

PROGRAMUJEMY MOBILNIE

AGNIESZKA BOROWIECKA



Świat rozwija się tak dynamicznie, że coraz częściej zaczynamy się zastanawiać, jakie jest nasze miejsce w otaczającej nas rzeczywistości. Czego powinniśmy się uczyć? Jakie umiejętności rozwijać? Jakie narzędzia wykorzystywać? Warto sobie uświadomić, że umiejętności cyfrowe, zarezerwowane jeszcze niedawno tylko dla wtajemniczonych, stają się naszą codziennością. Załatwienie sprawy w urzędzie lub banku wymaga od nas posługiwania się komputerem lub urządzeniem mobilnym. Zakupy robimy przez internet. Z rodziną i znajomymi kontaktujemy się, korzystając z komunikatora. Możemy uczyć się zdalnie, korzystając z zasobów internetowych lub spotkań synchronicznych z nauczycielem. W pewnych sytuacjach dopuszczalne jest zdawanie egzaminu lub praca na odległość. Nie oszukujmy się – nawet zwykłe urządzenia gospodarstwa domowego musimy zaprogramować, by działały prawidłowo. Jak to zmienia sytuację naszych uczniów? Może nie każdy z nich będzie zawodowym programistą, ale każdy powinien wiedzieć, na czym polega programowanie, co to jest algorytm, jak myśleć logicznie.

Czy można uczyć się mobilnie? Na pewno tak!
Czy można mobilnie uczyć się programować?
Spróbujmy wspólnie odpowiedzieć na to pytanie...

PROGRAMUJEMY MOBILNIE

ZANIM ZACZNIEMY PROGRAMOWAĆ

Programowanie wydaje się trudne, dodatkowo niektórym może kojarzyć się ze stojącym w ciasnym pomieszczeniu komputerem, przy którym siedzi zapatrzony w ekran programista i wstukuje niczym tajemne zaklęcia kolejne linijki kodu. Jednak jest to obraz z przeszłości. Dziś prawie każdy z nas nosi komputer ze sobą – w kieszeni, w torebce, w plecaku... Często jesteśmy tego nieświadomi, jednak wystarczy porównać możliwości współczesnych smartfonów z możliwościami pierwszych komputerów stacjonarnych, które pojawiały się w naszych szkołach. Jeszcze nie tak dawno wielkość ekranu w telefonach komórkowych była niewielka, brakowało pamięci na uruchamianie bardziej rozbudowanych programów, mogliśmy co najwyżej zagrać w grę Snake lub Tetris. Tablety miały trochę większe możliwości, ale próba czytania za ich pomocą dłuższego tekstu bardzo męczyła oczy. A dziś... Większość z nas nie wykorzystuje nawet ułamka możliwości używanych przez siebie urządzeń mobilnych, nie zdaje sobie także sprawy z zalet użycia ich w nauczaniu. Zresztą mobilność ma wiele twarzy.

Raspberry Pi to w pełni funkcjonalny miniaturowy komputer, który został wymyślony, by wspierać naukę podstaw informatyki. Mimo niewielkich rozmiarów wyposażony jest – jak każdy PC – w procesor, pamięć RAM oraz liczne peryferia. Elementami Raspberry Pi są łącza USB, gniazdo sieciowe RJ45, złącze HDMI, dodano do niego także łączność bezprzewodową. Dzięki temu może pracować jak normalny komputer po połączeniu z monitorem, myszką i klawiaturą; możliwe jest także podłączenie zdalne. Wśród zastosowań popularnej „malinki” jest zbieranie i monitorowanie danych w ramach internetu rzeczy.

Ogromną popularnością cieszą się różne urządzenia pozwalające łączyć naukę elektroniki i programowania, takie jak Arduino. Mikrokontroler ten posiada 14 uniwersalnych pinów GPIO (wejść/wyjść) i 6 wejść analogowych, które pozwalają budować ciekawe programowalne urządzenia elektroniczne.

BBC micro:bit jest wykorzystywany do promowania nauki elektroniki oraz programowania. Cechą charakterystyczną micro:bita jest wyświetlacz z 25 diod ułożonych w matrycę 5x5, który umieszczono na przodzie płytki. Pozwala on wyświetlać symbole, wskazania czujników oraz proste teksty. Oba mikrokontrolery mają niewielkie rozmiary i spokojnie mieszczą się w kieszeni. Powstały na potrzeby nauczania: Arduino jako pomoc dydaktyczna dla włoskich studentów, zaś micro:bit dla uczniów szkół podstawowych w Wielkiej Brytanii.

