

Agata M. Wijata, Anita Pollak

Wsparcie nauki programowania w edukacji wczesnoszkolnej z wykorzystaniem EduMATRIX

Edukacja - Technika - Informatyka nr 4(18), 319-324

2016

Artykuł został opracowany do udostępnienia w internecie przez Muzeum Historii Polski w ramach prac podejmowanych na rzecz zapewnienia otwartego, powszechnego i trwałego dostępu do polskiego dorobku naukowego i kulturalnego. Artykuł jest umieszczony w kolekcji cyfrowej bazhum.muzhp.pl, gromadzącej zawartość polskich czasopism humanistycznych i społecznych.

Tekst jest udostępniony do wykorzystania w ramach dozwolonego użytku.



AGATA M. WIJATA¹, ANITA POLLAK², MARCIN D. BUGDOL³,
MARIA J. BIENKOWSKA⁴, ANDRZEJ W. MITAS⁵

Wsparcie nauki programowania w edukacji wczesnoszkolnej z wykorzystaniem EduMATRIX

Support for learning programming in early education using EduMATRIX

¹ Magister inżynier, Politechnika Śląska, Wydział Inżynierii Biomedycznej, Katedra Informatyki i Aparatury Medycznej, Polska

² Doktor, Uniwersytet Śląski, Wydział Pedagogiki i Psychologii, Instytut Psychologii, Polska

³ Doktor inżynier, Politechnika Śląska, Wydział Inżynierii Biomedycznej, Katedra Informatyki i Aparatury Medycznej, Polska

⁴ Magister inżynier, Politechnika Śląska, Wydział Inżynierii Biomedycznej, Katedra Informatyki i Aparatury Medycznej, Polska

⁵ Profesor doktor habilitowany inżynier, Politechnika Śląska, Wydział Inżynierii Biomedycznej, Katedra Informatyki i Aparatury Medycznej, Polska

Streszczenie

Obecnie bardzo duży nacisk w procesie edukacji kładziony jest na rozwijanie umiejętności logicznego i abstrakcyjnego myślenia, co jest niezbędne do nauki programowania. Powszechnie dostępne są systemy, które pomagają oswoić się z tym zagadnieniem już od najmłodszych lat. W niniejszej pracy przedstawiona została alternatywa dla takich aplikacji – bloczki *EduMATRIX*. Ich głównym atutem jest nauczanie zagadnień związanych z programowaniem bez konieczności przebywania przed ekranem komputera. Ponadto zaproponowano metodę walidacji użyteczności *EduMATRIX* oraz innych dostępnych pomocy dydaktycznych, która pozwoli na wskazanie skutecznej formy nauki dla młodych użytkowników.

Słowa kluczowe: *EduMATRIX*, programowanie, edukacja wczesnoszkolna, zabawki.

Abstract

Currently, the development of the skills of logical and abstract thinking is emphasized in the education process. This is essential for learning programming. Systems that help to be familiar with this issue since an early age are widely available. In this paper an alternative for such applications is presented – *EduMATRIX*. Its main advantage is the teaching of programming without the need to stay in front of a computer screen. Moreover, a validation method of *EduMATRIX* and other available teaching aids, which will identify effective form of learning for young users, was proposed.

Key words: *EduMATRIX*, programming, early education, toys.

Wstęp

Rzeczony człowiek w okresie dziecięcym jest silnie ukierunkowany na poznawanie otoczenia i zasad nim rządzących. Odkrywanie świata wymaga uczenia się, które przebiega w oparciu o obserwację i naśladowanie innych. Z perspektywy antropologii kulturowej, dzięki tego rodzaju zachowaniom, młoda istota ludzka dąży do maksymalizacji swoich szans na samodzielne przetrwanie, gdy zabraknie jej opiekunów. Jednak obecnie w krajach rozwiniętych na naukę sposobów zaspokajania podstawowych potrzeb (zapewnienie sobie schronienia, zdobywanie pożywienia) nie poświęca się tak wiele czasu, jak setki lat temu. Zamiast tego dziecko skłania się ku szybszemu poznawaniu liter i nauce, a w ciągu ostatnich dwóch dekad również ku obsłudze urządzeń elektronicznych. Postęp technologiczny dokonujący się w sferze tzw. elektroniki użytkowej oraz powszechna dostępność tych urządzeń powoduje, że dzieci mają z nią kontakt od samego początku życia (aparatus fotograficzny jest nieodzownym atrybutem każdego ojca na porodówce).

Należy jednak zauważyć, że pomimo przeniesienia wysiłku na poznawanie nowych (w stosunku do poprzednich generacji) dziedzin umiejętność logicznego oraz abstrakcyjnego myślenia wciąż jest niezmiennie wymagana do zapewnienia sobie komfortowego życia, a nawet zyskuje ona coraz większe znaczenie w obecnym świecie. Taki rodzaj myślenia jest potrzebny m.in. w nauce programowania, z którą w chwili obecnej próbujemy oswoić dzieci już od najmłodszych lat, wykorzystując różne systemy wspomagające.

