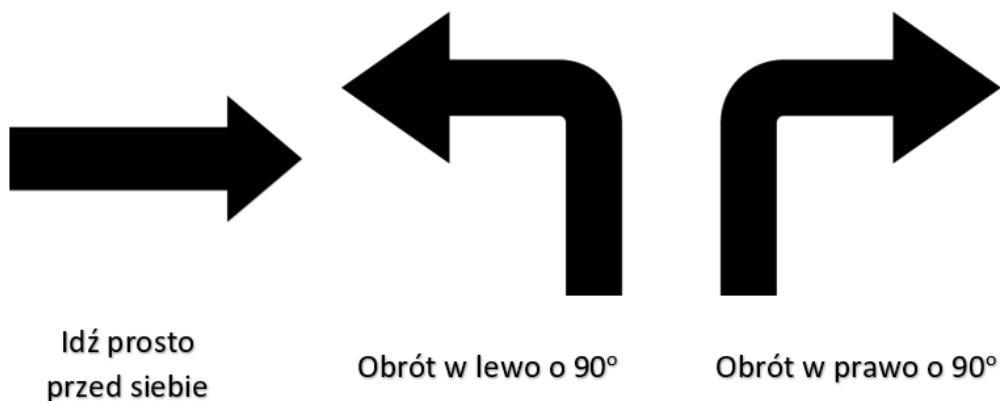
(Alvarez, 2016, s. 67). Innymi słowy, dziecko nieustannie konfrontuje swoją wiedzę z aktualnymi doświadczeniami i na ich podstawie dokonuje aktualizacji swojej wiedzy.

Wykorzystanie nowoczesnych środków dydaktycznych do nauki kodowania i programowania na zajęciach lekcyjnych podnosi ich walory dydaktyczne i atrakcyjność. Duży wybór dostępnych w internecie narzędzi i sposobów do ich samodzielnego wykonania ułatwia pracę nauczycielowi i daje możliwość zorganizowania różnorodnych zajęć dopasowanych do wieku i możliwości rozwojowych dzieci. Zajęcia takie pomogą w bezpieczny, odpowiedzialny sposób wprowadzić uczniów w cyfrowy świat. Pracując z uczniami najmłodszymi, w klasach 1-3 warto zajęcia z wykorzystaniem sprzętu komputerowego poprzedzić częścią wprowadzającą, do której przeprowadzenie nie będą potrzebne ani tablety, ani komputery. Podstawowe pojęcia dotyczące kodowania, takie jak: komenda, skrypt, pętla, instrukcja warunkowa, debugowanie może nauczyciel wprowadzić bez użycia sprzętu komputerowego, proponując uczniom zabawy offlinowe, uczące logicznego, algorytmicznego myślenia, jak również pracy zespołowej. Wykorzystując matę edukacyjną można stworzyć środowisko zbliżone do interfejsu aplikacji. Uczniowie będą układać skrypty w oparciu o powszechnie rozpoznawalne symbole. Na macie uczą się, co stanie się w momencie ułożenia strzałki w lewo, a co kiedy ułożą ją prosto. Praca z klockami konstrukcyjnymi oraz w aplikacji będzie kontynuacją zajęć na macie, utwaleniem pewnych umiejętności. Wszystkie wymienione propozycje dają nauczycielowi możliwość dopasowania stopnia trudności zajęć do możliwości dzieci.



3.1 Z komputerem i bez komputera

Orientacja w schemacie własnego ciała, orientacja przestrzenna, poprawne określanie stron świata i kierunków będą ważnym elementem zajęć przeprowadzonych według tego scenariusza. Sposób poruszania, jak również wykonywane gesty i czynności możemy przekazywać sobie, nie tylko za pomocą języka mówionego, czy pisanego, ale również stworzonego za pomocą symboli graficznych. Zajęcia "Labirynt możliwości" rozpoczynają się zabawami ruchowo - tanecznymi, które mają na celu zapoznanie uczniów ze znaczeniem poszczególnych plansz ruchu. Uczniowie wykorzystują nabytą wiedzę, do utworzenia skryptu, najpierw na macie edukacyjnej, następnie podczas zabaw klockami oraz w aplikacji internetowej.