Możliwości Arduino czy micro:bita są ogromne, jednak by móc z nich korzystać, musimy zaprogramować ich działanie. Ciekawy przykład eksperymentu związanego z mierzaniem pola magnetycznego za pomocą micro:bita został opisany w numerze 6 kwartalnika „W cyfrowej szkole”¹. Wyniki przeprowadzonych pomiarów są analizowane za pomocą programu w języku Python. W kolejnym artykule w tym samym piśmie Witold Kranas opisuje, jak za pomocą bloków MakeCode zasymulować na mikro:bitcie rzuty monetą lub kostką².

Prawie każdy uprawiając sport, nie tylko wyczerpująco, korzysta z różnych urządzeń pomiarowych, takich jak inteligentne zegarki, krokomierze, smartbandy. Charakteryzują się one niewielkimi rozmiarami i coraz większymi możliwościami. Dzięki nim możemy nie tylko monitorować swoje treningi, śledzić osiągnięcia, porównywać wyniki. Jak wielką rolę odgrywają w życiu wielu z nas można było się przekonać nie tak dawno temu, gdy podczas wakacji przestały działać serwery firmy Garmin. Podczas kolejnych dni na forach internetowych trwały spekulacje na temat przyczyn i skutków awarii, wiele osób martwiło się o zaplanowane treningi i rejestr wcześniejszych aktywności.

¹ Biszczyk J., Samulski A. *Eksperymenty na informatyce – sposób na integrację międzyprzedmiotową*, „W cyfrowej szkole” nr 1(6)/2020, OEIŁK.

² Kranas W. *Micro:bit gra w kości*, ibidem.