Przykładowe systemy wspomagania nauki programowania

W literaturze dostępne są opisy licznych narzędzi, najczęściej w postaci gier czy aplikacji, wprowadzających w tematykę programowania oraz wspierających jego trening.

Aplikacja Scratch jest wizualnym środowiskiem programistycznym umożliwiającym tworzenie interaktywnych projektów multimedialnych, takich jak animowane historie, gry, raporty, karty okolicznościowe, teledyski, projekty naukowe, podręczniki, symulacje oraz projekty muzyczne [Maloney i in. 2010]. Program Scratch jest narzędziem stworzonym dla dzieci i młodzieży celem zaprzyjaźnienia z programowaniem. Odbiorcy w wieku 8–16 lat uczą się ważnych pojęć matematycznych i obliczeniowych, jak również kreatywnego myślenia, systematyczności oraz pracy w zespole, które to elementy należą do podstawowych umiejętności pożądanych w XXI w. [Resnick i in. 2009]. Autorzy programu za cel główny wyznaczyli sobie pielęgnowanie nowego, młodego pokolenia kreatywnych, systematycznych myślicieli, którzy będą mieli możliwość wyrażenia swoich pomysłów za pomocą programowania. Sugerują oni, że w świecie, w którym każdy korzysta z komputera oraz telefonu komórkowego, nabycie umiejętności programowania wymaga jedynie umiejętności czytania.

Innym przykładem aplikacji pozytywnie odbieranej przez osoby początkujące w dziedzinie programowania jest program Alice, który jest wizualnym środowiskiem 3D. Przeprowadzone przez E.R. Sykes badania wykazały, iż osoby ćwiczące programowanie za pomocą aplikacji Alice znacząco lepiej radziły sobie z tym zagadnieniem [Sykes 2007].

Problem ze zrozumieniem sposobu przebiegu realizacji programu, czyli wizualizacja procesu wykonania zadania przez maszynę, uważa się za jedną z przyczyn trudności w programowaniu [Cooper i in. 2000]. Osoba ucząca się w oparciu o wizualizację zagadnienia poznaje pojęcie pętli i szkoli się w jej stosowaniu. W dalszej nauce korzysta ona ze gromadzonych doświadczeń, działań zakończonych sukcesem i porażką, na podstawie których koryguje podejmowane decyzje [Cooper i in. 2000]. Stąd, wizualizacja kolejnych kroków rozwiązywanego problemu może zarówno być pomocna w procesie zrozumienia programowania, jak i skutecznie wpływać na podniesienie motywacji do dalszej nauki rozwiązywania problemów.

Istnieją również propozycje pozwalające rozpocząć naukę programowania najmłodszym dzieciom (poniżej 8. roku życia). Przykładem mogą być Electronic Blocks, których ocena przez P. Wyeth i wsp. wykazuje, iż dzieci bez znajomości pojęć związanych z programowaniem oraz z niewielką wiedzą z zakresu czytania mogą z minimalną pomocą dorosłych nauczyć się prostych elementów programowania. Klocki Electronic Blocks są elektronicznymi zabawkami, które cechują pewne właściwości, np. klocek emituje światło, mruga nim lub też odtwarza dźwięki przyrody. Dzieci za pomocą klocków „budują” programy, czyli proponują wykonanie jakiegoś ciągu zadań, który to jest weryfikowany na bieżąco poprzez emisję światła lub dźwięku. Przeprowadzone badania na grupie 40 dzieci w wieku od 4 do 8 lat wykazały zainteresowanie i zaangażowanie dzieci podczas realizacji zadań z wykorzystaniem klocków. Odnotowano również frustrację dzieci, kiedy ułożony przez nie schemat nie działał zgodnie z oczekiwaniami. Poziom zrozumienia działania klocków wzrósł z czasem ich użytkowania [Wyeth, Purchase 2002].

Robot Light-Bot będący grą edukacyjną jest kolejnym narzędziem, które można wykorzystać w procesie nauki programowania przez najmłodszych. Celem projektu Light-Bot jest identyfikacja problemu, jego ocena oraz nauka szukania rozwiązania. Dziecko uczy się rozkładać problem na kilka etapów i w każdym z nich identyfikuje zadanie do wykonania, a następnie planuje kolejne kroki, których wykonanie umożliwi osiągnięcie wyznaczonego celu. W trakcie korzystania z narzędzia uczeń ma szansę na zapoznanie się i utrwalenie podstawowych pojęć związanych z informatyką. Obserwuje powtarzalność zjawisk, co przekłada się na zrozumienie oraz umiejętność zastosowania np. pojęcia pętli [Gouws i in. 2013].

