

Rzeszów, 30.04.2018

Projekt Techniki Multimedialne
Temat: Parallax Scrolling
Dokumentacja techniczna

Gracjan Kudra

3 EF-DI L03

1. Dokumentacja sekcji CSS

1.1. Znaczniki

Edycja kodu CSS opierała się głównie o zmodyfikowanie istniejących już identyfikatorów pod swoje potrzeby.

Nadpisane identyfikatory uwzględnione w Listingu 1.

Listing 1

```
body {  
  background: #2a3045; //ustawienie tła  
  margin: 0;           //margines  
}  
section {  
  position: relative;    //  
  display: flex;         //  
  justify-content: center; // ustawienie wyśrodkowanej sekcji dostosowującej  
  height: 99vh;          // się do zmian okienka przeglądarki  
  align-items: center;    //  
  min-height: 499px;     //  
}
```

2. Dokumentacja sekcji kodu JavaScript

2.1. Zmienne użyte w programie

Zmienne zadeklarowane w skrypcie wymienione w Listingu 2.

Listing 2

```
var warstwy = [];  
var height = 600;  
var width = 600;  
var zaladowane = 0;  
var container = document.getElementById("container");  
var snap = new Snap(width, height); //element z biblioteki Snap SVG  
var obrazki = [  
  {url: "img/kotek2.gif", x: -39, y: -99, offset: -0.25},  
  {url: "img/warstwa3.png", x: -439, y: 139, offset: 0.3},  
  {url: "img/warstwa2.png", x: -9, y: 279, offset: 0.1},  
  {url: "img/warstwa1.png", x: -89, y: 429, offset: -0.1},  
  {url: "img/logo.png", x: 99, y: 76, offset: 0.15},  
  {url: "img/granica.png", x: 79, y: 89, offset: 0.2},
```

```
{url: "img/granica2.png", x: -19, y: 1, offset: 0}  
];  
var obraz,_obraz,i,granica,deltaX,deltaY,lax,x3,y3,group,obj;
```

2.2. Funkcja program_glowny()

W ciele funkcji wywołuje się główna część programu, przypisywany do elementu 'container' jest element węzeł 'snap'. Kolejno do zmiennych 'group' i 'obj' przypisywane jest stworzenie grupy elementów obiektu z biblioteki Snap. Do obiektu 'obj' przypisywany jest atrybut 'transform', ustawiony na 'scale(1)' – odpowiada to rozmiarowi obrazków wczytywanych w skali 1:1. Do grupy dołączany jest obiekt 'obj'. Funkcja następnie wchodzi do pętli, która kończy się po osiągnięciu maksymalnej długości zmiennej 'obrazki'. W ciele pętli tworzony jest 'obraz' klasy Image() następnie do niego przypisywany jest i-ty element zmiennej 'obrazki'. Dla zmiennej obraz wywoływana jest funkcja zliczaj_obrazy(). Następnie do elementu '_obraz' przypisywana jest poprzez funkcje image() wywołaną na elemencie 'snap' przestrzeń i-tego obrazu, ładowane są atrybuty obrazu takie jak zawartość, długość i szerokość. Dalej przypisany do obiektu zostaje wcześniej załadowany obraz. Do tablicy 'warstwy' zostaje zapisana kopia obrazu.

Listing 3

```
function program_glowny(){  
  container.appendChild(snap.node); //podłączenie snap-a pod container  
  group = snap.g(); //stworzenie grupy elementów dla zmiennej group  
  obj = snap.g(); //stworzenie grupy elementów dla zmiennej obj  
  obj.attr({transform: "scale(1)"}); //ustawienie atrybutu transform na wartość scale(1) dla obiektu obj  
  group.appendChild(obj); //podłączenie grupy obj pod grupe group  
  for (i = 0; i < obrazki.length; i++) {  
    obraz = new Image(); //przypisanie zmiennej obraz klasie Image  
    obraz.src = obrazki[i].url; //do zmiennej obraz przypisanie i-tego adresu obrazu  
    obraz.onload = zliczaj_obrazy; // wywołanie dla zmiennej obraz funkcji
```

```
zaladuj_obraz
_obraz = snap.image(obrazki[i].url, obrazki[i].x, obrazki[i].y); //załadowanie
"powierzchni" obrazowej do zmiennej _obraz
obj.append(_obraz); //dołączenie powierzchni do obj
warstwy.push(_obraz); //dodanie do tablicy warstwy obrazu
}
}
```

2.3. Funkcja zliczaj_obrazy()

Funkcja ma za zadanie inkrementację wartości oznaczającą ilość załadowanych obrazów. W przypadku gdy zmienna ‘załadowane’ jest równa maksymalnej liczbie wczytywanych obrazków wywoływana jest funkcja zdarzenia().

