

УДК 681.142.7

Б.А. Демида, М.А. Назаркевич, Т.А. Марусенкова

**ОПТИМІЗАЦІЯ РОЗМІРІВ ФАЙЛІВ
ПРИ ПРОГРАМУВАННІ ЗОБРАЖЕНЬ ДЛЯ WEB**

Вимоги комп'ютерної підготовки зображень для Web

Дані в Інтернеті зберігаються на інтернет-сторінках, які містять інформацію в цифровому вигляді. Оцінимо їх у двійкових одиницях виміру. Сторінка з книги середнього формату має в собі близько 2000 байтів або (2000x8)

1600 бітів. При швидкості передачі інформації 16 000 біт/с сторінка, як бачимо, буде передана за одну секунду. За одну ж хвилину можна передати вміст таких 60 сторінок [2]. Сьогодні графіка є важливим елементом у складі Web-сторінок. Її застосовують як наочний і привабливий різновид інформації. Окрім інформативних, змістовних малюнків, у Web-сторінках можуть бути присутні декоративні графічні зображення – різноманітні написи, заголовки, логотипи, фігурні горизонтальні і вертикальні розділювачі, у тому числі анімовані. Все більше Web-сайтів будують свої гіперпосилання на основі графічних зображень. Зокрема, це такі загальновідомі знаки, як зображення стрілок, що позначають переходи між сторінками в дереві сайта, або будинку як символу домашньої сторінки, конверт, що асоціюється з електронною поштою, кошик для покупок, зроблених через всесвітню мережу, і т.д. Якщо врахувати, що інтернет-сторінка містить графіку й одна піктограма займає в середньому 2 кБ, а на сторінці залежно від дизайну є 5 – 10 піктограм, то розмір сторінки зростає до 22 кБ, і вона буде передана не за 1 с, а за 10. Розмір якісної Web-сторінки не повинен перевищувати 50 кБ, і завдання розробника полягає в знаходженні оптимального співвідношення якості та обсягу пам'яті для графіки у Web [1].

Способи подання графічних зображень

Сьогодні все більше розробників стали використовувати анімовані об'єкти. Розміри їх пропорційні до розмірів кадрів, з яких вони складаються. Суть деяких видів анімації полягає у повороті початкового зображення на деякий кут або ж у плавної зміни його кольору. Оскільки растрове зображення є матрицею пікселів, то для повороту зображення на 90° теоретично досить транспонувати цю матрицю. Оперувати кодом gif-файла неможливо, оскільки він недоступний для читання пересічному користувачеві, тому для повороту послугуємося двома окремими малюнками, збереженими у двох gif-файлах. Потреба в самостійному керуванні властивостями пікселів зображення за певними законами приводить до ідеї програмування зображень.

Графічні посилання повинні бути інтуїтивно зрозумілими. Це зумовлює простоту і чіткість, схематичність і, як правило, симетричність малюнка. Найпридатнішим для збереження піктограм вважається gif-формат. Проте він обмежений 256-ма кольорами. Для передачі мережею графічних зображень потрібно більше часу, ніж їх програмованих аналогів. Для збереження пам'яті пропонується програмувати піктограми виключно засобами JavaScript і HTML [5]. Розмір HTML-файла в байтах дорівнює кількості символів у його коді. Опис кольору та розміру кожного символу засобами HTML надто громіздкий. Цю проблему розв'язує умовна генерація документа засобами JavaScript. У HTML колір символу чи групи символів, кегль шрифту та інтерліньяж – керовані розробником величини. Нехай текст являє собою m рядків з n текстовими символами в кожному, причому всі символи – крапки. Можна підібрати такий кегль та інтерліньяж, щоби текст на екрані здавався суцільним прямокутником. Таким чином отримуємо текстовий аналог растрового зображення.

Оптимізація розмірів файлів при програмуванні зображень

Запрограмуємо найпопулярніші піктограми й дослідимо їх переваги. Нехай маємо два однотонні прямокутники, причому розміри їх однакові, а кольори різні. З точки зору растрової графіки ці зображення абсолютно не подібні, оскільки кольори жодної пари відповідних пікселів не збігаються. Таким чином, до-

АВТОМАТИЗАЦІЯ ТЕХНОЛОГІЇ ДРУКАРСТВА

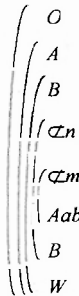
водиться формувати два різні графічні файли. Значно ефективніше описати одне зображення і здійснити зміни відносно нього в іншому.

Неефективність використання пам'яті зростає зі збільшенням числа подібних зображень. Приклад – анімація з плавною зміною “кадрів”. Нехай потрібно побудувати на екрані однотонний прямокутник розміром $m \times n$ точок. Запишемо код у мові HTML без використання JavaScript та з його використанням. Покажемо різницю між цими двома записами кодувань. Мовою HTML це можна зробити так: m разів повторити запис

```
<font color=   size=   > ..... <br></font>.
```

n точок

Подамо модель коду на JavaScript. Побудовано її засобами алгебри впорядкувань [3, 4] алгоритмів:



Тут O – початок алгоритму; W – кінець алгоритму; a – атрибут присвоєння кольору; b – атрибут присвоєння розміру; Aab – оператор побудови елементарних частинок зображення; B – оператор нового рядка; m – кількість стовпців у зображенні; n – кількість рядків програмованого зображення.