AGNIESZKA BOROWIECKA



Niezależnie od tego, z jakich urządzeń mobilnych będziemy korzystać, jednym z częstych zastosowań będą pomiary danych. Współczesne smartfony i tablety wyposażone są w różnego rodzaju czujniki, takie jak: akcelerometr, żyroskop, magnometr, czujnik światła, czujnik zbliżeniowy, GPS, barometr, termometr. Smartwatche pozwalają gromadzić między innymi dane dotyczące tętna, czasu aktywności, czasu i jakości snu, pokonanego dystansu, a także przebytej trasy, przewyższenia, maksymalnej i minimalnej osiągniętej wysokości. Do płytek programowalnych również możemy podłączać różnego typu czujniki. Zgromadzone dane należy następnie przeanalizować i przetworzyć. Oparcie się o realne wyniki na pewno zwiększy zainteresowanie uczniów. Na przykład zajmując się tabelami w arkuszu, osiągniemy o wiele lepsze efekty, gdy uczniowie do wyciągania wniosków będą wykorzystywali własnoręcznie zebrane i przygotowane dane.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<gpx version="1.1" creator="TomTom.2016 with Barometer"
xsi:schemaLocation="http://www.topografix.com/GPX/1/1
http://www.topografix.com/GPX/1/1/gpx.xsd" xmlns="http://www.topografix.com/GPX/1/1"
xmlns:gpxtpx="http://www.garmin.com/xmlschemas/TrackPointExtension/v1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <metadata>
    <name>Activity</name>
  </metadata>
  <trk>
    <name>Cycling</name>
    <trkseg>
      <trkpt lat="52.129069" lon="21.056836">
        <ele>104.0</ele>
        <time>2019-08-11T09:51:07.000Z</time>
        <extensions>
          <gpxtpx:TrackPointExtension>
            <gpxtpx:hr>91</gpxtpx:hr>
          </gpxtpx:TrackPointExtension>
        </extensions>
      </trkpt>
      <trkpt lat="52.129055" lon="21.056847">
        <ele>104.0</ele>
        <time>2019-08-11T09:51:08.000Z</time>
        <extensions>
          <gpxtpx:TrackPointExtension>
            <gpxtpx:hr>91</gpxtpx:hr>
          </gpxtpx:TrackPointExtension>
        </extensions>
      </trkpt>
