

# NAUKA PROGRAMOWANIA W SZKOLE PODSTAWOWEJ: SCRATCH

Lucyna Bała<sup>1</sup>, Piotr Bała<sup>2</sup>

<sup>1</sup>Zespół Szkół nr 8 w Toruniu

<sup>2</sup>ICM, Uniwersytet Warszawski

[wicedyrektorsp@zs8.eu](mailto:wicedyrektorsp@zs8.eu), [baala@icm.edu.pl](mailto:baala@icm.edu.pl)

*Abstract. The dedicated lessons on using computers are now included in early education. In this paper we present usage of the Scratch programming environment as a tool for teaching children programming. The lectures were delivered as a part of the Children's University at UMK Toruń (afterschool activity) and as regular lessons in the public school (Zespół Szkół nr 8, Toruń). We describe here realized lessons and provide insight on children achievements. The Scratch overview is also presented.*

## 1. Wstęp

Już od kilku lat zajęcia komputerowe są przedmiotem obowiązkowym w szkole podstawowej, jednak praktyka szkolna jeszcze nie wypracowała dobrych wzorców prowadzenia tych zajęć, zwłaszcza na pierwszym etapie edukacyjnym, w klasach 1-3. Co więcej, wydaje się że aktualne programy zajęć i podręczniki nie nadążają za rozwojem technologii i umiejętnościami uczniów. Tematyka zajęć w klasach 1-3 koncentruje się na rysowaniu czy korzystaniu z edytora tekstu. Umiejętności te są szybko przyswajane przez uczniów, proponowane ćwiczenia w posługiwaniu się myszką czy rysowaniu prostych elementów szybko się nudzą i pojawia się konieczność wprowadzania tematów bardziej zaawansowanych a jednocześnie dostosowanych do wieku uczniów i odpowiednio atrakcyjnych. Podobnie, w klasach 4-6 większość zajęć poświęcona jest edytorowi tekstu, tworzeniu prezentacji czy tworzeniu grafiki rastrowej.

Propozycja nowej podstawy programowej z informatyki [4] wprowadza do szkoły naukę programowania od najmłodszych klas. Propozycja jest ambitna, jednak w praktyce szkolnej, polskiej, nie tylko w Polsce, brak jest głębszych doświadczeń z realizacji tego typu zajęć w ramach zajęć obowiązkowych dla wszystkich uczniów. Większość doświadczeń pochodzi z zajęć pozalekcyjnych realizowanych w szkole lub poza nią.

W szczególności doświadczenia z zajęć prowadzonych w ramach Uniwersytetu Dziecięcego na UMK pokazują, że w szkole podstawowej, i to już w klasach 1-3, z powodzeniem można rozpocząć naukę tworzenia animacji i podstaw programowania z wykorzystaniem środowiska Scratch [1, 5] realizując w sposób atrakcyjny podstawę programową i to nie tylko w zakresie dotyczącym programowania. W starszych klasach można z kolei wprowadzać elementy programowania, takie jak zmienne czy instrukcje logiczne oraz tworzyć z uczniami proste gry.

W niniejszym artykule przedstawiamy praktyczne doświadczenia z nauki programowania realizowanej zarówno w formie zajęć pozalekcyjnych, jak też podczas zajęć obowiązkowych w szkole. W trakcie zajęć wykorzystywano środowisko Scratch umożliwiające programowanie w graficznym języku programowania.

