

MIĘDZY BLOKAMI A KODEM PROGRAMU W POSZUKIWANIU ŚRODOWISKA DO UCZENIA PROGRAMOWANIA

Witold Kranas

Ośrodek Edukacji Informatycznej i Zastosowań Komputerów
ul. Nowogrodzka 73, 02-006 Warszawa

witek@oeiizk.waw.pl

Abstract. There are a few environments allowing to build a project using blocks with a possibility to translate it immediately to programming language. Two of them will be introduced with a couple of examples. First is App Lab on code.org using a set of blocks to build a smartphone application which may be easily translated to JavaScript. The second is more widely known Blockly code by Google allowing to change block construction into six different languages (Python, JavaScript, PHP, Lua, Dart, XML). These environments give teachers possibility to teach basics of programming. We will explore this possibility searching the ways to introduce some topics from new computer science core curriculum.

1. Programowanie wizualne vs. tekstowe

W tym roku obchodziliśmy dziesiąte urodziny Scratcha – środowiska, które szturmem zdobyło pole edukacji informatycznej. Pomysł ułatwienia nauki programowania poprzez wizualizację poleceń języka i składanie programu z gotowych bloków nie jest nowy. Już w latach dziewięćdziesiątych poprzedniego wieku był rozpowszechniany w polskich szkołach program ELI Laboratorium Informatyki, w którym można było układać i uruchamiać schematy blokowe algorytmów.

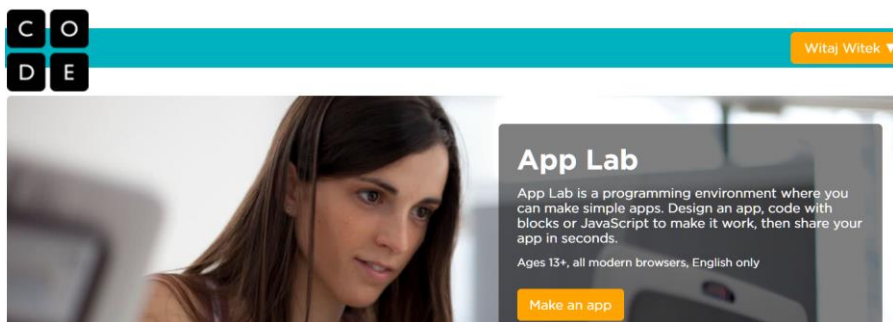
W tych czasach dominującym środowiskiem programistycznym w edukacji był język LOGO, a na wyższym poziomie również Pascal. Dokonujące się dziś przejście na środowiska wizualne spowodowało, że trzeba szukać sposobów na transfer nabytych w tych środowiskach umiejętności programistycznych do środowisk tekstowych, wymaganych na egzaminie maturalnym.

Zaczęły już postawać środowiska łączące budowanie programu z gotowych bloków i programowanie tekstowe. Poniżej zaprezentowane zostaną na przykła-

dach dwa z nich. Pierwsze to App Lab na code.org pozwalające na zbudowanie za pomocą zestawu bloków aplikacji smartfonowej. Ułożone bloki można jednym kliknięciem zamienić na kod programu w JavaScript (i odwrotnie). Drugie, szerzej znane – Blockly kod, umożliwia zamianę skryptu ułożonego z bloków na sześć różnych języków programowania (Python, JavaScript, PHP, Lua, Dart, XML).

2. App Lab – dziecko Godziny kodowania

Od ponad roku code.org instytucja, która zorganizowała akcję Godzina kodowania rozwija własne środowisko programowania wizualnego dedykowane uczniom 13+ o nazwie App Lab.



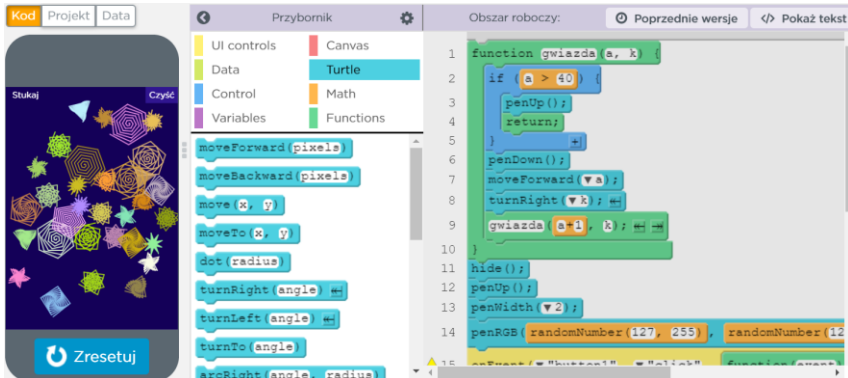
Rysunek 1 Główna strona środowiska AppLab na code.org