Listing 4

```
function zliczaj_obrazy(e) {
  zaladowane++; //inc
  if (zaladowane == obrazki.length) {
    zdarzenia(); //wywołanie funkcji zdarzeń
  }
}
```

2.4. Funkcja zdarzenia()

Głównym zadaniem funkcji jest obliczenie kolejnej granicy, w którą mogą być dodane obrazki i na podstawie tej granicy zostaje dodany atrybut ‘mask’, który określa tło następnego elementu jako cały poprzedni obraz. Następnie do ‘container-a’ dodawane są trzy zdarzenia po, których włączana dla każdego jest inna funkcja. Zdarzenia określają ruch myszką, najechanie myszką na element i odjechanie myszką z elementu.

Listing 5

```
function zdarzenia() {  
  granica = warstwy[warstwy.length - 1];  
  group.attr({mask: granica});  
  container.addEventListener("mouseover", f3);//najechnie na element  
  container.addEventListener("mouseout", f2);//odjechanie z elementu  
  container.addEventListener("mousemove", f1);//ruch myszką  
}
```

2.5. Funkcja f1()

Funkcja ta odpowiedzialna jest za zmianę położenia obrazków gdy myszka jest poruszana w granicy określonego ruchu. Funkcja wylicza zmienne 'deltaX' i 'deltaY', którym przypisane jest aktualne położenie myszki w osi X i osi Y. Następnie w ciele pętli funkcji do zmiennej tymczasowej 'lay' przypisywany jest i-ty element zmiennej 'warstwy'. Kolejno na potrzeby animowania warstw obliczane są nowe położenia według osi X i Y, i-tego obrazku(warstwy), które dalej wraz z węzłem 'lay' również i-tego obrazka, przekazywane są do funkcji animacyjnej. Funkcja animacyjna wywołana w ciele funkcji f1() odpowiada za animację poruszania się myszki po okręgu i co za tym idzie poruszaniem warstw po okręgu. Dodatkowo poza ciałem pętli dodana została ta sama funkcja tylko wywołana na węźle głównym 'snap', efektem czego jest poruszanie się nie tylko poszczególnych warstw ale również całego obiektu złożonych razem warstw. Dzięki zastosowaniu tej funkcji można odnieść wrażenie trójwymiarowości obiektu.

Listing 6

```
function f1(e) {  
  deltaX = e.offsetX - (width / 2);  
  deltaY = e.offsetY - (height / 2);  
  for (i = 0; i < warstwy.length; i++) {  
    lay = warstwy[i];  
    x3 = deltaX * obrazki[i].offset;  
    y3 = deltaY * obrazki[i].offset;  
    TweenMax.to(lay.node, 0.2, {x: x3, y: y3});  
  }  
  TweenMax.to(snap.node, 0.1, {rotationY: deltaX / 9, rotationX: -deltaY / 9});  
}
```

2.6. Funkcja f2()

Kolejna z funkcji zdarzeń odpowiada za to co się dzieje w przypadku gdy kursor myszy odjedzie z pola granicy ruchu. Pętla kolejno wczytuje warstwy a następnie dzięki funkcji animacyjnej programuje animacje typu 'Bounce.easeOut' czyli spadającej piłki. Do tego animacja obiektu zostaje wyrównana na wartości X i Y równym zeru.

Listing 7

```
function f2(e) {  
  for (i = 0; i < warstwy.length; i++) {  
    lay = warstwy[i];  
    TweenMax.to(lay.node, 0.8, {x: 0, y: 0, ease: Circ.easeOut});  
  }  
  TweenMax.to(snap.node, 0.8, {scale: 0.8, rotationY: 0, rotationX: 0, ease:  
    Bounce.easeOut});  
  TweenMax.to(obj.node, 1, {rotationY: 0, rotationX: 0});  
}
```

2.7. Funkcja f3()

Ostatnia z funkcji zdarzeń określa co się stanie gdy myszka najedzie na obiekt. Ustawiona zostaje animacja na całym obiekcie. Efekt animacji powoduje zwiększenie skali całego obiektu, kiedy myszka jest na danym obiekcie.

Listing 8

```
function f3(e) {  
  TweenMax.to(snap.node, 0.4, {scale: 1, ease: Elastic.easeOut});  
}
```