Ilustracja 1. Strzałki wraz z opisem do kodowania kierunku ruchu

Źródło: opracowanie własne



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

Ilustracja 2. Projekt maty edukacyjnej

Źródło: opracowanie własne

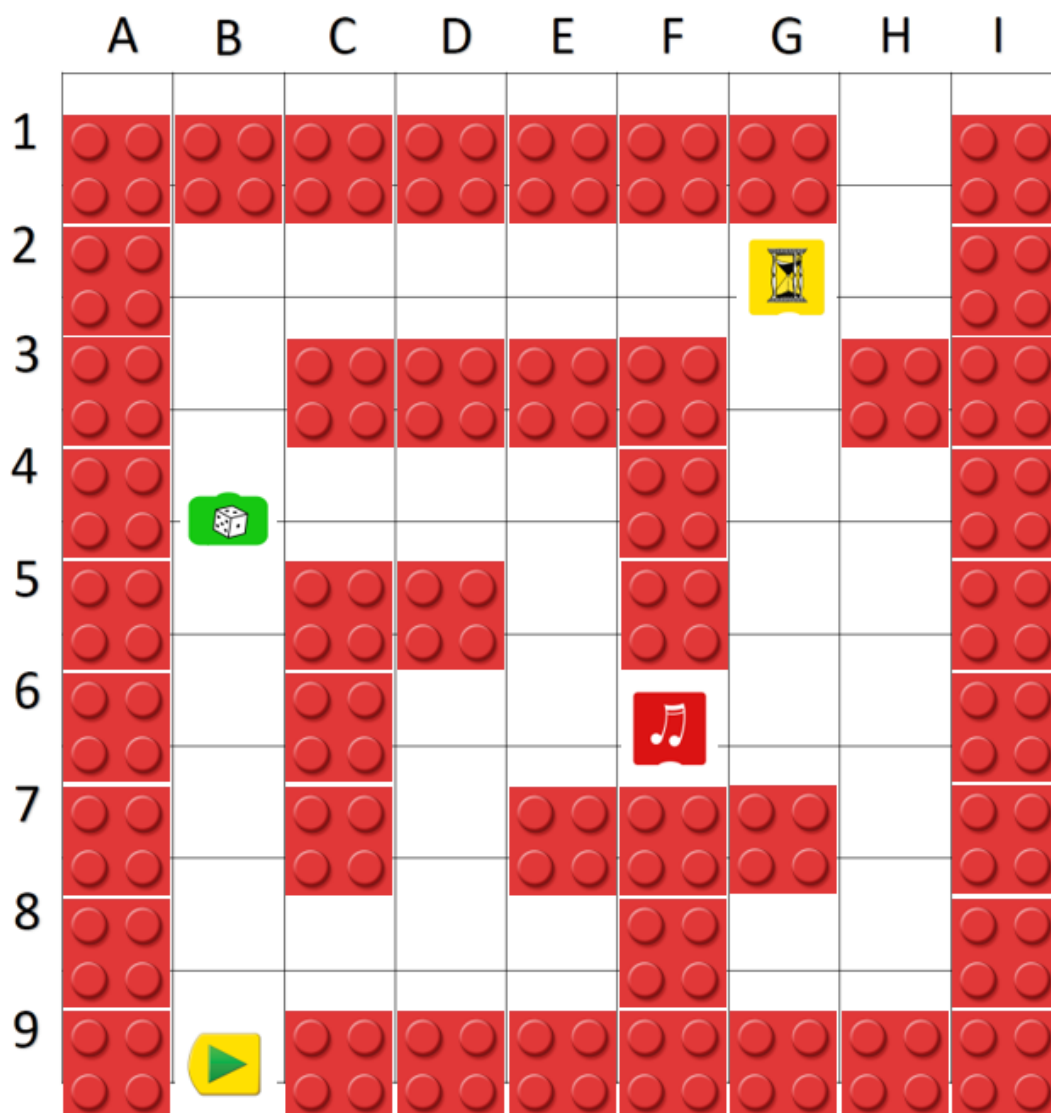
Cele ogólne:

- Nauka umiejętności budowania sekwencji skryptów pozwalających na precyzyjne poruszanie postaci.
- Ćwiczenie koncentracji uwagi,
- Rozwijanie logicznego algorytmicznego myślenia.



Cele szczegółowe:

- Uczeń zna symbolikę klocków ruchu i prawidłowo ją opisuje
- Potrafi poruszyć się zgodnie z kodem ułożonym z plansz ruchu,
- Chętnie pracuje w małych zespołach
- Używa funkcji powtórzeń
- Tworzy różnorodne rozwiązania



Ilustracja 3. Przykładowy labirynt ułożony na macie edukacyjnej

Źródło: opracowanie własne



Środki dydaktyczne:

Mata edukacyjna, plansze w różnych kolorach, plansze ze strzałkami ruchu, plansze z cyframi, kafelki z obrazkami, karta pracy.

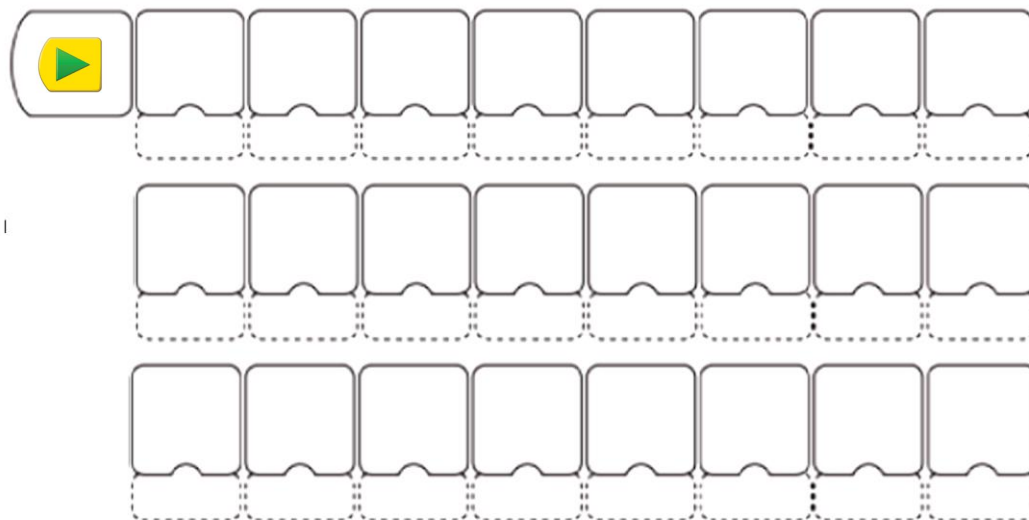
Przygotowanie do zajęć:

Zadania nauczyciela:

- Nauczyciel rozkłada matę edukacyjną oraz układa na niej kolorowe plansze tworząc układ korytarzy labiryntu
- Nauczyciel gromadzi w jednym miejscu wszystkie potrzebne do zajęć materiały, następnie prosi uczniów o podzielenie się w pary. Zespoły siadają nokoło maty.
- Nauczyciel wyjaśnia dzieciom cel zajęć, oraz to w jaki sposób będą przebiegały.