Można w nim zarówno programować wizualnie, korzystając z zestawu gotowych bloków, jak i tekstowo w języku JavaScript oraz jednym kliknięciem przechodzić od bloków do kodu i odwrotnie. Środowisko jest nastawione na tworzenie prostych aplikacji na smartfony i tablety. Działa ono wyłącznie w wersji angielskiej. Na głównej stronie jest obfitość materiałów wprowadzających do korzystania z App Lab, utrzymanych w stylu charakterystycznym dla Godziny kodowania. Jest wśród nich kurs wprowadzający do programowania (CSP Unit 3 – Programming), wiele krótkich filmów demonstrujących możliwości środowiska (RESOURCES), próbki gotowych projektów (Challenges) oraz filmy edukacyjne (video library).

Przejdzie do budowania projektu wymaga zalogowania się na konto w code.org, to samo, z którego korzystamy w trakcie pracy z Godziną kodowania.

2.1. Prosty projekt – rysowanie „gwiazdek”

W pierwszym projekcie wykorzystamy głównie zestaw bloków Turtle, pozwalający na korzystanie z grafiki żółwia. Będziemy rysować spiralne gwiazdki w miejscu kliknięcia na ekranie.



Rysunek 2. Projekt „gwiazdki” w środowisku App Lab – widok bloków

Procedura rysująca gwiazdki w wersji JavaScript wygląda następująco:

```
function gwiazda(a, k) {
  if (a > 40) {
    penUp();
    return;
  }
  penDown();
  moveForward(a);
  turnRight(k);
  gwiazda(a+1, k);
}
```

Korzysta ona z grafiki żółwia i rekurencji. Ta możliwość jest jedną z zalet środowiska, gdyż pozwala na wykorzystanie materiałów opracowanych w języku LOGO, który jest znany sporej części nauczycieli informatyki. Pozostała część projektu to obsługa zdarzeń: kliknięcia na ekranie oraz kliknięcia przycisku „Czyść”.

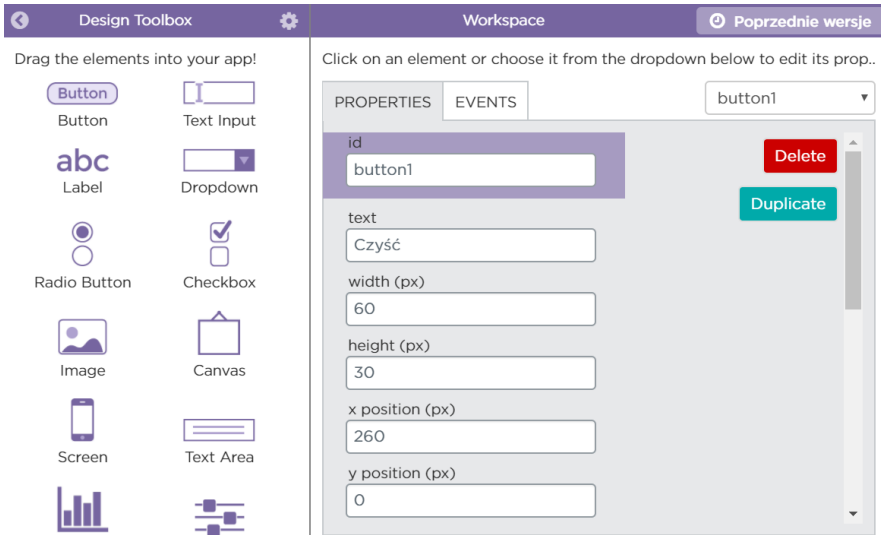
```
hide();
penUp();
penWidth(2);
penRGB(randomNumber(127, 255), randomNumber(127,
  255), randomNumber(0, 255));
onEvent("button1", "click", function(event) {
  moveTo(160, 240);
  penColor("#140058");
  dot(300);
});
onEvent("screen1", "click", function(event) {
```

```

moveTo(event.x, event.y);
gwiazda(1, randomNumber(60, 179));
penRGB(randomNumber(127,255), randomNumber
(127,255), randomNumber(0,255));
});

```

Ważną częścią projektu jest budowanie jego elementów takich, jak przyciski, wprowadzanie danych, etykiety, obrazki, pola tekstowe, itp.). Odbyna się to w zakładce Project. Pierwszy projekt zawiera przycisk „Czyść”.