## **2. Podstawy programowania w środowisku Scratch**

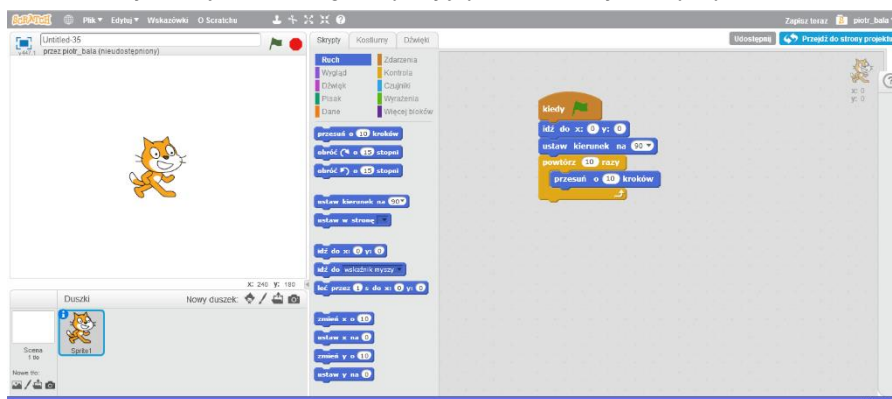
Scratch [5] jest tworzonym w MIT innowacyjnym środowiskiem programistycznym służącym do tworzenia prezentacji i animacji w sposób atrakcyjny dla ucznia. Umożliwia łatwe tworzenie aplikacji zawierających grafikę, dźwięk, ruch oraz zawierających szerokie możliwości interakcji z użytkownikiem. Scratch wykorzystuje bardzo ciekawe wizualne podejście do tworzenia aplikacji. Ogromną zaletą środowiska jest przyjazny interfejs, dostępność polskiej wersji językowej oraz dostępność bez opłat.

Scratch powstał w 2008 roku jako narzędzie do nauki programowania i tworzenia aplikacji i animacji. Pierwsza wersja była aplikacją instalowaną lokalnie na komputerze. W roku 2013 została udostępniona wersja 2.0 działająca jako aplikacja webowa dostępna bezpośrednio z przeglądarki www, a w roku 2015 wersja uproszczona na tablety (ScratchJunior). Istotną zaletą programu jest intuicyjny interfejs użytkownika oraz wizualny język programowania. W odróżnieniu od innych języków wizualnych opartych najczęściej o liniowy zapis poleceń, Scratch wykorzystuje oryginalny sposób zapisywania poleceń, przypominający programowanie w językach wyższego poziomu, takich jak C czy Java. Dzięki temu jest coraz powszechniej wykorzystywany do ilustrowania problemów czy wręcz nauki programowania na studiach, w tym na studiach informatycznych. Istotną cechą Scratch'a jest bardzo naturalne wprowadzenie do programowania obiektowego, w ciekawy sposób rozwiązano komunikację pomiędzy obiektami. Wszystkie te cechy przesadzają, że to środowisko jest bardzo atrakcyjnym i prostym narzędziem.

Najmłodszy użytkownicy Scratch'a mają 5 lat, najliczniejsza grupa użytkowników mieści się w przedziale 12-17 lat [5]. Wśród starszych uczniów zainteresowanie Scratch'em maleje na rzecz programowania w tradycyjnych językach. Obecnie Scratch ma ponad milion użytkowników na całym świecie, najwięcej w USA. W Polsce program jest popularyzowany od kilku lat, głównie dzięki takim inicjaty-

wom, jak Mistrzowie Kodowania. W ramach konferencji IwE od pięciu lat są prowadzone warsztaty dla nauczycieli. Nadal jednak brak jest odpowiednio przeszkolonych osób. Dodatkowo, w języku polskim dostępnych jest tylko kilka sylabusów dotyczących podstawowych tematów i najczęściej są to tłumaczenia materiałów angielskojęzycznych. Brak jest scenariuszy do wykorzystania w polskich warunkach edukacyjnych. Przykłady dostępne na stronach Scratch'a odnoszą się do warunków amerykańskich zarówno w zakresie praktyki szkolnej, jak też mentalności uczniów.

Scratch jest dostępny w postaci aplikacji webowej korzystającej z Adobe Air, wymaga więc komputera o niezbyt wygórowanych parametrach, ale z dostępem do Internetu. Istnieje też wersja *off-line* do zainstalowania lokalnie na komputerze. Scratch ma interfejs użytkownika w pełni przetłumaczony na język polski. Jeśli po uruchomieniu programu, komendy nie wyświetlą się w języku polskim, należy kliknąć na ikonę globusa, która znajduje się obok nazwy Scratch w lewym górnym rogu ekranu – pojawi się lista obsługiwanych języków, z której należy wybrać Polski.