Виявлено, що ефективність використання JavaScript може бути наближено виражена величиною $\frac{m}{3}$, де m – кількість рядків програмованого прямокутника.

Якщо в деякому зображенні присутні не менше трьох однакових рядків, розташованих послідовно, використання JavaScript ефективне. Коли пікселі прямокутника мають один колір, то для побудови прямокутника таких же розмірів, але іншого кольору, досить змінити атрибут *color*.

Документ доцільно генерувати у плаваючий фрейм.

Подання графічних файлів таким методом розв'язує проблему масштабування графічних зображень, оскільки завдання масштабування шрифтів є вирішеним. Стає гнучкою зміна кольорів у зображенні – для цього досить змінити числове значення кольору в JavaScript-кодi. Для збільшення чи зменшення розмірів зображення достатньо змінити кегль, а для розтягу чи стиснення – інтерліньяж. Подібний малюнок можна отримати з існуючого шляхом зміни в його описі числових значень. Викладені положення дають

можливість гнучко модифікувати графічні зображення. Це простіший спосіб, аніж зміни зображень у графічному редакторі. Застосування інших символів дозволяє урізноманітнити малюнок.

По даних моделях створено піктограми, горизонтальні та вертикальні розділювачі. Доцільно описувати функції створення зображень і передавати їм параметри (кольори, кегль шрифту та інші). Отже, один раз описавши функцію і викликаючи її з різними параметрами, можна в ході виконання JavaScript-коду міняти колір, форму, розмір малюнка, повертати його на 90, 180 і 270°, створювати дзеркальне відображення тощо.

При завантаженні документа розроблена функція має аргумент – додатне число $c=1$. Стрілка має вигляд














При натисканні на стрілку значення соокіє допоміжного документа в плаваючому фреймі стає нульовим. Нульове значення передається розробленій функції. Викликана функція будує дзеркальне відображення попереднього малюнка. Стрілка має вигляд



Розмір файлу з JavaScript-кодом – 675 байтів. Розмір кожного з двох файлів зі стрілками становить 852 байти. HTML-файл менший від двох gif-файлів на 60,38%.

Реалізація програмованих зображень

Запропонований підхід реалізований для чотирьох стандартних піктограм і різноманітних горизонтальних розділювачів.

1. 
2. 
3. 
4. 
5. 
6. 
7. 
8. 
9. 
10. 
11. 

Порівняльні характеристики розмірів HTML- і gif-файлів наведені в таблиці.

Оптимізація розмірів файлів при побудові графічних зображень

Опис зображення	Розмір gif-файла, байти	Оптимізовані розміри програмованих файлів, байти	Зменшення розмірів файлів, відсотки
Будинку (посилання на домашню сторінку)	890	588	66,9 ¹
Стрілки	852	426	50,0
Кошика	926	607	34,4
Конверта	906	760	16,1
Документа	904	822	9,0
Горизонтальної лінії №1	1320	492	62,7
Горизонтальної лінії №2	1176	685	41,8
Горизонтальної лінії №3	2300	685	70,2
Горизонтальної лінії №4	2609	685	73,7
Горизонтальної лінії №5	1629	687	57,8
Горизонтальної лінії №6	1134	500	55,9

Таким чином, із застосуванням описаного способу отримано збереження пам'яті від 9,0 до 73,7% початкового обсягу. Запрограмовані зображення можуть бути коректно відображені в браузері, що не підтримує роботи з графікою. Програмування зображень дозволяє організувати вид анімації, суть якого полягає у зміні кольорового складу зображення з часом. Можна створити js-файл з описаними в ньому функціями побудови типових зображень. Після підключення js-файла в довільну HTML-сторінку досить викликати в ній відповідну функцію та передати їй потрібні параметри. Створені програмно зображення мають властивість прозорості, що вважається важливою перевагою gif-формату над іншими форматами. Очевидно, що запропонований метод орієнтований на створення анімованих зображень, через те що дає змогу програмувати одне зображення і здійснювати відносно нього зміни в іншому. Оскільки розробник програмованого зображення має всі відомості про код малюнка, то це дозволяє шифрувати його для захисту від копіювання.

1. Вуль В.А. Электронные издания. СПб., 2003. 2. Дьяконов В. П. Интернет: Настольная книга пользователя. Изд. 4-е, переработ. и доп. М., 2002. 3. Овсяк В. Алгебра алгоритмів-секвенцій // Комп'ютерні технології друкарства: Зб. наук. пр. Львів, 1999. Вип. 4. С. 3 – 13. 4. Овсяк В. Алгоритми: аналіз методів, алгебра впорядкувань, моделі, моделювання. Львів, 1996. 5. Пасько В. Эффективная работа в Интернете. СПб., 2003.

¹ У HTML-файлі зміна кольорів зображення реалізована як реакція на подію Mouse-Over. При використанні графіки тут потрібні два різні gif-файли, тому для розрахунку процента покращення беремо подвійний розмір gif-файла.