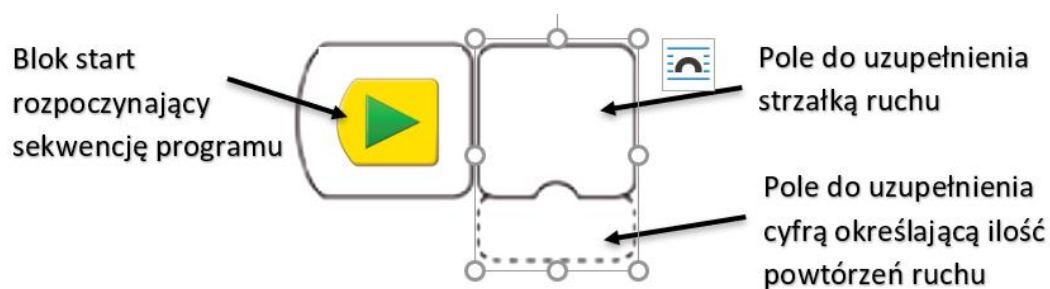
Zadania uczniów:

- Uczniowie dzielą się na 2-osobowe zespoły, zajmują miejsca wokół maty. Każdy zespół otrzymuje kartę pracy z pustymi polami do naniesienia właściwej sekwencji ruchu. W przypadku uczniów klasy trzeciej, karta pracy może zostać zastąpiona kartą do zapisania współrzędnych kratki, po której będzie poruszać się postać.
- Wśród kolorowych kwadratów w labiryncie znajdują się również obrazki, pod którymi mogą zostać ukryte różne zadania pogrupowane tematycznie (w zależności od wieku i możliwości rozwojowych dzieci, zadania te powinny być dostosowane).