Rysunek 1

Okno programu Scratch jest podzielone na kilka części (Rysunek 1). Pierwsza z nich, znajdująca się w środku, zawiera klocki – instrukcje, z których buduje się program. Klocki są podzielone na grupy odpowiadające typom instrukcji (ruch, wygląd, instrukcje sterujące, zmienne, i inne). Klocki różnych grup mają różne kolory, co poprawia czytelność tworzonego skryptu i umożliwia szybkie znajdowanie odpowiedniego polecenia. Prawa część zawiera ogólne właściwości aktualnie wybranego duszka lub sceny oraz kostiumy (lub tła), dźwięki oraz stworzone i przypisane do nich skrypty tworzące program. Pozostały fragment okna, znajdujący się po lewej stronie, składa się z dwóch części. Górna zawiera aktualny stan sceny. Natomiast w dolnej są przedstawione używane duszki oraz scena. W tej części znajdują się przyciski pozwalające na dodanie nowego duszka czy to przez jego

narysowanie, wybranie kostiumu duszka z pliku czy przez wylosowanie jednego z predefiniowanych kostiumów.

Tworzenie nowych skryptów jest niezwykle proste. Wystarczy przeciągnąć bloczek z pierwszej części okna i upuścić na środkowy obszar nazwany Skrypty. Instrukcje łączy się dokładając instrukcje do już istniejących. Za pomocą mechanizmu przeciągnij i upuść rozłącza się instrukcje. Aby usunąć instrukcję, należy najpierw odłączyć instrukcje poniżej klocka, który chcemy usunąć, następnie przesuwamy klocek z powrotem na pierwszą część okna lub klikamy prawym przyciskiem myszy na klocek i wybieramy z menu kontekstowego opcję Usuń.

Aby uruchomić skrypt, należy raz nacisnąć jedną z instrukcji z bloku. Zamiast tego, lepiej dodać na początek bloku instrukcji klocek Kiedy kliknięto [zielona flaga]. Po dodaniu tej instrukcji można uruchamiać programy za pomocą ikony zielonej flagi w prawym górnym rogu okna.

Niektóre klocki, na przykład większość w zakładkach Czujniki i Wyrażenia wykorzystuje się jako argumenty innych klocek. Mają one inny kształt – są prostokątami z zaokrąglonymi rogami lub sześciokątami. Klocki z zaokrąglonymi rogami mogą przekazywać różne wartości – liczby, napisy, listy; natomiast klocki z sześcioma kątami zwracają wartości logiczne – prawda/fałsz.

Instrukcje sterujące wykonywaniem programu zostały oznaczone odrębnym kolorem. Umożliwiają uruchomienie i zatrzymanie programu, wykonanie pętli, realizację wyrażeń sterujących. W przypadku instrukcji obejmujących swoim zasięgiem blok programu, zastosowano bardzo zgrabne obejmowanie bloku instrukcji klamrą, co jest bardzo intuicyjne a jednocześnie wprowadza przyzwyczajenia niezbędne do programowania w językach wykorzystujących bloki instrukcji. Instrukcje sterujące umożliwiają wykrywanie akcji użytkownika, takich jak naciśnięcie klawisza lub operacje za pomocą myszki. Do ciekawych elementów należy możliwość nadawania i odbierania komunikatów, co umożliwia wymianę informacji pomiędzy obiektami i synchronizację operacji.

Zmienne w Scratch'u są obsługiwane przez osobną grupę instrukcji. Użytkownik ma możliwość tworzenia zmiennych i tablic, jednocześnie generowane są klocki służące do wykonywania podstawowych operacji na zmiennych i tablicach, takich jak ustawianie wartości czy jej zmiana. Zmienne i tablice mogą być utworzone jako zmienne lokalne o zasięgu pojedynczego duszka, lub jako globalne, dostępne z wszystkich elementów aplikacji. Zaawansowani użytkownicy Scratch'a mogą korzystać także ze zmiennych chmurowych, przechowywanych na serwerze Scratch'a i dostępnych dla wszystkich instancji aplikacji. Status zaawansowanego użytkownika uzyskuje się po pewnym czasie, który zależy od aktywności na portalu Scratch'a zarówno w zakresie tworzenia nowych projektów, jak też udzielania się na forach dyskusyjnych.

