Tworzenie animacji z użyciem języka JavaScript

WPROWADZENIE DO TECHNIK ANIMACJI

Julian Shapiro

Helion

Tytuł oryginału: Web Animation using JavaScript: Develop & Design

Tłumaczenie: Mirosław Gołda

ISBN: 978-83-283-1572-3

Authorized translation from the English language edition, entitled: WEB ANIMATION USING JAVASCRIPT: DEVELOP & DESIGN; ISBN 0134096665; by Julian Shapiro; published by Pearson Education, Inc, publishing as Peachpit Press.

Wydawnictwo HELION
ul. Kościuszki 1c, 44-100 GLIWICE
tel. 32 231 22 19, 32 230 98 63
e-mail: *helion@helion.pl*
WWW: *http://helion.pl* (księgarnia internetowa, katalog książek)

Pliki z przykładami omawianymi w książce można znaleźć pod adresem:
ftp://ftp.helion.pl/przyklady/twanjs.zip

Drogi Czytelniku!
Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres
http://helion.pl/user/opinie/twanjs
Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Printed in Poland.

SPIS TREŚCI

ROZDZIAŁ 1.

Zalety animacji JavaScript

W tym rozdziale porównamy możliwości języka JavaScript oraz stylów CSS w zakresie tworzenia animacji i poznamy unikalne możliwości i korzyści w zakresie organizacji pracy przemawiające za zastosowaniem języka JavaScript.

Mówiąc w skrócie, przedstawię tu kontekst potrzebny do zrozumienia wszystkiego, czego nauczysz się z tej książki.

ANIMACJE JAVASCRIPT A ANIMACJE CSS

W społeczności webdeweloperskiej istnieje fałszywe przekonanie, że jedynym wydajnym rozwiązaniem w zakresie tworzenia animacji na stronach internetowych jest stosowanie animacji CSS. To nieporozumienie zaowocowało tym, że wielu deweloperów porzuciło całkowicie animacje oparte na języku JavaScript, co zmusiło ich do:

- Zarządzania całym interfejsem użytkownika za pomocą arkuszy stylów, co szybko staje się trudne w utrzymaniu.
- Rezygnacji z kontroli nad upływem czasu, dostępnej jedynie za pośrednictwem JavaScriptu. (Kontrola nad czasem potrzebna jest przy projektowaniu animacji w interfejsach odpowiadających na przeciąganie elementów, spotykanych na przykład w aplikacjach mobilnych).
- Rezygnacji z fizyki ruchu, która pozwala obiektom zachowywać się w sposób charakterystyczny dla realnego świata.
- Utraty wsparcia dla starszych wersji przeglądarek, które są wciąż popularne na świecie.

Animacje oparte na języku JavaScript są często równie szybkie jak animacje CSS. Animacjom CSS przypisywana jest niesłusznie duża przewaga szybkości w porównaniu z animacjami JavaScript. Wynika to z faktu częstego bezpośredniego utożsamiania animacji JavaScript z animacjami wprowadzonymi w bibliotece jQuery, które są rzeczywiście bardzo wolne. Alternatywne rozwiązania w postaci bibliotek JavaScript, które pomijają całkowicie wykorzystanie jQuery, potrafią być niesamowicie wydajne poprzez usprawnienie interakcji ze stroną.

UWAGA: Jedną z bibliotek, z której będziemy korzystać w tej książce, jest *Velocity.js*. Jest ona lekka, ale bardzo rozbudowana i odwzorowuje składnię animacji jQuery, aby ułatwić naukę.

Animacje CSS sprawdzają się jednak doskonale przy animacjach wyświetlanych w wyniku ruchu myszką (na przykład przy zmianie koloru łącza po najechaniu kursorem na kolor niebieski). Przekształcenia CSS wpasowują się świetnie w istniejące arkusze stylów, co pozwala deweloperom uniknąć zaśmiecania strony niepotrzebnymi bibliotekami JavaScript. Co więcej, animacje CSS są bardzo szybkie.