      <trkpt lat="52.129042" lon="21.056864">
        <ele>104.0</ele>
        <time>2019-08-11T09:51:09.000Z</time>
        <extensions>
          <gpxtpx:TrackPointExtension>
            <gpxtpx:hr>91</gpxtpx:hr>
          </gpxtpx:TrackPointExtension>
        </extensions>
      </trkpt>
    </trkseg>
  </trk>
</gpx>
```

ILUSTRACJA 1. Przykładowe dane z urządzenia TomTom Adventurer zapisane w pliku xml

PROGRAMUJEMY MOBILNIE

NAPISAŁEM WŁASNĄ APKĘ!

Znakomitą zachętą do nauki programowania jest możliwość podzielenia się wynikami pracy. Każdy uczeń będzie zachwycony możliwością uruchomienia na komórce czy tablecie przygotowanej przez siebie aplikacji. Okazuje się, że możemy tworzyć w ramach lekcji informatyki interesujące aplikacje, ucząc jednocześnie podstaw algorytmiki, projektowania aplikacji i zasad logicznego myślenia. Oczywiście raczej nie przygotujemy w pełni profesjonalnego i rozbudowanego programu, takiego jak Mapy Google, Spotify czy ScratchJr. Na to po prostu nie starczy nam czasu i środków. Raczej nie będziemy także korzystali z Android Studio do pisania natywnych aplikacji na Androida, ani z Xcode do

tworzenia oprogramowania na iOS. Język Java nadal jest popularnym językiem dla piszących aplikacje na urządzenia mobilne, choć coraz częściej mówi się o jego następcach, takich jak Kotlin czy framework Flutter przeznaczony do budowania natywnych aplikacji mobilnych na Androida i iOSa ze wspólnego kodu. Zamiast tego proponujemy zastosować jedną z kilku możliwości przygotowania mobilnej aplikacji.

Pierwszą z nich jest pisanie programów w środowisku Processing. Jest to język programowania, środowisko i społeczność, które od 2001 roku promują umiejętność programowania wśród zawodów artystycznych. Istnieje specjalna wersja tego środowiska (Processing for Android) dedykowana tworzeniu aplikacji na system Android.



ILUSTRACJA 2. Gra Pong zainstalowana na tablecie (Processing)