Rysunek 3. Zakładka Project w środowisku App Lab

Budowane obiekty mogą generować zdarzenia, definiowane w zakładce EVANTS.

2.2. Projekt – algorytm Euklidesa

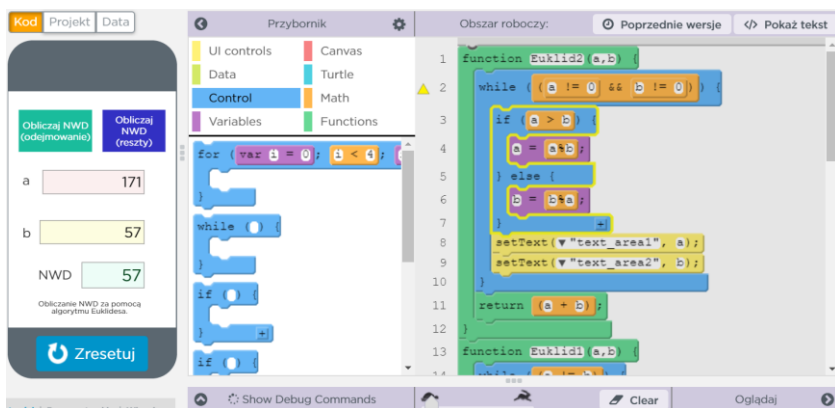
W drugim projekcie wykorzystujemy klasyczny algorytm Euklidesa, znajdujący NWD dwu liczb w wersji z odejmowaniem oraz z resztami z dzielenia. Obie wersje algorytmu są przywoływane w nowej podstawie programowej informatyki w szkole podstawowej (klasy VII, VIII).

Środowisko dysponuje standardowymi pętlami iteracyjnymi: while i for znajdującymi się w niebieskiej grupie Control. Wersje Algorytmu Euklidesa z odejmowaniem i resztami wyglądają następująco (polecenia setText zostały dodane, aby można było zobaczyć wartości zmiennych w kolejnych krokach iteracji):

```
function Euklid1(a,b) {
```

```
while ((a != b)) {  
  if (a > b) {  
    a = a - b;  
  } else {  
    b = b - a;  
  }  
  setText("text_area1", a);  
  setText("text_area2", b);  
}  
return a;}
```

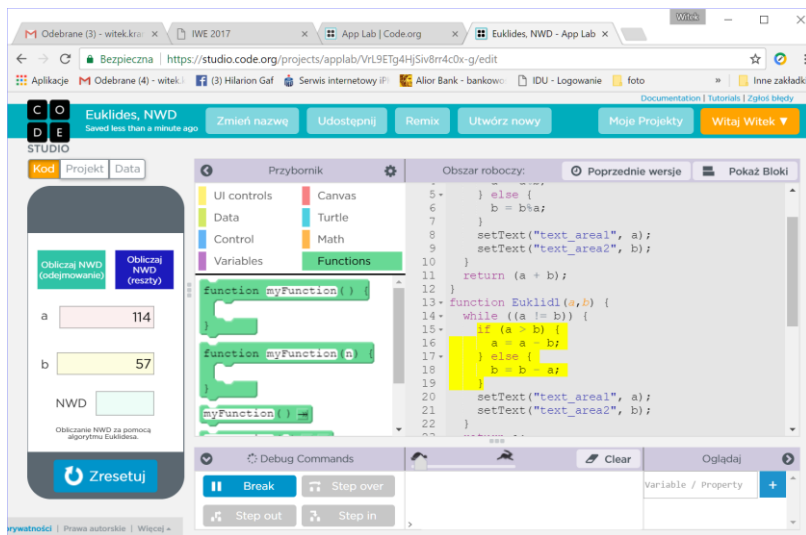
```
function Euklid2(a,b) {  
  while ((a !== 0 && b !== 0)) {  
    if (a > b) {  
      a = a%b;  
    } else {  
      b = b%a;  
    }  
    setText("text_area1", a);  
    setText("text_area2", b);  
  }  
  return (a + b);  
}
```



Rysunek 4. Projekt „Euklides, NWD” w środowisku App Lab, widok bloków

Jedną z istotnych cech środowiska pomagającą w nauczaniu algorytmiki jest możliwość śledzenia programu. Działa ona zarówno w wersji blokowej jak i teksto-

wej. Przed uruchomieniem programu trzeba jednak przestawić szybkość jego wykonywania z „zająca” na „żółwia”.