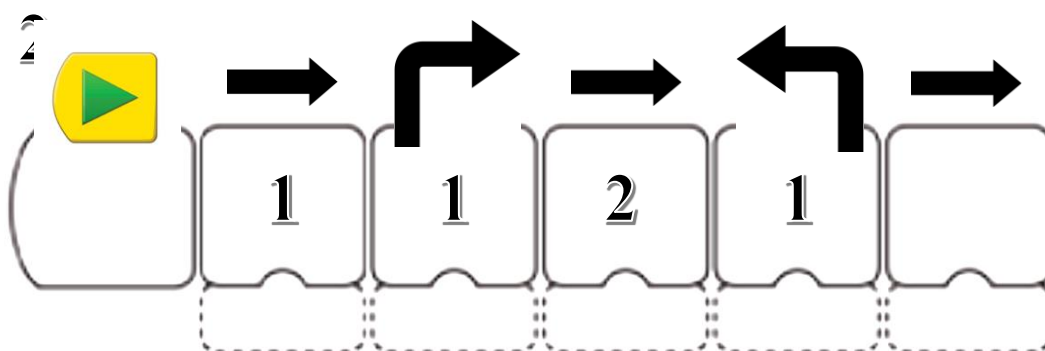
Ilustracja 4. Karta pracy dla uczniów

Źródło: opracowanie własne



Ilustracja 5. Wyjaśnienie pól karty pracy do uzupełnienia przez uczniów

Źródło: opracowanie własne



Ilustracja 6. Fragment przykładowego programu

Źródło: opracowanie własne

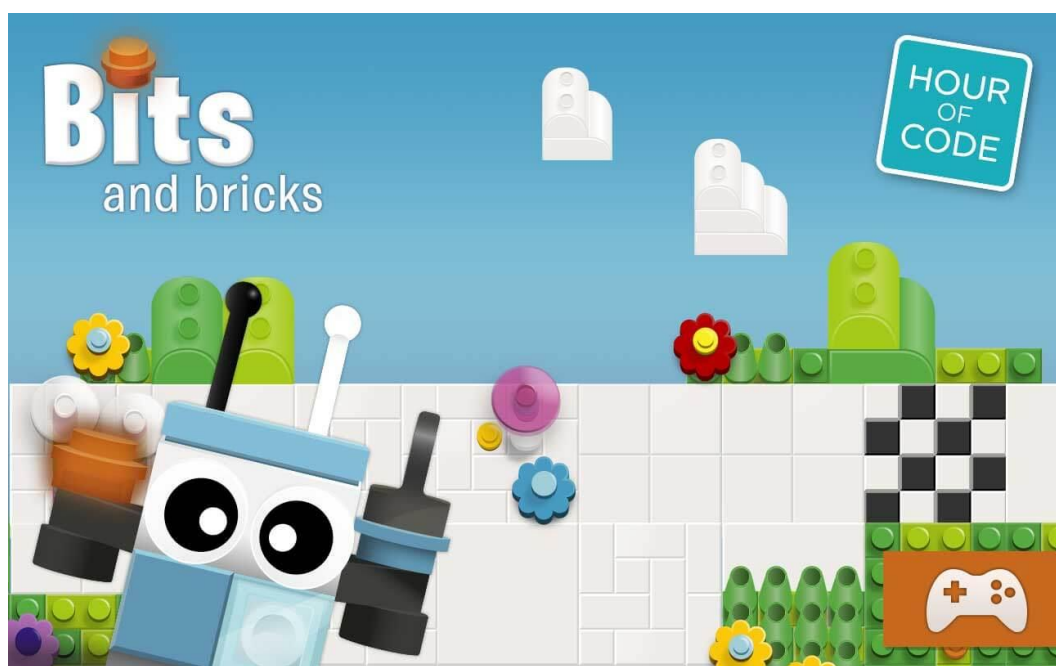


3.2 W labiryncie programowania

Ciekawą propozycją online do nauki programowania jest gra stworzona przez firmę LEGO i udostępniona na stronie firmy za darmo. Jest ona dostępna w wielu wersjach językowych – zarówno w języku angielskim jak również po polsku. Umiejętności nabyte podczas pracy ze scenariuszem offline z matą edukacyjną mogą być utrwalone i rozwinięte z wykorzystaniem bezpłatnej aplikacji internetowej:

[link do lego.com](https://www.lego.com)

Korzystając z programu możemy wymienić szereg rozwijanych umiejętności przydatnych w procesie nauczania programowania takich jak: wdrażanie uczniów do kodowania z wykorzystaniem narzędzi cyfrowych, nauka pisania kodu, współtworzenie algorytmów w celu dotarcia bohatera do wskazanego miejsca, stopniowanie trudności, praca w grupach i z całą klasą. Forma realizacji zadań pozwala nauczycielom na wykorzystanie zasobów pracowni komputerowej lub prywatnego sprzętu uczniów w domu (w sytuacji zdalnego nauczania wykonywania zadań w trakcie wideokonferencji).