Podstawowym elementem wykorzystywanym w programie Scratch jest duszek (ang. *Sprite*), który posiada wygląd (kostium) i może wykonywać różne operacje zdefiniowane przez skrypty. W programie można wykorzystać dużą liczbę duszków, jednak zarządzanie nimi może być kłopotliwe i naukę warto rozpocząć od jednej instancji. Duszek jest odpowiednikiem obiektu w C++ czy Javie. Specjalnym rodzajem duszka jest plansza, na której mogą poruszać się inne duszki, ale może zawierać jedynie zmienne globalne. Duszek może zostać pobrany z biblioteki gotowych obiektów dostępnej w dystrybucji programu lub też może zostać samodzielnie narysowany za pomocą prostego edytora przypominającego typowe narzędzia graficzne, takie jak Paint. W przypadku modyfikacji planszy, w analogiczny sposób można przygotować tło.

Jedną z podstawowych własności duszków jest możliwość poruszania się. Scratch wykorzystuje układ współrzędnych stosowany w większości tego typu rozwiązań, jednak oferuje polecenia umożliwiające poruszanie się w określonym kierunku w stosunku do aktualnej pozycji. Dzięki temu możliwe jest poruszanie duszkiem praktycznie bez znajomości układu współrzędnych, jedynie w oparciu o naturalną intuicję i znajomość kilku poleceń.

Korzystanie z programu Scratch nie wymaga od uczniów biegłego posługiwania się klawiaturą i pisania złożonych komend. Polecenia wybierane są graficznie i umieszczane w obszarze skryptu, z czym radzą sobie nawet najmłodsze dzieci. Różne kolory i kształty poleceń oraz charakterystyczne rozmieszczenie umożliwiają łatwe odnalezienie komend nawet przez dzieci nie potrafiące biegle czytać.

### 3. Przykładowe projekty

Jednym z pierwszych zadań, które może być zrealizowane z dziećmi jest „Pokój z przeszkodą”, wprowadzający w poruszanie się duszków i reagowanie na spotkaniu z określonym przez kolor polem.

Kolejne zadanie to pogoń za piłką, w którym uczniowie uczą się jak sterować ruchem duszka za pomocą myszki i jak kierować ruchem tak, by uwzględnić położenie innych obiektów. W zadaniu tym pokazano także, jak wykrywać sytuacje, w których jeden duszek dotyka innego.

Bardziej zaawansowanym zadaniem realizowanym przez uczniów w klasach 1-3 jest utworzenie dialogu między postaciami, synchronizacja wydarzeń i wykonywanie akcji następujących po sobie. W zadaniu tym dzieci uczyły się także sterować duszkiem za pomocą klawiszy oraz wykonywać akcje następujące podczas dotykania się duszków.

Na zakończenie każdego zadania uczniowie udostępniali swój projekt na serwerze korzystając ze specjalnie utworzonego konta (wspólnego dla wszystkich uczestników). Dzięki temu mogli wzajemnie obejrzeć efekty pracy, a po powrocie

do domu zaprezentować je rodzicom. Dzieci identyfikowały swój projekt za pomocą imienia i dodatkowego opisu, nie zalecano użycia nazwisk czy innych danych jednoznacznie identyfikujących uczniów.

#### 4. Scratch na zajęciach Uniwersytetu Dziecięcego

W roku szkolnym 2010/11 Fundacja Amicus Universitatis Nicolai Copernici we współpracy z czterema wydziałami UMK (Chemii, Biologii i Nauk o Ziemi, Matematyki i Informatyki oraz Fizyki, Astronomii i Informatyki Stosowanej) uruchomiła zajęcia dla dzieci pod nazwą Uniwersytet Dziecięcy, adresowane do uczniów z klas 1-3 oraz 4-6 szkół podstawowych. Uczestnicy podzieleni byli na 30. osobowe grupy wiekowe, w przypadku zajęć informatycznych zajęcia odbywały się w grupach dostosowanych do wielkości pracowni (15 stanowisk). Na każdym z wydziałów zajęcia miały na celu zachęcenie dzieci do zainteresowania się naukami ścisłymi. Zajęcia odbywały się w soboty, w pierwszej edycji trwały 90 minut, w drugiej edycji czas zajęć zwiększono do 120 minut. Czas trwania zajęć został zaproponowany przez organizatorów i w przypadku informatyki wymagał bardzo dokładnego planowania przebiegu zajęć, by nie znużyć dzieci, zwłaszcza najmłodsze.