Zespół MIT Media Lab³, znany mityśnikom Scratcha, rozwija środowisko App Inventor – znakomite narzędzie pozwalające tworzyć rozbudowane aplikacje i gry na urządzenia mobilne. Zawiera ono graficzny interfejs do modelowania wyglądu aplikacji i obszerny zestaw komponentów; programujemy za pomocą skryptów zbudowanych z bloków. Narzędzie to można z powodzeniem wykorzystać do nauki programowania w starszych klasach szkoły podstawowej.

³ Massachusetts Institute of Technology (MIT) – amerykańska uczelnia niepubliczna



ILUSTRACJA 3. Aplikacja Minigolf wykorzystująca przechyłanie tabletu do sterowania obiektem (p5.js)

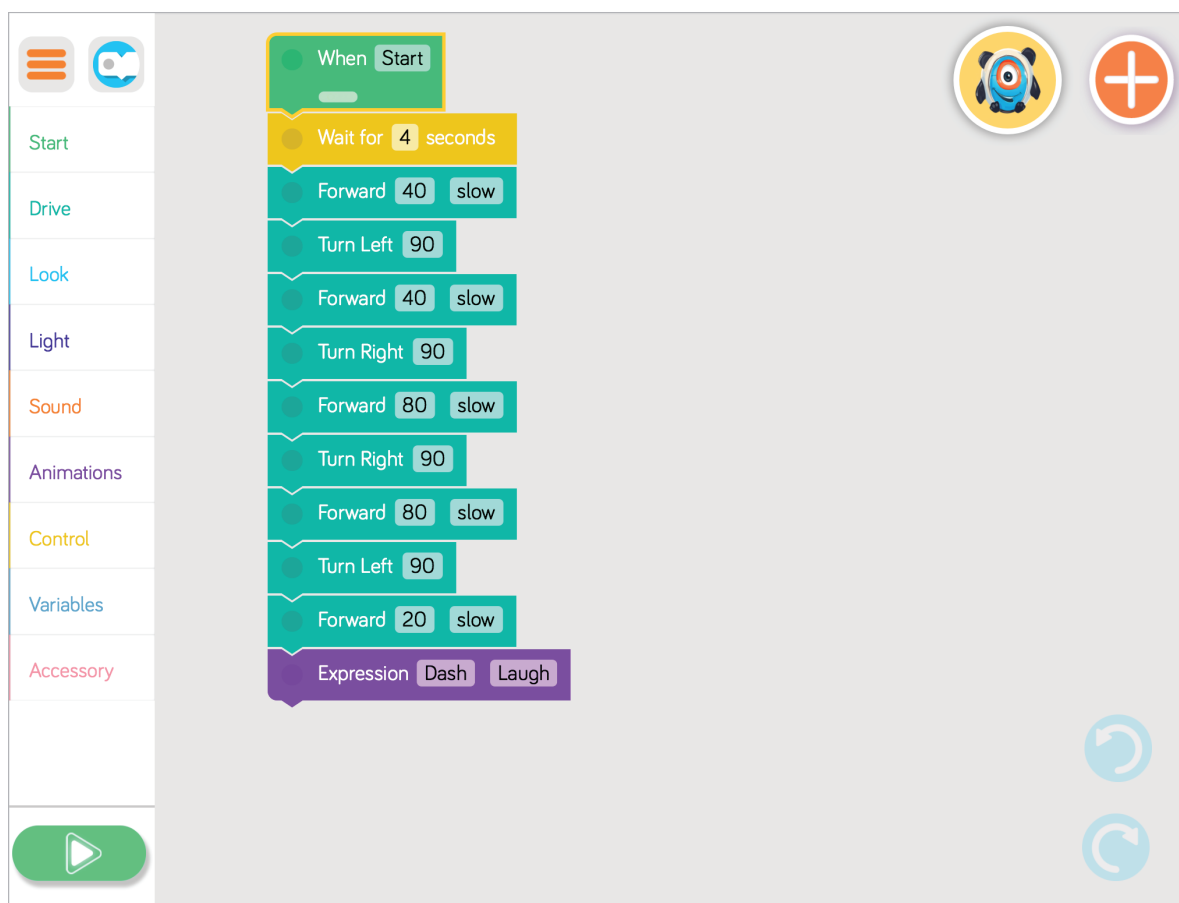
AGNIESZKA BOROWIECKA

Do uruchamiania napisanego przez siebie programu na smartfonie możemy wykorzystać także przeglądarkę internetową. Wystarczy w tym celu wczytać stronę internetową opartą o HTML5, CSS i JavaScript. Do tworzenia elementów interaktywnych na stronie WWW warto wykorzystać p5.js – bibliotekę JavaScript dostarczającą narzędzi zarówno do tworzenia i zarządzania multimediami, jak i do współpracy z użytkownikiem (obsługa DOM, formularze, myszki i klawiatury). Tego typu rozwiązanie jest ciekawe także ze względu na pewną niezależność tworzonego kodu, można go bowiem uruchamiać zarówno na urządzeniach mobilnych, jak i na komputerze stacjonarnym. Dodając pewne funkcjonalności, w prosty sposób możemy także rozbudować tworzone aplikacje np. o odwołanie do czujników takich jak akcelerometr.

Na stronie Programowanie w szkole (<https://programowanie.oeiizk.waw.pl>) w działach *Processing* oraz *JavaScript* można znaleźć przykłady takich zastosowań (gry **Minigolf** oraz **Labirynt**).

MAM TABLET, CZY MOGĘ NA NIM PROGRAMOWAĆ?

Naukę programowania zaczynamy zwykle od języków wizualnych. Wiele z nich jest dostępnych online na stronach internetowych lub w postaci aplikacji instalowanych na tabletach. Interesującym zadaniem dla uczniów może być zaprogramowanie działania robota (np. Dash & Dot, Cue, Ozobot, WeDo 2.0, Lego Mindstorms...) za pomocą skryptu zbudowanego w jednej z takich aplikacji.