EduMatrix jako pomoc dydaktyczna

Proponowany przez autorów zestaw edukacyjny *EduMATRIX* pozwala na doskonalenie umiejętności matematycznych, algorytmicznych i programistycznych już na wczesnym etapie nauczania. Jest to komplet drewnianych elementów, które podczas układania pozwalają nawiązać do systemu komputerowego. Drewniane bloczki i plansze nawiązują do sprzętu komputerowego, umieszczone na nich oznaczenia, symbole i podstawowe reguły oraz ustalony sposób ich zapisu odpowiadają systemowi operacyjnemu, natomiast proponowane zabawy i zadania korespondują z oprogramowaniem aplikacyjnym.

Podstawowe zadania wykonywane przy użyciu *EduMATRIX* dotyczą umiejętności liczenia, czytania oraz interpretowania znaków alfanumerycznych. Na wyższym poziomie kształtowane są umiejętności z zastosowaniem operatorów do działań arytmetycznych, a także wykorzystywane są instrukcje zawierające elementy myślenia algorytmicznego, programowania oraz działań właściwych dla arkusza kalkulacyjnego.

W skład zestawu wchodzi szereg bloczków, które reprezentują kolejno:

- kolory – zastosowane zostały bloczki w barwach czerwonej, zielonej, niebieskiej oraz żółtej, które nawiązują do przestrzeni roboczych RGB oraz CMYK,

- znaki alfanumeryczne – wykorzystane zostały cyfry od 0 do 9 oraz litery od A do J; wyróżnić można także symbol pusty – bloczek bez nadruku, który ma pomóc w zrozumieniu wartości zero,

- operatory arytmetyczne oraz inne symbole matematyczne – wśród nich wyróżniono dodawanie, odejmowanie, mnożenie, dzielenie, znak mniejszości/większości oraz równości,

- umowne symbole i instrukcje kodu programowania – jako symbole właściwe dla arkusza kalkulacyjnego zaproponowano schematycznie przedstawione funkcje sumy, minimum i maksimum oraz dwukropek; jako symbole dla składni i funkcji języka programowania przygotowane zostały bloczki z symboliczną grafiką pętli, kroku, warunku oraz nawiasy klamrowe; dla nawiązania do programowania obiektowego możemy korzystać z bloczków reprezentujących stan obiektu (jak np. jego pozycja na mapie czy kierunek ruchu) oraz akcji wpływających na stan obiektu (takich jak obrót w prawo lub lewo),

- losowanie – przygotowano kostki pozwalające na losowanie cyfr z zakresu 1–6 zapisanych jako cyfry arabskie oraz kropki (standardowa notacja jak na kostce do gier planszowych), liter z zakresu A–F oraz kolorów.

Bloczki umieszcza się na drewnianych planszach z równomiernie rozmieszczonymi kwadratowymi otworami, które pozwalają na stabilne ułożenie bloczka w odpowiedniej pozycji. Plansza posiada z każdej strony inny nadruk. Po jednej stronie nadruk składa się ze z liter i cyfr, które opisują odpowiednio kolumny i wiersze – zapis ten nawiązuje do arkusza kalkulacyjnego. Po drugiej stronie

nadrukowane zostały osie X i Y z naniesionymi znacznikami o interwale równym 1 – ta strona planszy nawiązuje do kartezjańskiego układu współrzędnych. Dodatkowo, w prawym górnym rogu na tej stronie planszy nadrukowano różę wiatrów, która pozwala na wykorzystanie planszy jako zorientowanej mapy cyfrowej. Kolejnym elementem są opisy kolumn przyporządkowujące je do określonych potęg liczby 10 oraz linie przecinające planszę po środku, które mogą stanowić osie symetrii przy nauce symetrii osiowej oraz punktowej.

Propozycja walidacji *EduMATRIX*

Walidację użyteczności blozków *EduMATRIX* jako narzędzia wspomagającego nauczanie programowania w szkole podstawowej proponuje się przeprowadzić w formie zajęć dodatkowych dla dzieci z klas nauczania początkowego. Uwzględniając intensywne zmiany rozwojowe u dzieci, w szczególności w zakresie myślenia logicznego oraz wymagania związane z nauką programowania, do eksperymentu zdecydowano zaprosić dzieci z klasy II, będące w wieku ok. 8 lat.

Eksperyment zaplanowano na czas jednego semestru (6 miesięcy), w czasie którego dzieci zostałyby poddane systematycznemu treningowi umiejętności tworzenia algorytmów, przy czym czas ten jest na tyle długi, aby rzeczywiście możliwe byłoby nabycie tej umiejętności. Cykl nauki obejmowałby 60 spotkań po 30 minut, zajęcia odbywałyby się trzy razy w tygodniu, co w sumie stanowiłoby 30 godzin zegarowych. Po tym czasie można przewidywać, że wykorzystanie określonych narzędzi wywoła zmianę polegającą na przyswojeniu sobie umiejętności tworzenia algorytmów rozwiązujących zadany problem.