W tej książce pokażę jednak, dlaczego JavaScript jest często lepszym rozwiązaniem przy animacjach bardziej złożonych niż proste zmiany stanu.

Nie utożsamiaj JavaScriptu z jQuery

WSPANIAŁA WYDAJNOŚĆ

JavaScript i jQuery są mylnie utożsamiane ze sobą. Animacje JavaScript są szybkie, a jQuery je spowalnia. Pomimo niesamowitej mocy tej biblioteki nie została ona zaprojektowana jako wysokowydajny silnik animacji. Nie posiada mechanizmu obronnego przed „uszkodzeniem układu strony", co skutkuje przeciążeniem przeglądarki zadaniami przetwarzania układu podczas odtwarzania animacji.

Co więcej, z uwagi na fakt, że kod bazowy jQuery służy nie tylko animacjom, ale także wielu innym celom, zużywa on dużo pamięci i wywołuje mechanizmy czyszczenia pamięci w przeglądarce, co doprowadza do przestojów w działaniu animacji. A do tego, w wyniku podjętych przez zespół jQuery decyzji mających ułatwić początkującym użytkownikom tworzenie przyzwoitej jakości kodu interfejsu użytkownika, jQuery zrezygnowało z zalecanej praktyki wykorzystywania funkcji `requestAnimationFrame`, która pozwala na znaczne zwiększenie liczby wyświetlanych klatek w animacji internetowej.

Biblioteki animacji JavaScript, które nie korzystają wcale z mechanizmów jQuery, oferują fenomenalną wydajność dzięki zastosowaniu przepływów interakcji na stronie. Jedną z bibliotek, którą wykorzystamy w tej książce, jest Velocity.js. Biblioteka ta jest lekka, ale też bogata w funkcjonalności, a jej składnia odwzorowuje składnię animacji jQuery, co znacznie ułatwia naukę.

Tematem wydajności zajmiemy się w rozdziale 7., „Wydajność animacji". Poznając niuanse wydajności renderowania w przeglądarce, zyskasz podstawy wiedzy, dzięki którym zbudujesz animacje działające prawidłowo na wszystkich przeglądarkach i urządzeniach, niezależnie od ich szybkości przetwarzania.

MOŻLIWOŚCI

Szybkość nie jest oczywiście jedynym powodem, dla którego powinniśmy zdecydować się na użycie języka JavaScript — równie istotne jest bogactwo dostępnych funkcjonalności.

PRZEWIJANIE STRONY

Przewijanie strony jest jednym z najczęstszych zastosowań dla animacji opartej na języku JavaScript. Najnowszym trendem w zakresie projektowania stron internetowych jest tworzenie długich stron, które po przewinięciu w dół są rozbudowywane o nowe treści za pomocą animacji.

Biblioteki animacji JavaScript, takie jak Velocity, udostępniają proste funkcje, które umożliwiają dodanie treści do strony:

```
$element.velocity("scroll", 1000);
```

Rezultatem działania tego kodu jest przewinięcie przeglądarki do górnej krawędzi elementu `$element` w ciągu 1000 milisekund. Warto zauważyć, że składnia Velocity jest niemal identyczna ze składnią funkcji `$.animate()` języka jQuery, którą zajmiemy się w dalszej części tego rozdziału.

ODWRÓCENIE ANIMACJI

Odwróceniem animacji nazywamy w skrócie odwrócenie efektów poprzedniej animacji na wybranym elemencie. Polecenie `reverse` instruuje element, że ma za pomocą animacji wykonać powrót do stanu sprzed animacji. Powszechnym zastosowaniem jest w tym przypadku okno dialogowe, które najpierw się pojawia, a następnie chowa po zamknięciu przez użytkownika.