Zajęcia na Wydziale Matematyki i Informatyki miały na celu naukę programowania przez zabawę. Jedno ze spotkań było przeznaczone na budowę i programowanie robotów LEGO, pozostałe wprowadzały uczniów w świat Scratch'a, w którym wystarczy tylko odrobina wyobraźni, by stworzyć program, animację czy grę. Zajęcia były dostosowane do wieku uczestników, w przypadku młodszych dzieci zakres wykorzystywanych instrukcji był ograniczony do podstawowych operacji związanych z ruchem, wyglądem i animacjami, w starszych grupach w większym stopniu były wykorzystywane instrukcje sterujące i zmienne. Zajęcia były prowadzone w sposób praktyczny, polegający na samodzielnym tworzeniu aplikacji przez uczniów. Prowadzący stymulowali uczniów do samodzielnego poszukiwania rozwiązań i końcowy efekt pracy zależał od pomysłów pojawiających się w trakcie zajęć. Efekty pracy na każdych zajęciach były udostępniane *on-line* z wykorzystaniem narzędzi oferowanych przez Scratch'a, co umożliwiało wzajemne zapoznanie się z efektami pracy i pochwalenie się nimi w domu.

Zajęcia prowadzone były przez dwóch prowadzących a każde dziecko pracowało samodzielnie przy komputerze. Spośród prowadzących jedna osoba miała doświadczenie pedagogiczne, a druga zapewniała zaawansowane wsparcie w zakresie programowania (informatyczne).

Ponieważ w drugiej edycji w każdej grupie było kilka osób, które uczestniczyły w poprzednim roku w zajęciach Uniwersytetu Dziecięcego, program był modyfikowany tak, by takie osoby nie nudziły się i miały możliwość poszerzenia swoich umiejętności.

W roku szkolnym 2011/12 (druga edycja) zajęcia ze Scratcha odbywające się w ramach Uniwersytetu Dziecięcego były podzielone na dwa 120 minutowe spotkania. Pierwsze zapoznawało uczniów z programem Scratch i podstawowymi operacjami, jakie można wykonać i odbywały się według podobnego dla wszystkich scenariusza. Ponieważ każdy z uczniów miał do dyspozycji komputer, nauka odbywała się w formie samodzielnego wykonywania zadań stawianych przez prowadzących. Typowe zadanie składało się z kilku instrukcji a jego realizacja trwała kilka minut. Konkretnie pomysły czy zadania były często formułowane przez dzieci na zasadzie zgłaszanego zapotrzebowania na uzyskanie określonego efektu. Sposób realizacji był przedstawiany na ekranie i dostępny do czasu przejścia do następnego zadania. Osoby prowadzące zajęcia na bieżąco pomagały uczniom w realizacji poleceń a w przypadki dzieci radzących sobie szybciej od innych proponowały wykonywanie zadań dodatkowych.

W trakcie drugich zajęć dzieci z klas 1-3 szkoły podstawowej tworzyły animację, w której występowały dwie postacie i przeprowadzały krótką rozmowę, natomiast dzieci starsze (klasy 4-6 szkoły podstawowej) tworzyły własną wersję gry Tetris.

W pierwszej edycji zajęcia dla dzieci młodszych i starszych przebiegały według podobnego scenariusza: pierwsze zajęcia dotyczyły zapoznania z programem i różnym sposobom poruszania się duszka. Na kolejnych zajęciach były tworzone gry. Pierwszą z nich był Labirynt, w którym użytkownik porusza się duszkiem po planszy do wyznaczonego pola zbierając po drodze nagrody. Kolejną, realizowaną tylko w grupie dzieci starszych była gra Armaty polegająca na strzelaniu do celu.

W przypadku zajęć z dziećmi młodszymi, w miarę możliwości dokonywano podziału na grupy uwzględniając wiek, co nieco zmniejszało różnice w umiejętnościach i przygotowaniu. W przypadku zajęć trwających 120 minut pod koniec pierwszej godziny następowało przejście do wykonywania zadań związanych z rysowaniem, co pozwalało na zmianę charakteru pracy i utrzymanie koncentracji uczniów do końca zajęć.

