Nazwa implementacji: Canvas

Autor:

Opis implementacji: Stworzymy aplikację pozwalającą na rysowanie prostych figur geometrycznych.

Gdy chcemy stworzyć aplikację, która prezentuje informacje, wykorzystując różne obiekty graficzne, obrazki i animacje najlepiej jest wykorzystać element <canvas>, który daje nam najwięcej możliwości w rysowaniu obiektów, oraz najefektywniej wykorzystuje możliwości komputera włącznie ze wsparciem karty graficznej przy złożonych animacjach.

Zadanie: Żeby rozpocząć pracę, wystarczy stworzyć prosty dokument HTML z jednym elementem:

```
<canvas id=”canvas” width=”700” height=”500”></canvas>
```

Kilka parametrów CSS określajacych wielkość i położenie naszego pola do rysowania,

```
#canvas {
 position: relative;
 display: block;
 margin: 100px auto 0;
 border: 3px solid #999;
```

oraz proste linijki JavaScript tworzącą obiekt, na którym będziemy dokonywać operacji rysowania:

```
var canvas = document.getElementById('canvas');
var ctx = canvas.getContext('2d');
```

Uwaga: W tym przypadku nie korzystamy z pomocy biblioteki jQuery, gdyż byłaby użyta zaledwie w kilku miejscach, które możemy bez problemu zastąpić zwykłym kodem JavaScript:

- Konstrukcję, która opóźnia wywołanie kodu, dopóki nie załaduje się cały dokument HTML

```
$(function() { /* kod aplikacji */ });
```

Zastępujemy obsługą zdarzenia

DOMContentLoaded:

```
document.addEventListener('DOMContentLoaded', function() {
/* kod aplikacji */
       });
```

- Wyszukiwanie elementu po atrybucie id: **$('#canvas')** zamieniamy na:

```
document.getElementById('canvas');
```

Rysowanie podstawowych obiektów geometrycznych

Utworzony obiekt pod zmienną ctx pozwala nam na rysowanie z wykorzystaniem metod:

- fillRect(x,y,width,height) – rysuje prostokąt zaczynający się w punkcie x,y o szerokości i wysokości: width, height.
- strokeRect(x,y,width,height) – tak samo, jak fillRect() tylko w trakcie rysowania powstaną tylko krawędzie prostokąta.

  clearRect(x,y,width,height)

- – pozwala na wyczyszczenie wskazanego fragmentu lub całości canvas.
- beginPath() – informuje, że zaczynamy rysować wielokąt.

KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI ORE FWIOO UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY

Projekt “Strategia Wolnych i Otwartych Implementacji jako innowacyjny model zainteresowania kierunkami informatyczno-technicznymi oraz wspierania uczniów i uczennic w kształtowaniu kompetencji kluczowych” współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

- closePath() – informuje o zakończeniu rysowania wielokąta.
- lineTo(x,y) – rysuje linie do punktu x,y.
- moveTo(x,y) – oznacza punkt, w którym rozpoczynamy rysowanie.
- fill() – po zdefiniowaniu wielokąta wypełnia jego zawartość.
- stroke() – po zdefiniowaniu wielokąta rysuje jego krawędzie.

Oprócz powyższych metod mamy jeszcze dwa parametry:

fillStyle

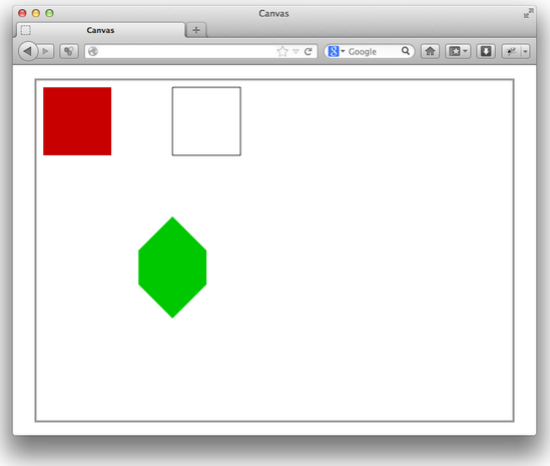
- – przypisujemy do niego kolor wypełnienia rysowanego obiekt.

strokeStyle

- – przypisujemy do niego kolor dla krawędzi obiektu.

Uwaga: Jeśli chcemy, żeby rysowany obiekt miał zarówno wypełnienie jak i krawędzie, musimy wywołać zarówno metodę fillRect() i strokeRect() lub fill() i stroke().

Zadanie: Napisz kod rysujący krawędzie i wypełnienie prostokąta w różnych kolorach, oraz co najmniej 5 ramienny wielokąt.



Obsługa zdarzeń myszki

Żeby pozwolić użytkownikowi rysować nowe figury, potrzebujemy stworzyć obsługę dwóch zdarzeń:

- mousedown – użytkownik naciska przycisk myszki i zaczyna rysować prostokąt.
- mouseup – użytkownik puszcza przycisk myszki i kończy rysować prostokąt.