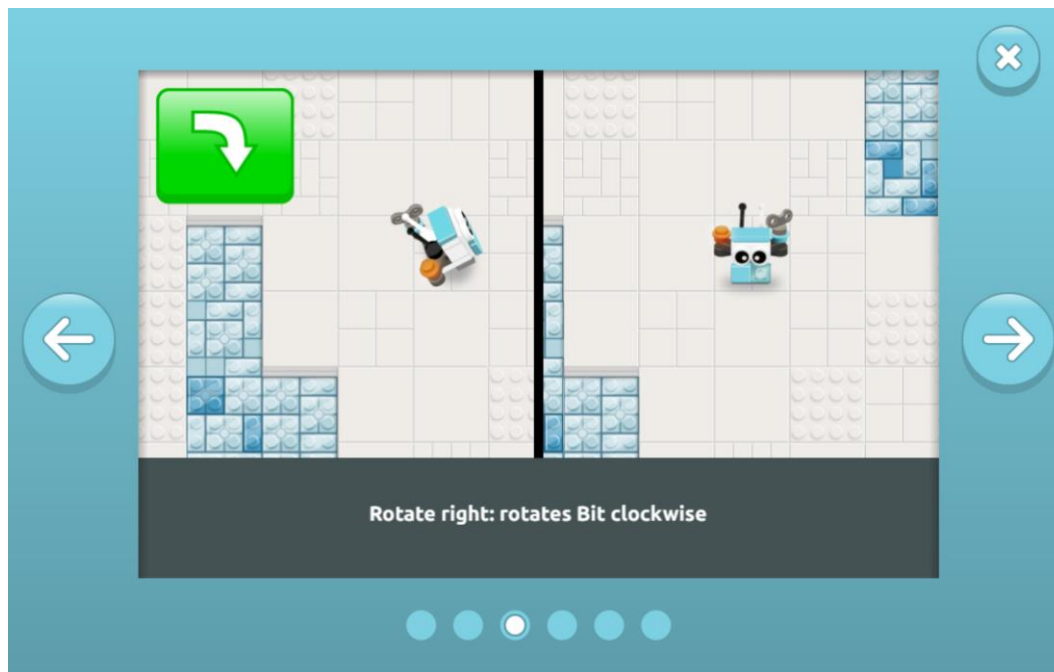


Ilustracja 7. Okno programu aplikacji internetowej LEGO® Bits and Bricks

Źródło: zrzut ekranu, opracowanie własne

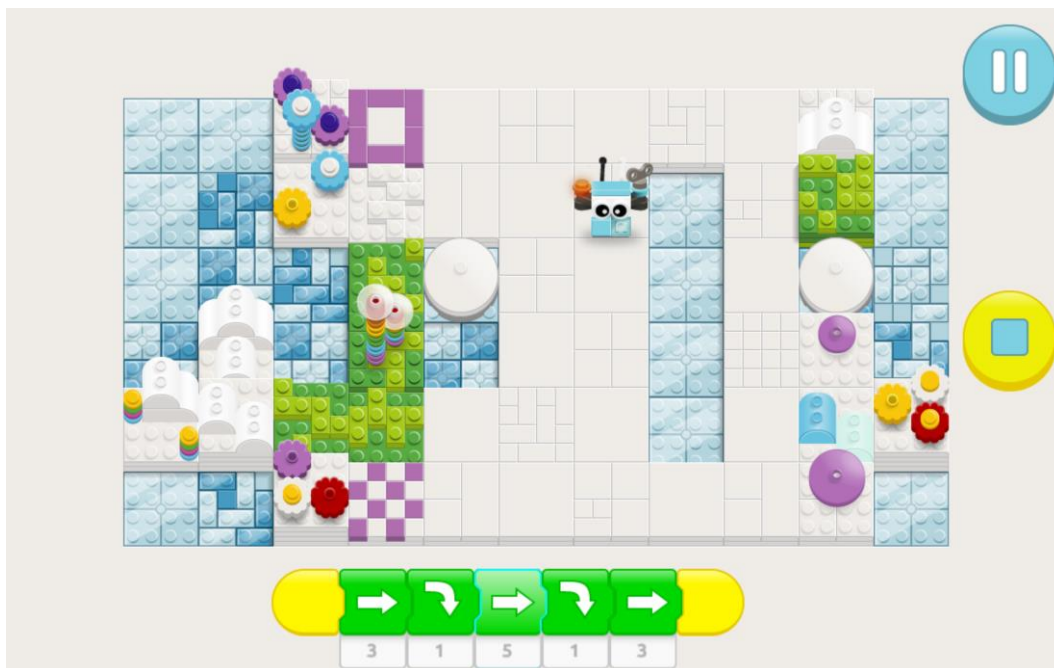


Zanim zmierny się z konkretnym zadaniem, użytkownik zostaje zapoznany z symboliką poleceń sterujących bohaterem gry. Przygodę rozpoczynamy od prostej planszy i wraz z realizacją kolejnych, trudność zadań rośnie.



Ilustracja 8. Okno programu aplikacji internetowej LEGO® Bits and Bricks
Źródło: zrzut ekranu, opracowanie własne

Podstawowe zadanie polega na przejściu z klocka START na klocek META (biało-fioletowa szachownica) z koniecznością pokonania różnych przeszkód. Tworząc algorytm, możemy używać powtórzeń, wskazując po wybranym poleceniu ile razy będzie ono wykonane. Odważny robot LEGO potrzebuje Twojej pomocy! Powiedz Bitowi, co ma robić.



Ilustracja 9. Okno programu aplikacji internetowej LEGO® Bits and Bricks z przykładowym rozwiązaniem zadania

Źródło: zrzut ekranu, opracowanie własne

Świetnie materiały szkoleniowe skierowane do nauczycieli i innych osób zajmujących się działalnością edukacyjną znaleźć można na stronie oficjalnej Europejskiego Tygodnia Kodowania - www.codeweek.eu. CodeWeek to społeczna inicjatywa, w ramach której europejskie państwa rywalizują w liczbie zorganizowanych wydarzeń związanych z programowaniem. Akcja dedykowana jest wszystkim, którzy chcą rozpocząć lub kontynuować przygodę z programowaniem. Jak mówią organizatorzy, w CodeWeek jest miejsce dla każdego – nie trzeba być specjalistą, by móc wziąć udział w tej inicjatywie. Wystarczą dobre chęci. Na tej platformie można znaleźć wiele inspiracji dotyczących prowadzenia zajęć z programowania dla dzieci w różnych wieku i o różnym poziomie zaawansowania.

Darmowe materiały szkoleniowe i informacje, które pomogą postawić pierwsze kroki na drodze do zaplanowania następnych innowacyjnych zajęć dla uczniów dostępne są na stronach [link do codeweek.eu](http://link.do.codeweek.eu) czy [link do code.org](http://link.do.code.org). Bezpłatne materiały szkoleniowe, przygotowane zostały tak, żeby mógł z nich skorzystać



każdy nauczyciel, bez względu na dotychczasowe doświadczenia w nauce kodowania.

Z wykorzystaniem dostępnych materiałów nauczyciel poprowadzi zajęcia z kodowania m.in. na edukacji przyrodniczej. Bez wykorzystania specjalistycznych środków dydaktycznych, za to z wykorzystaniem kart pracy oraz tablicy interaktywnej. Ćwiczenia w kodowaniu cyfr, tworzeniu i odczytywaniu liczb zapisanych w postaci kodu, wyszukiwaniu dzielników liczb w różnych działaniach arytmetycznych, zapisywaniu dat na osi czasu, kodowaniu rysunków, graficzne dyktando, gry online to tylko kilka z propozycji innowacyjnych i kreatywnych lekcji.

3.3 Mierniki efektów kształcenia

Potwierdzeniem, że dany uczeń osiągnął określone efekty uczenia się, możliwe jest zwykle z wykorzystaniem więcej niż jednej metody weryfikacji. Podstawową zasadę doboru metod weryfikacji można sformułować następująco: „Efektom uczenia się o charakterze praktycznym dobierz praktyczne metody weryfikacji”. Różnica między „wiedzieć, jak coś zrobić” a „umieć coś zrobić” jest kluczowa. Jeżeli efekty uczenia dotyczą sprawdzenia umiejętności wykonania określonej czynności, metody weryfikacji, powinny przewidywać przestrzeń do prowadzenia obserwacji lub narzędzia wykonania tej czynności. W wypadku umiejętności uniwersalnych czy kompetencji społecznych nie zawsze możliwe będzie zaprojektowanie jednego, sumatywnego procesu weryfikacji efektów uczenia się po zakończeniu kształcenia. W wypadku tego rodzaju kompetencji konieczne jest innowacyjne i elastyczne podejście. Na przykład sprawdzenie umiejętności dotyczących komunikacji interpersonalnej może odbywać się na podstawie obserwacji oraz wniosków zestawionych w formie tabeli 1.