Po dwóch edycjach, prawdopodobnie ze względów finansowych, Fundacja Amicus zrezygnowała z zajęć nauki programowania na rzecz nowych technologii (takich jak edycja grafiki na tabletach). W ostatnim czasie zajęcia z programowania w języku Scratch ponownie wróciły do programu Uniwersytetu Dziecięcego, jednak są realizowane przez jednego prowadzącego. Program i tematyka zajęć jest podobna jak w pierwszych dwóch edycjach.

## 5. Lekcje informatyki w ZS8

W Zespole Szkół nr 8 w Toruniu nauka programowania na lekcjach zajęć komputerowych po raz pierwszy była realizowana w roku szkolnym 2013/14 z uczniami klasy trzeciej. Wprowadzeniem do tematu była pierwsza edycja Godziny

Kodowania [3]. Uczniowie w trakcie zajęć rozwiązywali zadania *on-line* polegające głównie na poruszaniu obiektami na ekranie w celu dotarcia do wyznaczonego celu. Na kolejnych lekcjach uczniowie korzystali z oprogramowania Scratch poruszając duszkiem i poznając podstawy programowania – analogicznie jak w przypadku zajęć na Uniwersytecie Dziecięcym. Następnie przygotowywali krótką animację – historyjkę, w której wykorzystywane były polecenia służące do poruszania się duszków i prowadzenie dialogów.

W kolejnych latach (rok szkolny 2014/5 i 2015/6) nauka programowania była realizowana w klasach piątych i szóstych szkoły podstawowej. Podobnie jak w poprzednim roku szkolnym, wprowadzeniem były kolejne edycje Godziny Kodowania, po których następowały lekcje z wykorzystaniem środowiska Scratch. Uczniowie realizowali typowe projekty, takie jak Labirynt, Wyścigi czy gry oparte na śledzeniu wskaźnika myszy (Pokój z przeszkodą).

## 6. Wnioski praktyczne

### Zajęcia pozalekcyjne

Rezultaty uzyskiwane przez dzieci uczestniczące w Uniwersytecie Dziecięcym pokazują, że programowanie nie musi być trudne i nie jest skierowane jedynie do chłopców. Dziewczynki równie dobrze, a czasem nawet lepiej, radziły sobie z tworzeniem programów, w zasadzie nie zdarzały się sytuacje, w których uczniowie nie realizowali stawianych im zadań.

Zaskoczeniem dla prowadzących zajęcia były występujące braki w podstawowych umiejętnościach korzystania z komputera. Posługiwanie się myszką nie stanowiło generalnie problemu, chociaż były dzieci, które początkowo radziły sobie z tym mało pewnie i ich umiejętności ograniczały się do klikania w obrazki na ekranie, a przesuwanie obiektów stanowiło już problem. Nagminnie pojawiały się problemy z wprowadzaniem danych z klawiatury. Uczniowie nie wiedzieli jak wprowadzić znaki uzyskiwanych przez kombinację klawiszy (polskie litery, wykrzyknik, znak zapytania). U najmłodszych dzieci występowały trudności z odczytywaniem tekstu znajdującego się na klockach z poleceniami i wybór właściwego często odbywał się poprzez dopasowanie kształtu, położenia lub pozycji w grupie.

Zaobserwowane problemy wynikają z poziomu zajęć komputerowych w klasach 1-3 i niedostosowaniem praktyki szkolnej, programów, podręczników i oprogramowania do potrzeb i umiejętności uczniów. Scratch, jak pokazuje przykład zajęć realizowanych w ramach Uniwersytetu Dziecięcego jest bardzo dobrą wyborem do realizacji założeń podstawy programowej i wiele więcej. Należy podkreślić, że grupa dzieci uczestniczących w zajęciach była duża i reprezentatywna, w każdej grupie były osoby nadpobudliwe, często w stopniu znacznym czy też posiadające inne specyficzne cechy, z którymi musieli zmagać się prowadzący.



Z punktu widzenia organizacji zajęć, dwóch prowadzących w grupie 15 osobowej było sytuacją komfortową i raczej trudną do osiągnięcia w realiach szkolnych. Dzięki temu na każdych zajęciach udało się w pełni zrealizować program i zapewnić indywidualne podejście do uczniów, co było niezbędne w przypadku mocno zróżnicowanych grup. W każdej parze prowadzących zajęcia była osoba posiadająca przygotowanie i praktykę w pracy z dziećmi z klas 1-3, co było kluczowe dla powodzenia przedsięwzięcia a jednocześnie było okazją do zdobycia doświadczenia w tym zakresie przez doktorantów informatyki, którzy współprowadzili zajęcia.