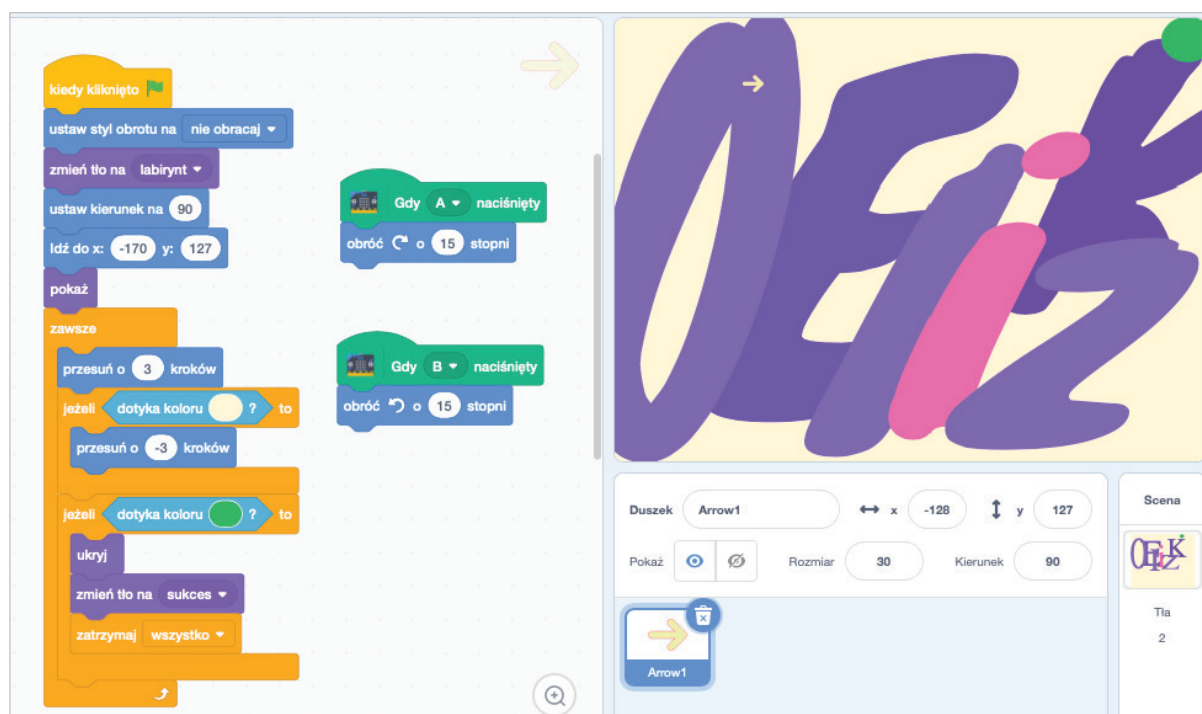
ILUSTRACJA 4. Przykładowy program sterujący robotem Dash

PROGRAMUJEMY MOBILNIE

Dla najmłodszych polecamy program **ScratchJr**. Jest to wizualny język programowania przeznaczony dla dzieci w wieku 5-8 lat, dostępny w postaci aplikacji instalowanej na tabletach lub smartfonach o dostatecznie dużym ekranie. Nie ma odpowiednika tej aplikacji na urządzenia stacjonarne.

Wspominany wcześniej zespół MIT Media Lab przygotował m.in. wersję 3.0 środowiska **Scratch**.

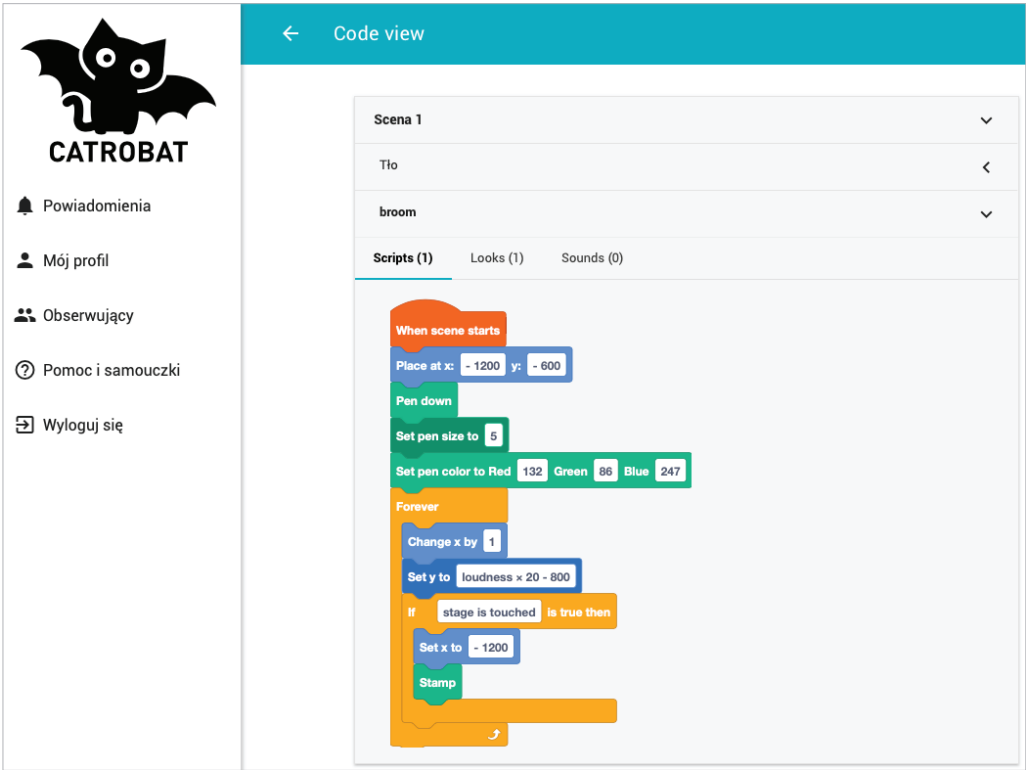
Jest ona oparta o HTML5 i dzięki temu dostępna w przeglądarce, także na urządzeniach mobilnych (poprzednia wersja korzystała z technologii Flash niewspieranej przez te urządzenia). Dzięki temu możemy programować skrypty na tablecie, uruchamiać i testować ich działania, a także prezentować wyniki swojej pracy bez konieczności posiadania komputera stacjonarnego lub laptopa. Jedynym wymaganiem jest dostęp do internetu.



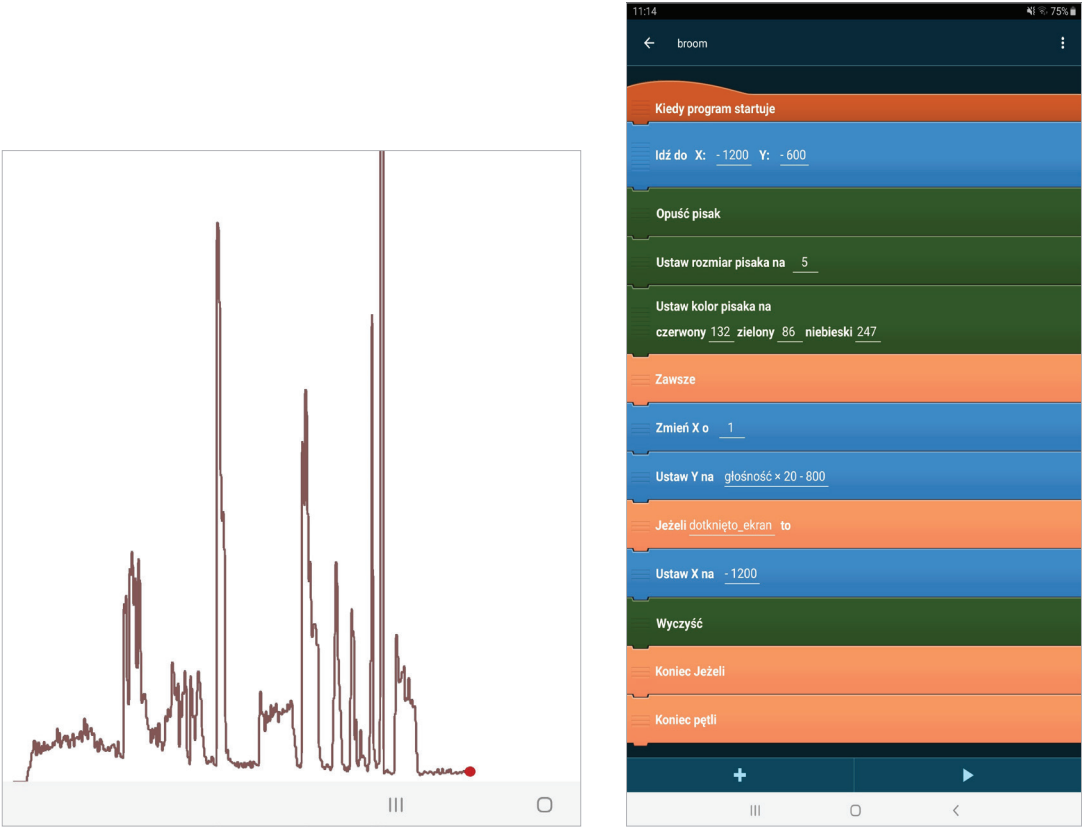
ILUSTRACJA 5. Sterowanie strzałką za pomocą Micro:bita – aplikacja w Scratchu

Pochodną języka Scratch jest Catrobat. **Pocked Code** to aplikacja dostępna na urządzenia z systemem Android i iOS, która pozwala programować w tym języku. Podobnie jak w Scratchu budujemy programy z bloczków, efekt możemy oglądać na tablecie lub smartfonie, a także publikować w ramach serwisu <https://share.catrob.at>. Istnieje możliwość zaimportowania projektu utworzonego w Scratchu do Pocked Code i rozbudowania w tym środowisku. Przygotowany program można skonwertować do pliku *apk* i zainstalować tak jak każdą inną aplikację.

AGNIESZKA BOROWIECKA



ILUSTRACJA 6. Przykładowy kod w języku Catrobot – tworzenie wykresu dźwięku

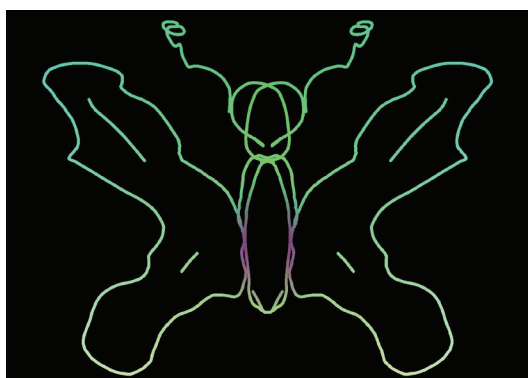
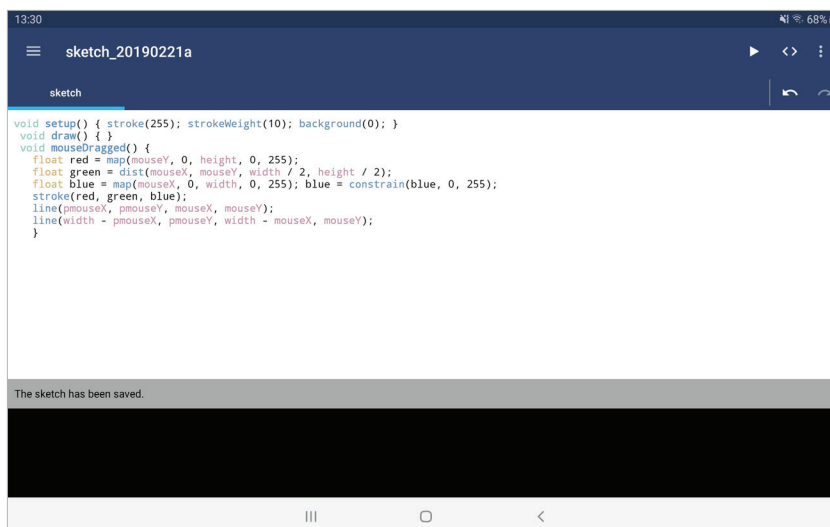


ILUSTRACJA 7. Wynik działania programu i kod widoczny w aplikacji Pocked Code

PROGRAMUJEMY MOBILNIE