Tabela 1. Efekty kształcenia i kryteria weryfikacji

Uczeń potrafi pracować w grupie

Kryteria/poziom osiągnięcia	Niesatysfakcjonujący	Satysfakcjonujący	Bardzo dobry
Dzieli się równo pracą	Bazuje na pracy innych	Wykonuje całość powierzonych mu prac	Wykonuje wszystkie zadania oraz pomaga innym
Wykonuje swoje zadania w ramach grupy	Bazuje na pracy innych, musi być często motywowany	Wykonuje prawie wszystkie zadania związane z powierzoną rolą, czasami musi być motywowany	Samodzielnie odgrywa wszystkie powierzone role z dużym zaangażowaniem
Słucha innych	Nie dopuszcza innych do głosu	Słucha innych z niewielkim zaangażowaniem	Zachowuje odpowiednie proporcje w słuchaniu i wypowiedzianiu się

Źródło: opracowanie własne

Nauczyciel dokonuje weryfikacji założonych celów lekcji, stosując m.in. metody obserwacji uczniów, technikę zdań niedokończonych, a także samoocenę ucznia. Uczeń w procesie nauczania i uczenia się powinien wiedzieć, z czego może być dumny, co potrafi, a nad czym powinien się pochylić i co udoskonalić. Powinien umieć dostrzegać swoje mocne i słabe strony, dzielić się tą wiedzą z innymi i móc z całym przekonaniem powiedzieć, czego się nauczył i jaką drogą do tego dotarł. Uczniowie wykazują zróżnicowany poziom rozwoju. Każdy z nich rozwija się w swoim tempie. Jedni szybciej, inni trochę wolniej, ale równie skutecznie. Niektórzy obcuja z technologią na co dzień i potrafią swobodnie posługiwać się

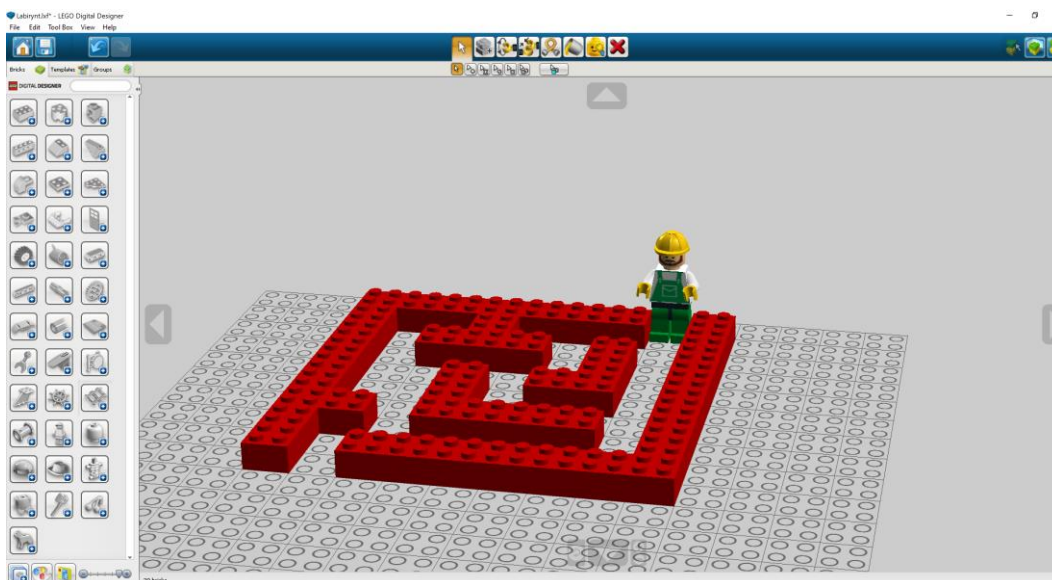


komputerem czy tabletem, dla innych ten kontakt będzie świeży, dlatego nie sposób oceniać wszystkich w ten sam sposób. Dlatego w obszarze oceniania proponuję wykorzystanie takich rozwiązań, które pozwolą dostrzec wysiłek i zaangażowanie, a także dadzą szansę na rozwój ucznia i osiągnięcie sukcesów. Należy kłaść nacisk na wykorzystanie oceniania kształtującego jako wiodącego w procesie nauczania informatyki w klasach 1-3. Jest ono formą oceniania opisowego, nastawionego na wspieranie uczenia się. Możemy śmiało powiedzieć, że jest ocenianiem wspierającym rozwój, ponieważ pokazuje uczniowi, jakie umiejętności udało mu się już opanować, a nad jakimi powinien jeszcze popracować. Uczeń ma świadomość tego, co umie, potrafi to wyartykułować, co nie byłoby możliwe przy ocenianiu tradycyjnym wyrażonym cyfrą. Dzieje się tak za sprawą jasno określonych i sformułowanych w języku ucznia kryteriów osiągnięcia sukcesu, które nauczyciel przedstawia przed każdą lekcją.