Miejsce, w którym użytkownik kliknął myszką, możemy pobrać z event.layerX i event.layerY lub w przypadku gdy korzystamy z przeglądarki Opera: event.offsetX i event.offsetY. Uwaga: Parametry te podają odległość względną do krawędzi najbliższego kontenera, który w CSS ma ustawione: position:relative; lub position:absolute;. Bez tej właściwości otrzymamy odległość przycisku do krawędzi okna przeglądarki i będziemy musieli dodatkowo obliczać punkt początkowy, w którym znajduje się nasz tag <canvas>.

Zadanie: Korzystając z poniższego szablonu napisz kod, który pozwoli użytkownikowi rysować prostokąty.

KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
ORE
FWIOO
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY

Projekt “Strategia Wolnych i Otwartych Implementacji jako innowacyjny model zainteresowania kierunkami informatyczno-technicznymi oraz wspierania uczniów i uczennic w kształtowaniu kompetencji kluczowych” współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

```
canvas.addEventListener('mousedown', function(event){
 /* Zachowaj wartości początkowe layerX i layerY */
});

canvas.addEventListener('mouseup', function(event){
 /* Korzystajac z zapisanych wartości w mousedown
       oblicz szerokość i wysokość rysowanego prostokąta
       i narysuj go używając fillRect(); */
});
```

Zadanie dodatkowe: Dodaj kilka przycisków, które pozwolą zmienić kolor rysowanych obiektów.

Animacje

Animacja w przypadku korzystania z canvas to nic innego jak ciągłe czyszczenie i ponowne rysowanie obiektów, które zmieniły swoje położenie od poprzedniego rysowania. W naszej aplikacji możemy zaprezentować prostą animację w trakcie, kiedy użytkownik rysuje swój prostokąt, pokazując w trakcie jego kontury. Żeby móc to zrobić będziemy potrzebować też tabeli, w której będziemy przechowywać informacje o wszystkich już wcześniej narysowanych obiektach, żeby móc je wykorzystać przy odświeżaniu ekranu. Pamiętaj też o zapisaniu koloru prostokąta.

```
ctx.fillRect(x,y,width,height);
       objects.push({x:x, y:y, w:width, h:height, c:ctx.fillStyle});
```

Zadanie: Dodaj obsługę kolejnego zdarzenia mousemove które będzie:

- czyściło całą przestrzeń do rysowania: ctx.clearRect(0,0,700,500);
- ponownie rysowało wszystkie wcześniej stworzone obiekty;
- rysowało aktualne kontury rysowanego prostokąta;

Animacja z użyciem setInterval, setTimeout i requestAnimationFrame

Poprzednia animacja była uruchamiana poprzez akcje wywoływane przez użytkownika. Natomiast jeśli chcemy, żeby animacja działała nawet, gdy nic nie jest robione w aplikacji, możemy odświeżać ekran, używając jeden z metod:

- **setInterval() – wywołuje bez przerwy podaną funkcję po podanej liczbie milisekund.**

setTimeout()

– wywołuje podaną funkcję po podanej liczbie milisekund. Może zastąpić działanie setInterval() jeśli na końcu wywołanej funkcji będzie ponownie użyty

setTimeout().

Zadanie:

Spraw, że prostokąty po narysowaniu będą spadały na ziemię. Porównaj działanie obu opisanych metod.

Zadanie dodatkowe:

Wykonaj powyższe zadanie z użyciem

requestAnimationFrame().

requestAnimationFrame() jest specjalnie zmodyfikowaną wersją setTimeout(). Silnik przeglądarki sam oblicza jak często ma nastąpić odświeżenie ekranu w zależności od ilości obliczeń i obciążenia procesora.

KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI ORE FWIOO UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY

Projekt “Strategia Wolnych i Otwartych Implementacji jako innowacyjny model zainteresowania kierunkami informatyczno-technicznymi oraz wspierania uczniów i uczennic w kształtowaniu kompetencji kluczowych” współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

```
var draw = function() {
 /* Uruchamiany kod */
 requestAnimationFrame(draw);
};
requestAnimationFrame(draw);
```

Uwaga: Funkcja ta nie jest jeszcze stabilnie zaimplementowana we wszystkich przeglądarkach, więc każda z nich udostępnia ją pod nazwą z prefixem. Żeby móc z niej swobodnie korzystać dodaj ten na początku ten kod:

```
var requestAnimationFrame = window.requestAnimationFrame ||        window.mozRequestAnimationFrame ||        window.webkitRe
```


KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
ORE
FWIOO
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY

Projekt “Strategia Wolnych i Otwartych Implementacji jako innowacyjny model zainteresowania kierunkami informatyczno-technicznymi oraz wspierania uczniów i uczennic w kształtowaniu kompetencji kluczowych” współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.