

PROGRAMOWANIE GRAFIKI I ELEMENTÓW INTERAKTYWNYCH NA STRONY WWW W P5.JS

Agnieszka Borowiecka
Ośrodek Edukacji Informatycznej i Zastosowań Komputerów
w Warszawie, Raszyńska 8/10
agnieszka.borowiecka@oeiizk.waw.pl

Abstract. A JavaScript library p5.js is designed for programming multimedia on websites. It is a perfect tool for combining programming lessons with creation of interesting projects. On our workshop we will prepare a website with Caesar's cipher encryption.

1. Wprowadzenie

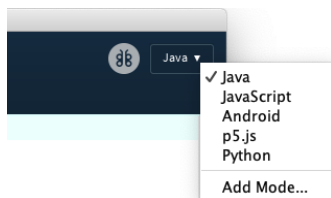
Internet jest naturalnym środowiskiem dla naszych uczniów. Umieją tam znaleźć interesujące ich treści, nawiązać kontakt z rówieśnikami, dzielić się swoimi osiągnięciami. Spróbujmy wykorzystać ich zainteresowania łącząc chęć do zaistnienia w sieci z nauką programowania. Proponujemy pokazać możliwość stworzenia ciekawej grafiki oraz różnych elementów interaktywnych na strony www w języku JavaScript, a konkretnie za pomocą biblioteki p5.js. Do pracy wykorzystamy środowisko Processing zainstalowane na komputerze oraz edytor p5.js dostępny online.

2. Rozpoczynamy pracę

Pierwotnym celem powstania środowiska Processing było udostępnienie narzędzia przybliżającego programowanie artystom, projektantom, nauczycielom i początkującym. Biblioteka JavaScript p5.js rozszerza i modyfikuje te zadania, dostosowując je możliwości dzisiejszych stron WWW. Jednak p5.js nie tylko dostarcza pełen zestaw narzędzi do rysowania, ale równocześnie pozwala traktować całą stronę widoczną w przeglądarce jako nasz szkicownik. Po dołączeniu dodatkowych bibliotek możemy wchodzić w interakcje z obiektami HTML5, takimi jak tekst, pola formularza, wideo czy dźwięk.

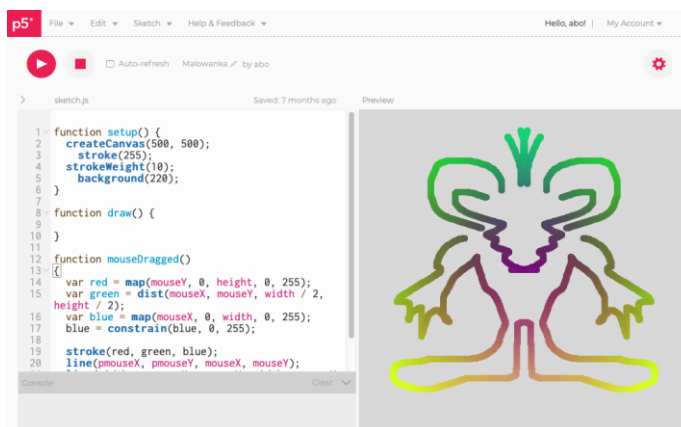
W środowisku Processing domyślnie pracujemy w trybie Java, posługując się uproszczoną wersją tego języka. Możemy dodać inne tryby, np. Android, Python lub

p5.js Mode, za pomocą okna Contribution Manager. Po zainstalowaniu niezbędnych plików wystarczy przełączyć tryb pracy wybierając p5.js z podręcznego menu w prawym górnym rogu okna aplikacji.



Rysunek 1 Wybieranie trybu pracy środowiska Processing

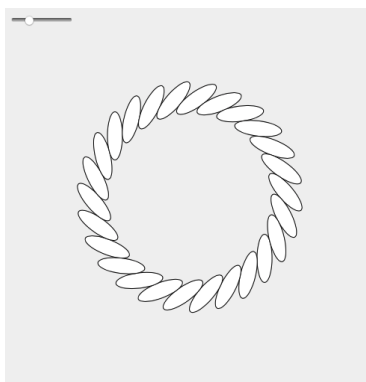
Bibliotekę p5.js możemy także dołączyć do strony pisząc skrypty w dowolnym edytorze tekstowym, wystarczy pobrać odpowiednie pliki ze strony domowej projektu[1]. Wygodniej jest jednak skorzystać z gotowego edytora online[2]. Warto przy tym założyć konto, dzięki któremu będziemy mogli łatwo zapisać, nazwać i udostępnić swoje prace.