## Lekcje informatyki w szkole

Nauka programowania w ramach zajęć komputerowych w szkole podstawowej znacząco różni się od zajęć pozalekcyjnych. Podstawowa różnica wynika z motywacji dzieci: w przypadku zajęć pozalekcyjnych mamy do czynienia z dziećmi zainteresowanymi zajęciami (poza nielicznymi wyjątkami), podczas gdy zajęcia w szkole obejmują wszystkich uczniów. Zajęcia lekcyjne są krótkie (45 minut), co nie pozwala na realizację bardziej złożonych projektów. Zajęcia odbywające się w cyklu tygodniowym powodują trudności z kontynuowaniem projektów pomimo zapisywania ich na serwerze Scratch'a. Najczęściej uczniowie tłumaczą się, że ich projekt zniknął (*ktoś mi go skasował*), nie pamiętają nazwy konta i hasła lub nie mogą znaleźć projektu. Dodatkowo uczniowie mają duże problemy z przypomnieniem sobie umiejętności poznanych na poprzednich lekcjach.

Po początkowej fascynacji programowaniem bardzo szybko pojawia się zniechęcenie. Uczniowie tłumaczą się, że *programowanie jest trudne, bo trzeba myśleć*. Po kilku lekcjach poświęconych programowaniu należy przejść do realizowania innych zagadnień programowych, by nie doprowadzić do całkowitego zniechęcenia. To zalecenie nie jest specyficzne dla nauki programowania i dotyczy wszystkich zagadnień realizowanych w szkole na zajęciach komputerowych.

Istotną obserwacją jest to, że uczniowie rozpoczynający naukę programowania w klasie szóstej radzą sobie gorzej i szybciej się zniechęcają niż ci, którzy rozpoczęli naukę programowania w klasie czwartej czy piątej. Wydaje się, że naukę programowania w oparciu o środowiska programowania wizualnego, takie jak Scratch należy rozpoczynać w trzeciej, najpóźniej w czwartej klasie. Odsuwanie tego momentu do starszych klas utrudnia zainteresowanie uczniów programowaniem.

Dużym problemem są warunki prowadzenia zajęć informatycznych w szkole podstawowej. W klasach 1-3 zajęcia te są prowadzone w dużych grupach (cała klasa, czyli 20-25 osób), w pracowniach, w których nie ma możliwości zapewnienia komputera dla każdego dziecka. W efekcie z komputera często korzystają 2 osoby, a umiejętności praktyczne nabywa najczęściej tylko jedna. W przypadku tak dużych grup jeden nauczyciel musi pomagać wszystkim uczniom, co zmniejsza intensywność

ność zajęć w porównaniu z zajęciami pozaszkolnymi prowadzonymi generalnie w mniejszych grupach.

W klasach 4-6 zajęcia komputerowe odbywają się już z podziałem na grupy, co umożliwia realizację zajęć w modelu jeden uczeń – jeden komputer. Nadal jednak nauczyciel musi pomagać poszczególnym uczniom, ponieważ mają oni duże trudności z obserwowaniem przykładów prezentowanych przez nauczyciela na ekranie i ich samodzielnym odtworzeniem. Wydaje się, że jest to związane z percepcją uczniów wykształconą w oparciu o filmy i szybko zmieniające się obrazy, z trudnościami w koncentracji i generalnie brakiem umiejętności reagowania na docierające z różnych stron bodźce.

Lekcje z programowania z wykorzystaniem środowiska Scratch są okazją do realizacji innych zagadnień wymaganych w podstawie programowej. Uczniowie korzystają z serwisu internetowego dedykowanego dla nich, przechodzą przez proces zakładania konta w serwisie i korzystają z poczty elektronicznej. Zapoznają się z kwestiami związanymi z udostępnianiem treści w Internecie, czy podstawami ochrony własności intelektualnej (remiksowanie projektów).