Na urządzeniach mobilnych możemy także programować ze starszymi uczniami, korzystając z tekstowych języków programowania. Wśród różnych aplikacji znajdziemy między innymi **APDE – Android Processing IDE**, zintegrowane środowisko umożliwiające pisanie programów w języku Processing bezpośrednio na tablecie. Podczas testowania wprowadzonego kodu przeprowadzana jest jego kompilacja do formatu *apk*, gotowa aplikacja zostaje zainstalowana na urządzeniu i można z niej korzystać jak z innych aplikacji pobieranych ze sklepu Google. Mamy także możliwość definiowania różnych parametrów działania aplikacji, np. orientacji i zablokowania obracania ekranu. Podobna aplikacja o nazwie Processing jest dostępna w Apple Store, jednak za jej pomocą można jedynie testować działanie napisanego programu, nie jest dostępna kompilacja i instalowanie otrzymanej aplikacji na tablecie z systemem iOS.

Pojawiają się stale nowe pozycje, które możemy wykorzystać w pracy z uczniami. Przykładem może być program **Pydroid 3**, pozwalający pisać programy w języku Python. Należy sobie zdawać jednak sprawę z tego, że pisanie programów na urządzeniach mobilnych jest mniej wygodne niż na urządzeniu wyposażonym w klawiaturę i myszkę. Dodatkową niedogodnością może być trudna do wyłączenia automatyczna korekta tekstu zmieniająca wprowadzany przez nas kod. Często wygodniej jest napisać program na urządzeniu stacjonarnym, a następnie skopiować go na urządzenie mobilne i jedynie wykonać niezbędne poprawki.



ILUSTRACJA 8. Wynik działania programu i fragment kodu w aplikacji APDE

AGNIESZKA BOROWIECKA

UCZYMY SIĘ PROGRAMOWANIA

Łatwo zauważyć, że zarówno pisanie programów bezpośrednio na urządzeniach mobilnych, jak i przeznaczonych do zainstalowania na nich, może stanowić prawdziwe wyzwanie. Jednak wykorzystanie tego typu urządzeń do samej nauki programowania jest nie tylko wygodne, ale i godne polecenia. Istnieje wiele aplikacji wprowadzających dzieci i młodzież w świat programowania, w