

## РОЗШИРЕННЯ БАЗОВИХ МОЖЛИВОСТЕЙ ЕЛЕМЕНТУ CANVAS ЗА ДОПОМОГОЮ БІБЛІОТЕКИ FABRIC

Волинець Н.С.<sup>1)</sup>, Кодола Г.М.<sup>2)</sup>

ДВНЗ Український державний хіміко-технологічний університет

<sup>1)</sup> викладач; <sup>2)</sup> викладач

### I. Постановка проблеми

Елемент Canvas (від англ. canvas – «полотно»), що використовується разом з мовою програмування JavaScript надає широкі можливості для розробників різних web-додатків, включаючи ігри, освітні ресурси, побудову графіків і анімації.

Останнім часом популярність Canvas зростає. Інтернет розробники вигадують все більше нових застосувань цього елемента.

Canvas – це растровий холст, на якому можна малювати через спеціальний API для JavaScript, який надає базові можливості: малювання примітивів і тексту, побітовий доступ до зображення, виведення зображень і афінні перетворення контексту малювання [1].

Головна проблема у роботі з Canvas полягає в тому, що його API занадто низького рівню. Для відображення на сторінці простих фігур цього набору функцій цілком достатньо, але коли з'являється потреба в інтерактивності, перетворенні зображення в будь-який момент або малювання більш складних фігур, виникає потреба в більш зручному функціоналі.

Canvas – растровий, малювання відбувається за допомогою вбудованих методів, які змінюють растр всього холста, і пам'ятає тільки свій останній стан у вигляді масиву пікселів. Не можна отримати доступ до намальованих ліній як до окремих об'єктів, які можна було б повернути. Можна побачити тільки останній зліпок, і для того, щоб намалювати щось нове (оновлення екрану), потрібно перемальовувати все (через очищення екрану або поверх).

Тому почали з'являтися бібліотеки, написані мовою JavaScript, які дуже розширюють базові можливості Canvas. Одна з таких бібліотек – Fabric. Вона надбудовує об'єктну модель над методами низького рівня Canvas, зберігає стан холста і дозволяє працювати з фігурами як з об'єктами.

### II. Мета роботи

Дослідити можливості використання бібліотеки Fabric на прикладі створення та управління простим об'єктом. Показати відмінність використання вбудованих методів Canvas від управління об'єктом, створеного за допомогою бібліотеки Fabric. Визначити основні можливості бібліотеки Fabric.

### III. Особливості програмної реалізації об'єктів за допомогою бібліотеки Fabric

Якщо в звичайному Canvas доводиться працювати з контекстом, то в Fabric можна управляти саме об'єктами – створювати, змінювати параметри, додавати їх на Canvas.

Розглянемо різницю у роботі з Canvas і Fabric. Створимо квадрат зі стороною 50 пікселів, синього кольору, повернений на 45 градусів (рис. 1).

Якщо використовувати тільки вбудовані методи Canvas, це буде виглядати так:

```
var canvasEl = document.getElementById('c');
var ctx = canvasEl.getContext('2d');
ctx.fillStyle = 'blue';
ctx.translate(100, 100);
ctx.rotate(Math.PI / 180 * 45);
ctx.fillRect(-25, -25, 50, 50);
```

За допомогою бібліотеки Fabric приклад створення об'єкта-квадрата:

```
var canvas = new fabric.Canvas('x');
var rect = new fabric.Rect({ left: 100,
top: 100,
fill: 'blue',
```

```
width: 50,  
height: 50,  
angle: 45 });  
canvas.add(rect);
```

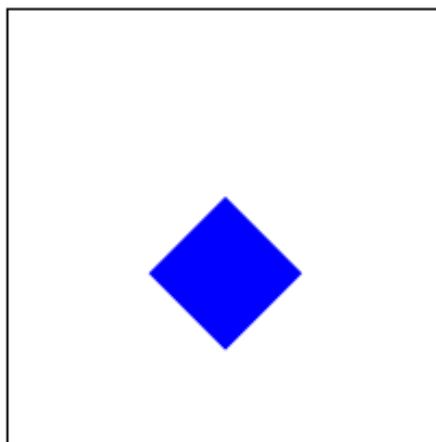


Рисунок 1 – Відображення квадрата

Тут спочатку створюється «оболонка» навколо Canvas-елементу з `id='x'`. Далі створюється об'єкт «прямокутник» з заданими характеристиками. Потім додається цей елемент на Canvas.

Використовуючи Fabric, для того щоб зробити повернення прямокутника необхідно тільки змінити значення кута на 45. Звичайними методами не можна управляти об'єктами безпосередньо і тому, спочатку доводиться міняти позицію і кут самого холста (`ctx.translate`, `ctx.rotate`). Потім малювати прямокутник, при цьому не забуваючи відсунути холст відповідно (-25, -25), так, щоб прямокутник з'явився на 100,100. Ще треба перевести кут з градусів в радіани при повороті холста.

Тобто використовуючи бібліотеку Fabric, просто працюють з об'єктами, змінюючи їх атрибути, а потім перемальовують Canvas, щоб побачити зміни. Таким чином можна змінити десятки об'єктів, і в кінці однією командою оновити екран.

Відзначимо основні можливості бібліотеки Fabric [2]:

- трансформації об'єктів не перемальовуючи весь холст;
- розширена кількість базових фігур: коло, еліпс, багатокутник, лінія, полілінія, прямокутник і трикутник;
- динамічна зміна будь-яких параметрів об'єктів за допомогою методу `set()`;
- надання можливості користувачу маніпулювати об'єктною моделлю;
- надання можливості групувати об'єкти та керувати ними як одним цілим;
- спрощена можливість анімації всіх об'єктів;
- спрощена робота з зображеннями та можливість застосування фільтрів до них;
- спрощена можливість роботи з текстом і градієнтами;
- розширена обробка подій;
- функції для експорту зображення з Canvas в формати JSON, SVG і навпаки – відтворення зображень на холсті з цих форматів;
- можливість розширення об'єктної моделі шляхом додавання власних підкласів;
- можливість включення режиму вільного малювання для користувача;
- можливість використання не лише на клієнтському браузері, але і на сервері Node.js

### Висновок

Досліджені можливості використання бібліотеки Fabric на прикладі створення та управління простим об'єктом. Показана відмінність використання вбудованих методів Canvas від управління об'єктом, створеного за допомогою бібліотеки Fabric. Визначені основні можливості бібліотеки Fabric.

### Список використаних джерел

1. Fulton S., Fulton J. HTML5 Canvas. Sebastopol: O'Reilly Media, Inc., 2011. – 749 p.
2. Not familiar with Fabric? [Електронний ресурс]. Режим доступу: <http://fabricjs.com/docs/> – 13.04.2017. – Назва з екрану