Niezoptymalizowany proces odwracania animacji mógłby polegać na śledzeniu szeregu właściwości elementów poddawanych ostatnio animacji, których efekty mogą zostać odwrócone. Utrzymywanie tych wszystkich stanów animacji w kodzie interfejsu użytkownika stałoby się jednak szybko trudne w utrzymaniu. Polecenie `reverse` biblioteki Velocity wykonuje tę pracę za nas.

Składnia polecenia `reverse` przypomina składnię polecenia `scroll`. Wywołujemy je, przekazując jako pierwszy argument ciąg `"reverse"`:

```
// Pierwsza animacja: Zmniejszamy wartość opacity
// (nieprzezroczystość) do zera
$element.velocity({ opacity: 0 });
// Druga animacja: Odwracamy poprzedni efekt
// i zwiększamy wartość opacity do 1
$element.velocity("reverse");
```

Kontrola nad czasem trwania animacji oferowana przez JavaScript nie polega jedynie na jej odwracaniu: JavaScript umożliwia również globalne spowolnienie lub przyspieszenie wszystkich wykonywanych aktualnie animacji. Więcej na ten temat dowiesz się z rozdziału 4., „Praca z animacjami”.

RUCH WYKORZYSTUJĄCY PRAWA FIZYKI

Wykorzystanie fizyki w projektowaniu animacji jest czynnikiem umożliwiającym uzyskanie wspaniałych doznań użytkownika (UX) korzystającego z Twojej strony: interfejsów, które w naturalny sposób reagują na jego działania. Inaczej mówiąc — interfejsów naśladujących ruch obiektów w prawdziwym świecie.

Jako proste, a zarazem rozbudowane wprowadzenie w możliwości Velocity w zakresie fizyki ruchu pokażę efekt wygładzania ruchu (ang. *easing*) bazujący na fizyce sprężyny. (Koncepcją wygładzania ruchu zajmę się w następnym rozdziale). W celu wykorzystania standardowych ustawień wygładzania przekazujemy ciąg znaków odpowiadający zdefiniowanej krzywej wygładzania (na przykład `"ease"` lub `"easeInOutSine"`). Typ wygładzania fizyki sprężyny przyjmuje z kolei dwuelementową tablicę.

```
// Animujemy szerokość elementu do "500px" z wykorzystaniem
// łagodzenia w fizyce sprężyny, korzystając z jednostek
// 500 nacisku i 20 jednostek tarcia
$element.velocity({ width: "500px" }, { easing: [ 500, 20 ] });
```

Elementy tablicy reprezentują kolejno napięcie i tarcie symulowanej sprężyny. Wyższe wartości nacisku zwiększają ogólną szybkość i sprężystość animacji. Niższe wartości zwiększają szybkość wibracji w końcowym etapie animacji. Modyfikacja tych wartości pozwala nadać animacjom niepowtarzalny profil ruchu, co z kolei pozwala uwydatnić ich indywidualne zachowania.

PROCES ŁATWY W UTRZYMANIU

Projektowanie animacji jest eksperymentalnym procesem wymagającym wielokrotnych modyfikacji wartości czasu i wygładzania w celu uzyskania unikalnego efektu na stronie. Niemal pewne jest, że właśnie wtedy, gdy już maksymalnie dopracujesz swój projekt, klient zażąda wprowadzenia jakichś znaczących zmian. W takich momentach najważniejsze jest posiadanie kodu, który jest łatwy w utrzymaniu.

Rozwiązanie tego problemu w JavaScripcie jest bardzo eleganckie i zostanie opisane dokładnie w rozdziale 4., „Praca z animacjami". W tej chwili przedstawię tylko krótkie wyjaśnienie: istnieją techniki łączenia ze sobą pojedynczych animacji JavaScript — o różnym czasie trwania, wygładzaniu itd. — w których czas działania jednej animacji nie wpływa na pozostałe. Oznacza to, że mamy możliwość zmiany pojedynczych czasów trwania bez konieczności wykonywania przeliczeń i możemy w prosty sposób modyfikować ustawienia tak, aby działały równolegle lub w ustalonej kolejności.