4. Zaprogramowane klocki

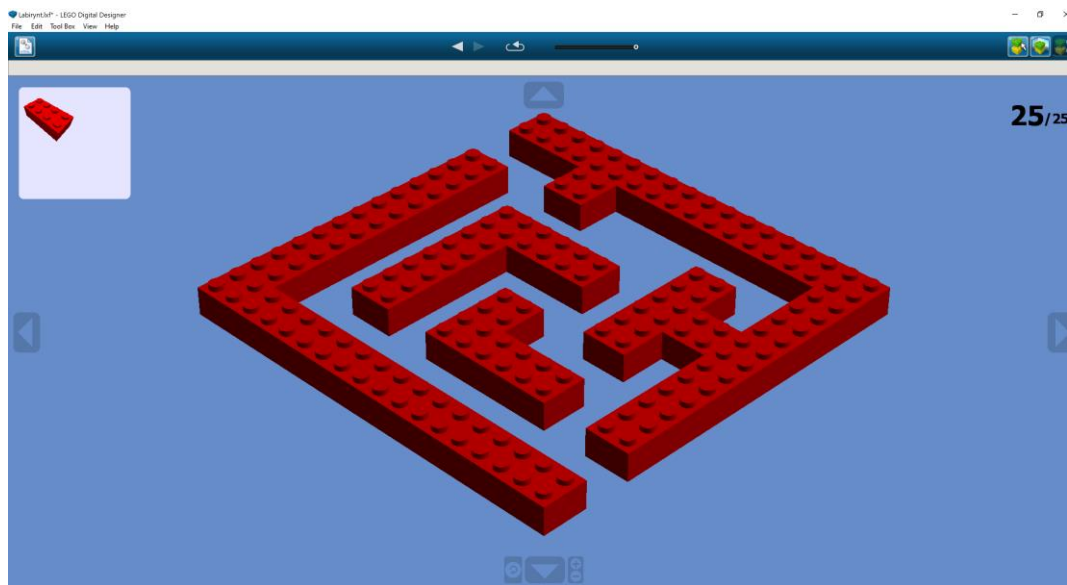
Dzięki wykorzystaniu programu LEGO® Digital Designer zarówno nauczyciel jak i uczniowie mogą budować wirtualnie modele z klocków LEGO. Program pozwala na budowanie z uwielbianych przez dzieci klocków w wirtualnym środowisku. Po stworzeniu modelu możemy obejrzeć go we wszystkich trzech wymiarach. Aplikację możemy obsługiwać zarówno myszką jak i klawiaturą, co bardzo ułatwia tworzenie nowych modeli i ogólną obsługę programu.



Ilustracja 10. Okno programu LEGO® Digital Designer

Źródło: zrzut ekranu, opracowanie własne

Wykorzystanie narzędzia do tworzenia wirtualnego labiryntu lub dowolnej konstrukcji rozwija kreatywność uczniów, myślenie przestrzenne, oraz umożliwia wykorzystanie nowoczesnych technologii w wspieraniu kompetencji kluczowych u uczniów. Umiejętność odczytywania modelu wirtualnego, planowania i przewidywania skutków własnych działań stanowi idealne narzędzie dla uczniów klas 1-3 szkoły podstawowej. Bardzo ważne jest również poczucie sukcesu i dumy z własnej pracy, motywowanie do działania, wyzwalanie inicjatywy.



Ilustracja 11. Labirynt zaprojektowany w programie LEGO® Digital Designer
Źródło: zrzut ekranu, opracowanie własne



PODSUMOWANIE

I etap edukacyjny to według ujęcia D. Super'a i E. Ginzberga okres przejścia od orientacji na zabawę do orientacji na pracę, co wzmacniane jest poprzez odgrywanie różnorodnych ról zawodowych. Uczeń ćwiczy umiejętność planowania własnych działań i opracowuje na podstawie swoich doświadczeń strategie rozwiązywania pojawiających się trudności. Kompetencje te mają charakter transferowalny, a więc mogą być wykorzystywane zarówno w szkole, jak i poza nią, np. na rynku pracy. Orientacja na pracę, a więc inicjowanie i podtrzymywanie działań ukierunkowanych na rozwiązanie konkretnego problemu (nawet pomimo niepowodzeń) ma kluczowe znaczenie w procesie kształtowania się zaangażowania zadaniowego i uogólnionej postawy do pracy i edukacji. Stąd tak istotne jest wspieranie uczniów w tym obszarze już na samym początku nauki w szkole podstawowej.

Efektywne działania na rzecz kształtowania orientacji zawodowej powinny również uwzględniać zróżnicowanie zdolności i zainteresowań uczniów tak, aby wzmocnić ich świadomość w zakresie posiadanych uzdolnień i pasji, a także zwiększyć motywację do działania.

Uczenie programowania wpisuje się w strategiczne cele związane z rozwojem kapitału ludzkiego i społecznego Polski, wyrównywaniem szans osób z różnych środowisk i regionów oraz budowaniem gospodarki konkurencyjnej, opartej na wiedzy i zrównoważonej. Kształcenie w tym kierunku jest jednym z warunków rozwoju innowacji technologicznych.

Należy zauważyć, że umiejętność programowania to tylko część szeroko rozumianych kompetencji cyfrowych i medialnych. Powszechna obecność technologii komunikacyjnych (zarówno tradycyjnych mediów masowych, jak i mediów elektronicznych) sprawia, że jakość życia współczesnych społeczeństw w coraz wyższym stopniu zależy od zdolności rozumienia i korzystania z informacji, które docierają do nas za pośrednictwem mediów.



BIBLIOGRAFIA Z UWZGLĘDNIENIEM NETOGRAFII

Kwiatkowski, S.M. (2018). *Kompetencje przyszłości*. Warszawa: Fundacja Rozwoju Systemu Edukacji

Pitler, H., Hubbell, E.R., Kuhn, M. (2015). *Efektywne wykorzystanie nowych technologii na lekcjach*, przeł. Patrycja Szmyd. Warszawa: Centrum Edukacji Obywatelskiej, sygn. 32944-945.

Borawska-Kalbarczyk, K. (2015). *Kompetencje informacyjne uczniów w perspektywie zmian szkolnego środowiska uczenia*. Warszawa: Wydawnictwo Akademickie Żak, sygn. 32974.