Rysunek 2 Edytor online – testowanie projektu

3. Matematyczne zabawy z elipsą

Pierwszym zadaniem będzie narysowanie 26 elips rozmieszczonych równomiernie na okręgu. Wykorzystamy dwie standardowe funkcje **translate()** i **rotate()**, umożliwiające przemieszczanie i obracanie obiektów wewnątrz obszaru rysowania. Najpierw przeniesiemy środek układu współrzędnych z górnego lewego narożnika do środka okna, następnie będziemy w pętli obracać się o pewien kąt i rysować kolejne elipsy.



Rysunek 3 Rysowanie 26 elips położonych na okręgu

Do ustalenia kąta obrotu możemy wykorzystać stałą `TWO_PI`, którą podzielimy przez liczbę rysowanych elips. W pierwszej wersji programu wielka i mała oś każdej elipsy jest taka sama – na ekranie pojawi się 26 kółek.

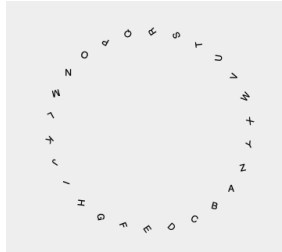
```
function setup() {  
  createCanvas(500, 500);  
}  
  
function draw() {  
  background(240);  
  translate(width/2, height/2);  
  for (var i = 0; i < 26; i++) {  
    ellipse(100, 80, 20, 20);  
    rotate(TWO_PI/26);  
  }  
}
```

Rysunek 4 Rysowanie elips położonych na okręgu – program

Projekt możemy szybko rozbudować dodając kolory lub możliwość zmiany wielkości elips za pomocą suwaka. Biblioteka `p5.js` pozwala korzystać z różnych elementów HTML5 (np. suwaki, przyciski, pola wyboru, odnośniki), wymaga to jedynie dołączenia dodatkowej biblioteki `p5.dom`.

4. Tworzymy pomoc do szyfrowania

Zamiast rysować kółka, możemy wypisywać na ekranie kolejne litery alfabetu. Za-uważmy, że przytoczony skrypt wymaga tylko niewielkiej korekty – zamiast funkcji `ellipse()` musimy wywołać funkcję `text()`. Warto także przygotować zmienną *alfabet* przechowującą wszystkie litery alfabetu łacińskiego.



Rysunek 5 Rysowanie koła z alfabetem

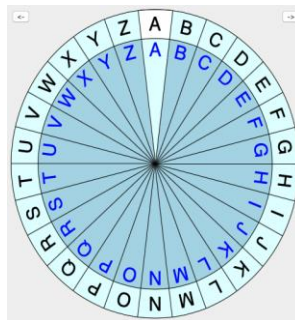
Łatwo zauważyć, że wyświetlane litery są nieco za małe i w nietypowy sposób obrócone. Poprawimy kod, zmieniając miejsce wypisywania tekstu i dodając przesunięcia – skok ze środka okna do punktu wypisywania tekstu i powrót.

```
let alfabet = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";

function setup() {
  createCanvas(500, 500);
  textSize(30);
}

function draw() {
  background(240);
  translate(width/2, height/2);
  for (var i = 0; i < 26; i++) {
    translate(0, -200);
    text(alfabet[i], 0, 0);
    translate(0, 200);
    rotate(TWO_PI/26);
  }
}
```

Rysunek 6 Program wyświetlający poprawiony alfabet



Rysunek 7 Koło szyfrowe Cezara

Możemy teraz dodać drugie nieco mniejsze koło, także złożone z kolejnych liter alfabetu. Jeśli dodatkowo obrócimy wewnętrzne koło względem zewnętrznego o wielokrotność kąta między literami ($360/26$), to otrzymamy rysunek tzw. koła szyfrowego Cezara. Możemy teraz bawić się w szyfrowanie i odszyfrowywanie różnych wiadomości. Dodając do projektu dwa przyciski pozwolimy użytkownikowi samodzielnie ustalić klucz szyfrowania. Warto także dodać kilka elementów graficznych uatrakcyjnających aplikację.

5. Podsumowanie

W opisywanych projektach skupiliśmy się na warstwie wizualnej projektując narzędzie wspomagające uczniów przy próbach szyfrowania i rozszyfrowywania tekstów za pomocą szyfru Cezara. Biblioteka p5.js może być również wykorzystana do przygotowania aplikacji pozwalającej szyfrować dowolne teksty w języku polskim i wypisującej wyniki na ekranie.

Podaj klucz klucz = 3

3

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

Podaj tekst do tłumaczenia

Ala ma kota i psa

A	L	A		M	A		K	O	T	A		I		P	S	A									
D	O	D		P	D		N	R	W	D		L		S	V	D									

Rysunek 8 Program szyfrujący z podanym kluczem

Literatura

1. Strona domowa p5.js, <https://p5js.org>, ostatni dostęp 28.05.2019 roku.
2. Edytor p5.js, <https://editor.p5js.org>, ostatni dostęp 28.05.2019 roku.