Rysunek 5. Projekt „Euklides, NWD” w App Lab, widok tekstu, śledzenie

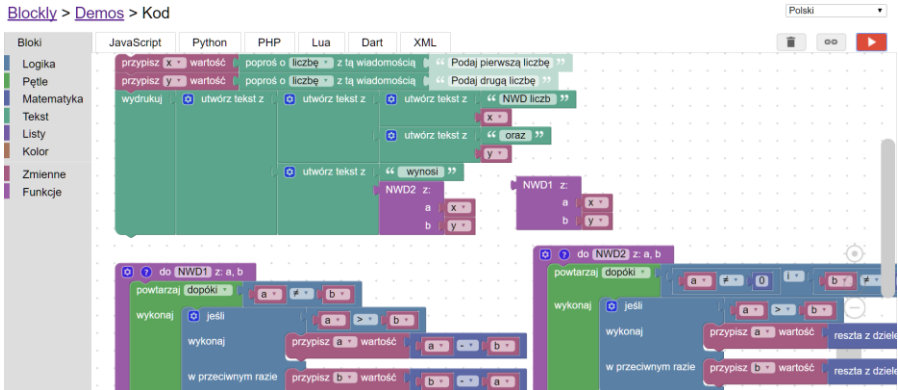
Pozostała część projektu zawiera procedury obsługi kliknięć w przyciski obliczania NWD.

```
var a1 = promptNum("Podaj pierwszą liczbę: ");
setText("text_area1", a1);
var b1 = promptNum("Podaj drugą liczbę: ");
setText("text_area2", b1);
onEvent("button1", "click", function(event) {
  setText("text_area1", a1);
  setText("text_area2", b1);
  setText("naNWD", Euklid2(a1, b1));
});
onEvent("button2", "click", function(event) {
  setText("text_area1", a1);
  setText("text_area2", b1);
  setText("naNWD", Euklid1(a1, b1));
});
```

Aby uruchomić projekt na smartfonie wystarczy przesłać link, odebrać go na smartfonie i stuknąć link. W pełni funkcjonalna aplikacja zostanie otwarta w przeglądarce.

3. Środowisko Blockly kod

Drogi środowisko programowania wizualnego z możliwością zamiany na kod programu jest częścią projektu Google Blockly. Znaczącym utrudnieniem jest brak możliwości zalogowania się. Aby zapamiętać projekt należy zapisać link do niego.



Rysunek 6. Algorytm Euklidesa w środowisku Blockly kod

W repertuarze dostępnych bloków nie ma poleceń grafiki żółwia, za to bloki są spolszczone. Po zbudowaniu projektu można zobaczyć odpowiadający mu kod programu. Tu przykład w JavaScript.

```

var a, b, x, y;
function mathRandomInt(a, b) {
  if (a > b) {
    // Swap a and b to ensure a is smaller.
    var c = a;
    a = b;
    b = c;
  }
  return Math.floor(Math.random() * (b - a + 1) + a);
}
/* Algorytm Euklidesa z odejmowaniem */
function NWD1(a, b) {
  while (a != b) {
    if (a > b) {
      a = a - b;
    } else {
      b = b - a;
    }
  }
  return a;
}

```

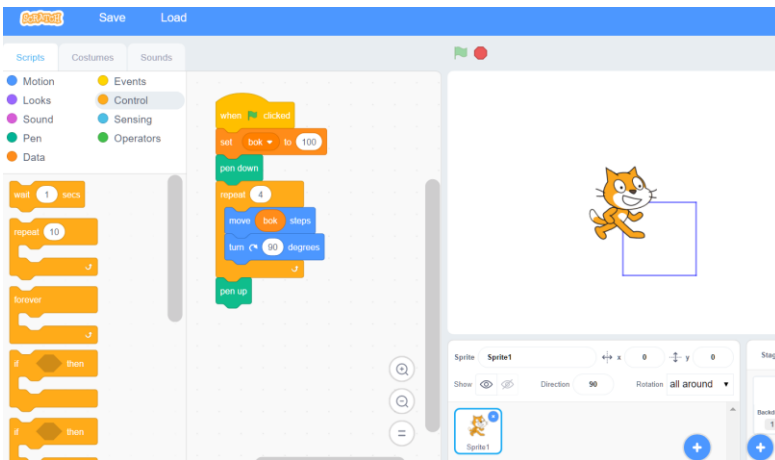
```
/* Algorytm Euklidesa z resztami */  
function NWD2(a, b) {  
    while (a != 0 && b != 0) {  
        if (a > b) {  
            a = a % b;  
  
        } else {  
            b = b % a;  
        }  
    }  
    return a + b;  
}  
mathRandomInt(1, 100);  
x = parseFloat(window.prompt('Podaj pierwszą liczbę'));  
y = parseFloat(window.prompt('Podaj drugą liczbę'));  
window.alert(String(String(String('NWD liczb ') +  
String(x)) + String(String(' oraz ') + String(y))) +  
String(String(' wynosi ') + String(NWD2(x, y))));
```