Podstawa programowa (2017). *Szkoła podstawowa I-III* [online]. Dostępny w: <https://podstawaprogramowa.pl/Szkoła-podstawowa-I-III> [Dostęp 15.10.2021]

Chilmon, E. (2015). *Edukacja wczesnoszkolna - scenariusze lekcji z komputerem, tabletem i nie tylko*. Wrocław: Presscom.

Burski, J. (2013). *Umiejętności Polaków – wyniki Międzynarodowego Badania Kompetencji Osób Dorosłych (PIAAC)*. Warszawa: Instytut Badań Edukacyjnych. Dostępny w: <http://eduentuzjasci.pl/images/stories/publikacje/ibe-raport-PIAAC-2013.pdf> [Dostęp 24.10.2021].

Kuźmińska-Sołśnia, B. (2018). *Nauka programowania/kodowania w edukacji najmłodszych*. Rzeszów: Uniwersytet Rzeszowski.

Stańko, J. (2017). *Organizacja przestrzeni sprzyjającej edukacji informatycznej w przedszkolu i klasach I-III szkoły podstawowej*. Warszawa: Ośrodek Rozwoju Edukacji.

European Commission/EACEA/Eurydice, 2012. *Developing Key Competences at School in Europe: Challenges and Opportunities for Policy – 2011/12. Eurydice Report* [online]. Luxembourg: Publications Office of the European Union.

Dostępny w: <https://eurydice.org.pl/wp-content/uploads/2013/03/145EN.pdf> [Dostęp 11.10.2021]

Dziennik Urzędowy Unii Europejskiej (2006/962/WE), 2006. *Kompetencje kluczowe w uczeniu się przez całe życie - Europejskie ramy odniesienia*. Załącznik



do Zalecenia Parlamentu Europejskiego i Rady z dnia 18 grudnia 2006 r. w sprawie kompetencji kluczowych w procesie uczenia się przez całe życie [online], Dziennik Urzędowy Unii Europejskiej (2006/962/WE) z dnia 30.12.2006.

Dostępny w:

<http://eurlex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2006:394:0010:0018:pl:PDF> [Dostęp 20.10.2021].

Janczak, D. (2018). *TIK w podstawie programowej*. Dyrektor Szkoły, nr 3, s. 42-45.

Rostkowska, M. (2019). *Model SAMR a TIK: nauczyciel buduje kompetencje swoje i swoich uczniów*. Meritum, nr 1, s. 95-101.

Bałachowicz, J., Korwin-Szymanowska, A., Lewandowska, E., Witkowska-Tomaszewska, A. (2017). *Zrozumieć uczenie się – zmienić wczesną edukację*. Warszawa: Wydawnictwo Akademii Pedagogiki Specjalnej.

Kwapiszewska-Wolska, A., Bałdyga, B., Walter, N. (2018). *Przygotowanie do programowania. Gry logiczne i ćwiczenia unplugged z darami Froebela*. Lublin: Froebel.pl.

Papert, S. (1996). *Burze mózgow. Dzieci i komputery*. Warszawa: Wydawnictwo Naukowe PWN.

Syśło, M. (2014). *Myślenie komputacyjne. Nowe spojrzenie na kompetencje informatyczne*. Materiały konferencyjne, „Informatyka w Edukacji, XI”, UMK Toruń, s. 15-32.

Wing, J. M. (2008). *Computational thinking and thinking about computing*. Philosophical transactions of the royal society of London A: mathematical, physical and engineering sciences, 366(1881), 3717-3725.

Pałys, K. (2015). *Język komputerów jest dziecinnie prosty – wystarczy Godzina Kodowania*. Dostępny w: <http://natemat.pl/163635,jezyk-komputerow-jest-dziecinnie-prosty-wystarczy-godzina-kodowania>. [Dostęp 8.10.2021].

Polak, M., *Programowanie rozwija, zatem mądrze uczmy programować*, <http://superkoderzy.pl/programowanie-rozwija-zatem-madrze-uczmyprogramowac>.



Ośrodek Rozwoju Kompetencji Edukacyjnych. (2018). *Programowanie to nie tylko kodowanie. Co oznacza programowanie dla nie informatyków*. Dostępny w:

<https://orke.pl/blog/programowanie-nie-tylko-kodowanie-co-oznacza-programowanie-dla-nieinformatykw/> [Dostęp 8.10.2021].



WYKAZ ILUSTRACJI

Ilustracja 1. Strzałki wraz z opisem do kodowania kierunku ruchu	18
Ilustracja 2. Projekt maty edukacyjnej.....	19
Ilustracja 3. Przykładowy labirynt ułożony na macie edukacyjnej	20
Ilustracja 4. Karta pracy dla uczniów	22
Ilustracja 5. Wyjaśnienie pól karty pracy do uzupełnienia przez uczniów	22
Ilustracja 6. Fragment przykładowego programu.....	22
Ilustracja 7. Okno programu aplikacji internetowej LEGO® Bits and Bricks	23
Ilustracja 8. Okno programu aplikacji internetowej LEGO® Bits and Bricks	24
Ilustracja 9. Okno programu aplikacji internetowej LEGO® Bits and Bricks z przykładowym rozwiązaniem zadania	25
Ilustracja 10. Okno programu LEGO® Digital Designer	29
Ilustracja 11. Labirynt zaprojektowany w programie LEGO® Digital Designer	30

WYKAZ TABEL

Tabela 1. Efekty kształcenia i kryteria weryfikacji	27
---	----