PODSUMOWANIE

Tworząc animacje z wykorzystaniem CSS, ograniczeni jesteśmy do funkcji udostępnianych przez specyfikację CSS. Język JavaScript, dzięki specyfice języków programowania, umożliwia tworzenie bibliotek, które posiadają nieograniczone możliwości kontroli. Silniki animacji wykorzystują te możliwości do przygotowywania rozbudowanych funkcjonalności, które znacznie usprawniają pracę i poszerzają możliwości interakcji tworzonych animacji. To właśnie jest tematem tej książki: projektowanie pięknych animacji w jak najefektywniejszy sposób.

W następnym rozdziale wyjaśnię, jak korzystać z *Velocity.js*, silnika animacji, który wybrałem na potrzeby tej książki. Aby nabrać biegłości w stosowaniu tego silnika, musimy zrozumieć, jak wykorzystać funkcje już poznane i wiele, wiele innych.

SKOROWIDZ

M

O

P

S

T

U

V

W

Z

Twój unikalny przewodnik po świecie animacji JavaScript!

Do niedawna umieszczenie zaawansowanej animacji na stronie WWW wymagało zastosowania technologii flashowej opracowanej przez firmę Adobe. Takie rozwiązanie powodowało wiele zagrożeń, a ponadto konieczna była instalacja dodatkowego oprogramowania. Ciągły rozwój możliwości przeglądarek, języka JavaScript oraz HTML umożliwił w końcu tworzenie nawet bardzo zaawansowanych animacji bez potrzeby korzystania z zewnętrznych narzędzi.

Jeżeli chcesz się nauczyć tworzyć animacje, czerpiąc z możliwości JavaScript, trafiłeś na właściwy podręcznik. W trakcie lektury kolejnych rozdziałów zdobędziesz bezcenną wiedzę na temat biblioteki Velocity.js oraz zasad projektowania ruchu. Potem przejdziesz do praktycznych aspektów pracy z animacjami — dowiesz się, jak rozdzielić style i logikę, jak zorganizować różne sekwencje animacji oraz jak dodawać efekty. Następnie nauczysz się animować teksty i korzystać z grafiki wektorowej w formacie SVG oraz zadbasz o wydajność animacji. Ta książka stanowi doskonałą lekturę, po której zaskoczysz użytkowników Twoich stron WWW atrakcyjnymi animacjami.

DZIĘKI TEJ KSIĄŻCE:

- Poznasz bibliotekę Velocity.js
- Zrozumiesz teoretyczne podstawy projektowania ruchu
- Zaznajomisz się z najlepszymi technikami kodowania animacji
- Przygotujesz animację tekstu
- Poznasz format SVG
- Wykonasz atrakcyjny pokaz dla Twoich klientów

JULIAN SHAPIRO — założyciel startupu oraz deweloper. Obecnie skupia uwagę na poprawie prezentacji ruchu w internecie. Autor biblioteki Velocity.js, najczęściej używanej do tworzenia animacji JavaScript na stronach WWW. Velocity.js jest stosowana przez takie serwisy jak WhatsApp oraz Tumblr.

Helion
37666 numer katalogowy
księgarnia internetowa
http://helion.pl
zamówienia telefoniczne
0 801 339900
0 601 339900
Sprawdź najnowsze promocje:
http://helion.pl/promocje
Książki najchętniej czytane:
http://helion.pl/bestsellery
Zamów informacje o nowościach:
http://helion.pl/nowosci
Helion SA
ul. Kościuszki 1c, 44-100 Gliwice
tel.: 32 230 98 63
e-mail: helion@helion.pl
http://helion.pl
ISBN 978-83-283-1572-3
9 788328 315723
Informatyka w najlepszym wydaniu
cena: 34,90 zł

ALWAYS LEARNING
PEARSON