Jeśli umieścimy ten tekst jako skrypt w kodzie HTML, to będzie on działać. Podobnie z kodem w języku Python – wystarczy przejść do środowiska...

4. Nauczanie programowania

Nowa podstawa programowa kładzie nieco większy nacisk na programowanie i algorytmikę. W klasach VII i VIII wskazane jest wprowadzanie tekstowego środowiska programowania zamiast/obok środowiska wizualnego.

O ile podstawę dla klas IV-VI można w pełni zrealizować wykorzystując środowisko wizualne, takie jak Scratch, to w dwóch ostatnich klasach szkoły podstawowej powinniśmy mieć aplikację umożliwiającą programowanie tekstowe. Dobrze byłoby również, gdyby to środowisko umożliwiało śledzenie przebiegu programu, abyśmy mogli analizować algorytmy wymienione w podstawie.



Rysunek 7. Scratch 3.0 w nowych szatach, wersja testowa

Wszystkie te postulaty spełnia środowisko App Lab. Google Blockly kod pozwala na zamianę bloków na kod programu, ale nie pozwala na pisanie w kodzie, a śledzenie przebiegu trzeba realizować w innym środowisku. Nie bez znaczenia jest też możliwość korzystania w środowisku App Lab z bloków grafiki żółwia. Część nauczycieli przez wiele lat wykorzystywała w nauczaniu język LOGO i w App Lab będą oni mogli wykorzystać przynajmniej część swoich materiałów. Wreszcie możliwość uruchomienia stworzonej aplikacji na smartfonie może dodatkowo motywować uczniów do pracy w tym środowisku.

Przed nami jeszcze jedna zmiana. W 2016, w dziewiąte urodziny Scratcha zapowiedziano nową wersję tego środowiska, która zapewne będzie się nazywać Scratch 3.0. Ma ona mieć nowe szaty – bloki oparte na oogle Blockly i nowy „napęd” – HTML5. Zapowiadanych jest trochę nowych bloków i premiera pod koniec 2017 roku. Niestety nic nie słychać o możliwości zamiany skryptu w Scratchu na kod programu, ale...

Literatura

1. Algorytm Euklidesa w środowisku Blockly kod: <https://blockly-demo.appspot.com/static/demos/code/index.html?lang=pl#onsvvo>, ostatni dostęp maj.2017 roku.
2. Dokumentacja środowiska App Lab: <https://docs.code.org/applab>, ostatni dostęp maj.2017 roku.

3. Projekt „Euklides, NWD” w środowisku App Lab: <https://studio.code.org/projects/applab/VrL9ETg4HjSiv8rr4c0x-g>, ostatni dostęp maj.2017 roku.
4. Projekt „Gwiazdy” w środowisku App Lab: <https://studio.code.org/projects/applab/h4ca4yTVkbPDCHn2x2Ephw>, ostatni dostęp maj.2017 roku.
5. Nowa podstawa programowa z informatyki dla szkoły podstawowej <https://men.gov.pl/wp-content/uploads/2016/11/podstawa-programowa-z-informatyki-szkola-podstawowa.pdf>, ostatni dostęp maj 2017 roku.
6. Strona główna środowiska App Lab: <https://code.org/educate/applab>, ostatni dostęp maj.2017 roku.
7. Strona główna środowiska Blockly kod: <https://blockly-demo.appspot.com/static/demos/code/index.html?lang=pl>, ostatni dostęp maj.2017 roku.
8. Wersja testowa środowiska Scratch 3.0: <https://llk.github.io/scratch-gui/>, ostatni dostęp maj 2017 roku