

ВИКОРИСТАННЯ БІБЛІОТЕКИ THREE.JS ДЛЯ ВІДОБРАЖЕННЯ 3D-ГРАФІКИ У БРАУЗЕРІ

Поглянувши в минуле, можемо побачити велику кількість плагінів працюючих на Flash (мультимедійна платформа компанії Adobe Systems для створення веб-додатків або мультимедійних презентацій). Але їх супроводжує ряд недоліків, таких як:

- надмірне навантаження на центральний процесор, пов'язане з неефективністю віртуальної машини;
- недостатній контроль помилок, що призводить до частих відмов як самих додатків, так, у деяких випадках, і всього браузера, а також можливість Flash-додатків порушувати роботу всього браузера.

Еволюція техніки, створення мобільних пристроїв, зокрема смартфонів, планшетів, гонитва за можливостями своїх пристроїв – стали одними з основних потреб сучасної людини. Більшість користувачів бажає чогось нового, чогось, що зробить їх життя більш зручним, надасть їм змогу бути завжди в курсі всіх подій, працювати, спілкуватись, відпочивати. Багато людей для досягнення цих потреб користується мережею Інтернет. В даному випадку можна помітити, що мобільні пристрої погано працюють з Flash і це факт. Тому необхідно створювати удосконалений мобільні Flash-плеєри.

Порівняння Stage3D (набір 3D API, що дозволяє відображати 3D-графіку на Flash-платформі) з WebGL (Web-based Graphics Library – програмна бібліотека для мови програмування JavaScript, що дозволяє створювати на JavaScript інтерактивну 3D-графіку, що функціонує в широкому спектрі сумісних з нею веб-браузерів) з використанням бібліотеки three.js. Дане порівняння дало неочікувані результати. З перевищенням у 30 разів кількості об'єктів, Flash демонструє вищий FPS (Frames Per Second – кількість кадрів в секунду). Але як потім виявилось, відомі тести ставили дві технології у нерівні умови.

Переваги WebGL, які можна помітити – це доступність 3D на мобільних пристроях у браузері, хоча і кількість їх невелика. Однак Android Firefox версії 4 і вище, Android-браузери (в залежності від мобільних пристроїв) та Android Opera 12-ої і вищих версій дають можливість відображення графіки за допомогою WebGL. Що стосується десктопних браузерів, то майже всі вони можуть працювати з WebGL. До браузерів, які не підтримуються WebGL належать: Internet Explorer, Safari (на платформі Windows), Mobile Safari і Opera (на платформі iOS). І загалом, сьогодні, можна впевнено говорити, що не тільки HTML5 остаточно став стандартом, але і такі технології, як WebGL, досягли достатньої зрілості, але не можна не відмітити той факт, що більшість застосувань (ігри, додатки) написані на Flash. Тут потрібно згадати соціальні мережі, які містять тисячі таких ігор і додатків, реалізованих на Flash. А для мобільних платформ ігрові додатки створюються окремо, їх необхідно скачувати і встановлювати. І тут спрощень і покращень немає для людини, яка користується мобільним пристроєм і не має під рукою персонального комп'ютера. Справа не тільки в іграх. Інтернет стає швидшим, люди – вибагливішими.

Отже, сформулюємо переваги використання WebGL:

- кросбраузерність і відсутність прив'язки до певної платформи (не важливо, чи це буде ОС Windows, MacOS, Linux, головне, щоб браузер підтримував WebGL);
- використання мови JavaScript;
- автоматичне управління пам'яттю (на відміну від OpenGL, у WebGL не треба виконувати спеціальні дії для виділення і очищення пам'яті);
- оскільки WebGL для рендеринга графіки використовує графічний процесор на відеокарті (GPU), то для цієї технології характерна висока продуктивність, яка порівнянна з продуктивністю нативних додатків;

ІТ-спеціалістам необхідні нові технології та бібліотеки, які можна було б використовувати на мові програмування, зручній для використання. Цим обумовлена поява різних бібліотек, що використовують технологію WebGL, зокрема, three.js і Babylon.js.

Бібліотека three.js є однією з найпопулярніших бібліотек. Three.js – це невелика за розмірами кросбраузерна бібліотека JavaScript, що використовується для створення і відображення анімованої комп'ютерної 3D графіки при розробці веб-додатків. Three.js скрипти можуть використовуватися спільно з елементами HTML5, CANVAS, SVG або WebGL.

Якщо поставити за мету відобразити елементи у тривимірному просторі, то узагальнена схема роботи з WebGL має такий порядок:

1. Отримуємо WebGL контекст з Canvas.
2. Завантажуємо програму шейдерів:
 - створюємо програму шейдерів;
 - отримуємо вихідний код окремо для вершинного і фрагментного шейдерів;
 - компілюємо коди шейдерів;
 - приєднуємо до програми;
 - активуємо програму.
3. Встановлюємо дві матриці: model-view і projection.
4. Розміщуємо, заповнюємо, активуємо буфери даних вершин.
5. Малюємо.

Схема викликів при відображенні сцени за допомогою three.js, поділяється на дві частини (фази):

1. *Підготовча фаза*. На даній фазі для кожного об'єкта графа сцени необхідно:

- ініціалізувати матриці;
- створити геометричні групи відповідним поверхням матеріалів; при цьому для кожного об'єкта геометричної групи потрібно:
 - створити WebGL буфери;
 - створити Non-WebGL буфери;
 - додати елементи до загального списку.

Останній пункт підготовчої фази – це оновлення буферів (буфер вершин, буфер індексів та ін.).

2. Фаза рендерінга:

- оновлення матриці на всіх нащадків сцени;
- налаштування камери матриць.

Після налаштування камери для кожного елемента списку необхідно:

- налаштувати матриці;
- обрати матеріал.

Також для кожного елемента списку циклічно потрібно:

- налаштувати вершинні програми (і текстури);
- підготувати буфери;
- перейти до малювання примітивів.

Яскравим прикладами роботи WebGL є демо-сайт «Assassin's Creed Pirates» і сайт «Dino Hunt TV», а такі бібліотеки як «Three.js» спрощують створення додатків на базі WebGL.

Знання можливостей WebGL, а також таких бібліотек як Three.js ставить розробників веб – додатків на сходинку вище до створення кросбраузерних та кросплатформових веб-додатків.

ВІДОМОСТІ ПРО АВТОРІВ:

ПЛЕЧИСТИЙ Дмитро Дмитрович, кандидат технічних наук, доцент, доцент кафедри комп'ютерної інженерії Житомирського державного технологічного університету. Наукові інтереси: сучасні веб-технології, оптимізація обчислень. E-mail: pzs.ztu@gmail.com.

ІВАНІЮК Роман Леонідович, магістрант групи ПІ-41М кафедри програмного забезпечення систем факультету інформаційно-комп'ютерних технологій Житомирського державного технологічного університету. Наукові інтереси: робота з графікою, Інтернет-технології.