Należy podkreślić, że Scratch został dostrzeżony przez autorów podręczników do zajęć komputerowych. W szczególności podręczniki dostępne *on-line* [2] zawierają omówienie przykładowych projektów do realizacji na zajęciach. Uczniowie z tych pomocy chętnie korzystają, co ułatwia prowadzenie zajęć.

## 7. Podsumowanie

Nauka informatyki już w pierwszych klasach szkoły podstawowej jest niewątpliwie celowa i nie musi ograniczać się do rysowania i wykorzystywania edytorów tekstu. Zajęcia polegające na nauce programowania w języku Scratch prowadzone w ramach Uniwersytetu Dziecięcego, czy jako element zajęć komputerowych w szkole masowej pokazują, że uczniowie, odpowiednio zachęceni, chcą, mogą i potrafią programować wykonując przy tym wiele innych działań, takich jak rysowanie czy wprowadzanie tekstu, a także korzystanie z dedykowanych serwisów internetowych.

Korzystając ze Scratch'a dzieci mimowolnie doskonalą umiejętność posługiwania się komputerem i Internetem, uczą się wprowadzania tekstów z klawiatury, posługują się przeglądarką internetową i korzystają z dedykowanych serwisów internetowych. Co więcej, zadania te realizowane w sposób naturalny, umożliwiają wprowadzenie do korzystania z komputera w dorosły sposób, a nie poprzez pryzmat dedykowanych, często infantylnych nakładek czy programów.

Dzięki programowi Scratch, przez zabawę można wpoić uczniom założenia programowania strukturalnego i przedstawić działanie podstawowych konstrukcji, jakimi są instrukcje warunkowe i pętle, których zrozumienie nawet przez studentów

uczących się programowania nie zawsze jest łatwe. Scratch umożliwia również wprowadzenie kluczowych elementów programowania obiektowego. Konkludując, Scratch jest bardzo dobrym narzędziem służącym realizacji założeń aktualnej i proponowanej podstawy programowej.

Wydaje się, że aktualne wykorzystanie Scratch'a nie wyczerpuje dostępnych możliwości. Serwer Scratch'a oferuje fora dyskusyjne poświęcone programowaniu, w tym forum w języku polskim. Korzystają z niego głównie uczniowie, aczkolwiek nie za wielu, brakuje na forum nauczycieli czy osób mogących służyć pomocą merytoryczną.

Duże potrzeby występuje w zakresie dostępnego wsparcia dla nauczycieli. Brak jest szerokiego i dostępnego systemu szkoleń, czy to stacjonarnych czy realizowanych *on-line*. Należy podkreślić, że jednorazowe szkolenia są niewystarczające, nauczyciele decydują się na wprowadzenie nauki programowania dopiero po kilku spotkaniach ze Scratch'em, kiedy są już przekonani o wystarczającej znajomości tematu. Dodatkowo, większość dostępnych scenariuszy lekcji dotyczy standardowych projektów, brakuje w nich podejścia motywującego kreatywność, własną inwencję czy projekty interdyscyplinarne, łączące tematy omawiane na różnych przedmiotach.

Z punktu widzenia organizacji zajęć, dwóch prowadzących w grupie 15 osobowej (Uniwersytet Dziecięcy) było sytuacją komfortową i raczej trudną do osiągnięcia w realiach szkolnych. Optymalna realizacja zajęć z programowania w szkole wymaga pracy z podziałem na grupy i realizację zajęć w odpowiednio wyposażonych pracowniach zapewniających uczniom indywidualną pracę z komputerem. Należy też zastanowić się nad blokowaniem zajęć komputerowych (np. 2 godziny tygodniowo przez semestr) co może zapewnić lepsze warunki do realizacji bardziej złożonych projektów.

## Literatura

1. Bała P., Nowicki M., Scratch – środowisko do nauki programowania., *Informatyka w Edukacji*, materiały z warsztatów VII Konferencji IwE 2010, Anna Beata Kwiatkowska (red.), Wydawnictwo Naukowe UMK, 2011, s. 33-40.
2. E-podręczniki <http://www.epodreczniki.pl>
3. Godzina kodowania <http://godzinakodowania.pl>
4. Propozycja zmian w Podstawie programowej z informatyki (14 grudnia 2015): <https://www.oeiizk.waw.pl/joomla/images/pliki/men/2016ppki.pdf>
5. Scratch: <http://scratch.mit.edu/>