Struktura grupy wyglądałaby następująco: 20 dzieci korzystałaby z środowiska programistycznego Scratch (rozwiązanie softwarowe), 20 uczniów programowałoby robota Light-Bot (rozwiązanie hardwarowe), 20 korzystałoby z blozków *EduMATRIX* oraz 20 stanowiłoby grupą kontrolną, która uczyłaby się programowania za pomocą metody kartka i ołówek.

Kursy poprzedzone byłyby standaryzowanym zadaniem oceniającym umiejętność rozwiązywania zadań problemowych. Ma to na celu uzyskanie informacji o poziomie tej zdolności u dzieci, aby w dalszym etapie walidacji porównać zachodzące zmiany i zestawić je z rezultatami prowadzonych treningów w poszczególnych grupach. Do oceny wybrano test matryc Ravena w wersji standardowej, kolorowej dla dzieci, jako że uznaje się go za najlepszą miarę czynnika inteligencji ogólnej (czynnik g), który przejawia się m.in. w tworzeniu konstruktywów umożliwiających radzenie sobie ze złożonymi problemami obejmującymi wiele wzajemnie powiązanych zmiennych [Kostrzewski 1997: 35]. Wybór testu podyktowany jest wysoką przydatnością do pomiaru myślenia konkretnego i abstrakcyjnego. Test obejmuje 36 zadań polegających na ustaleniu relacji między elementami wzoru (matrycy) i wskazaniu brakującego elementu wzoru z podanych poniżej matrycy. Niestety test nie posiada wersji równoległej, cho-

ciaż zespół badawczy stoi na stanowisku, że po sześciu miesiącach można badać ponownie, używając wersji takiej jak poprzednio. Test planuje się powtórzyć po ostatnim spotkaniu z nauki programowania.

Analiza uzyskanych wyników ma uwzględniać czy zachodzą istotne związki pomiędzy efektami uzyskanymi w wyniku szkolenia (w ocenie prowadzącego) a poziomem umiejętności rozwiązywania problemów (mierzonych za pomocą testu matryc Ravena) celem wskazania formy nauki skutecznej dla młodych użytkowników.

Podsumowanie

W artykule przedstawiono opis systemu *EduMATRIX*, który umożliwia rozwój umiejętności logicznego i abstrakcyjnego myślenia poprzez realizację zadań z zakresu matematyki, algorytmiki i programowania. Użycie zestawu drewnianych bloczków stanowi doskonałą alternatywę dla popularnych narzędzi dostępnych obecnie dla dzieci, wśród których znakomita większość wymaga spędzania czasu przy komputerze. Głównym problemem, który należy obecnie rozwiązać, jest przekonanie rodziców oraz nauczycieli do korzyści płynących z wykorzystania *EduMATRIX* – kontakt z rzeczywistym przedmiotem zamiast z komputerem czy rozwój intelektualny i manualny. Potencjalny użytkownik może zniechęcić się nauką na „drewnianym komputerze”, gdyż kojarzyć mu się będzie z czymś archaicznym, a przecież przedstawione narzędzie do edukacji w takim obszarze jest absolutnie nowatorskie.

Literatura

- Cooper S., Dann W., Pausch R. (2000), *Alice: A 3-D Tool for Introductory Programming Concepts*, „Journal of Computing Sciences in Colleges” vol. 15/5.
- Gouws L.A., Bradshaw K., Wentworth P. (2013), *Computational Thinking in Educational Activities: An Evaluation of the Educational Game Light-bot*, „Proceedings of the 18th ACM Conference on Innovation and Technology in Computer Science Education”.
- Kostrzewski J. (1997), *Testy inteligencji “Culture-fair” (powstanie, założenia, rodzaje, ocena)*, „Acta Universitatis Lodziensis. Folia Psychologica” no. 1.
- Maloney J., Resnick M., Rusk N., Silverman B., Eastmond E. (2010), *The Scratch Programming Language and Environment*, „ACM Transactions on Computing Education” vol. 10/4.
- Resnick M., Maloney J., Monroy-Hernández A., Rusk N., Eastmond E., Brennan K., Millner A., Rosenbaum E., Silver J., Silverman B., Kafai Y. (2009), *Scratch: Programming for All*, „Communications of the ACM” vol. 52/11.
- Sykes E.R. (2007), *Determining the Effectiveness of the 3D Alice Programming Environment at the Computer Science I Level*, „Journal of Educational Computing Research” vol. 36/2.
- Wyeth P., Purchase H. C. (2002), *Tangible Programming Elements for Young Children*, „Proceeding CHI EA '02 CHI '02 Extended Abstracts on Human Factors in Computing Systems”.