sposób zabawny i interesujący. Możemy zacząć np. od sterowania różnymi obiektami w grach **Code Hour – Think & Learn Code-a-pilar**, **Kodable**, **Lightbot** czy **SpriteBox**. Firma Apple oferuje specjalne środowisko **Swift Playgrounds**, wprowadzające w świat programowania w języku Swift. Zaczynamy od sterowania postacią, budując skrypty z bloków, by przejść do pisania prostych fragmentów kodu przypominających takie, jakie powstałyby podczas programowania w języku Swift pełnoprawnej aplikacji do zainstalowania na urządzenia z systemem iOS.

Istnieje także cały szereg profesjonalnych samouczków wprowadzających w świat programowania w wielu językach. Firma SoloLearn udostępnia materiały do nauki programowania w językach Python, C++, Java, JavaScript, PHP, HTML i wielu innych, na stronie internetowej i w postaci aplikacji na urządzenia mobilne. Jest to doskonała propozycja zarówno dla uczniów, jak i dla samych nauczycieli.

PODSUMOWANIE

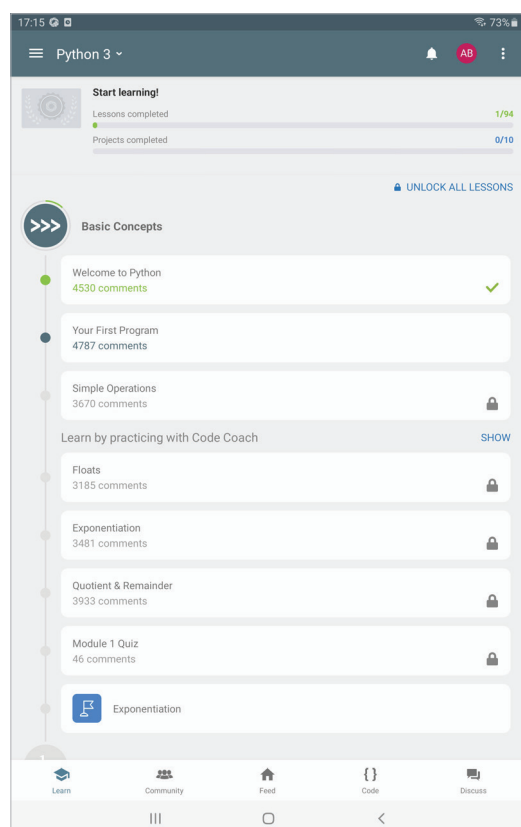
Ciągle wiele szkół zabrania korzystania ze smartfonów w klasie zamiast zastanowić się, w jaki sposób wykorzystać je w nauczaniu. Warto zachęcić naszych uczniów, by urządzeń mobilnych używali nie tylko do zabawy i działań społecznościowych. Zamiast tego możemy pokazać im, że umiejętność tworzenia algorytmów i nauka programowania stanowi więcej niż tylko realizację założeń podstawy programowej. Napisanie działającego programu i uruchomienie go na własnej komórce – każdy z nas krzyknął kiedyś *WOW, udało mi się!* ●

ŹRÓDŁA

1. Biszczuk J., Samulski A. *Eksperymenty na informatyce – sposób na integrację międzyprzedmiotową*, „W cyfrowej szkole” nr 1(6)/2020, OEIiZK.
2. Kranas W., *Micro:bit gra w kości*, „W cyfrowej szkole” nr 1(6)/2020, OEIiZK.
3. <https://programowanie.oeiizk.waw.pl>, dostęp 30.10.2020.

AGNIESZKA BOROWIECKA

nauczyciel konsultant
w Ośrodku Edukacji Informatycznej
i Zastosowań Komputerów w Warszawie



ILUSTRACJA 9. Przykładowy kurs SoloLearn – Python 3