책에서는 맛볼 수 없는 HTML5 Canvas 이야기 (부제: Web Worker를 이용해 캔버스 성능을 극대화하기)

- https://deview.kr/2018/schedule/259
- https://www.slideshare.net/deview/122-html5-canvas

```
<iframe src="//www.slideshare.net/slideshow/embed_code/key/zxWYaoZqux7aRN" width="595" height="485" frameborder="0" marginwidth="0" marginheight="0" scrolling="no" style="border:1px solid #CCC; border-width:1px; margin-bottom:5px; max-width: 100%;" allowfullscreen> </iframe> <div style="margin-bottom:5px"> <strong> <a href="//www.slideshare.net/deview/122-html5-canvas" title="[122]책에서는 맛볼 수 없는 HTML5 Canvas 이야기 " target="_blank">[122]책에서는 맛볼 수 없는 HTML5 Canvas 이야기 </a> </strong> from <strong><a href="https://www.slideshare.net/deview" target="_blank">NAVER D2</a></strong> </div>
```

HTML5를 선도하였던 기술 <canvas> 에 대해서 다룹니다. 캔버스는 등장 이후 급속도로 성장해왔고, 이제 브라우저에서 빠질 수 없는 기능이 되었습니다. 전체 웹 페이지의 30% 이상이 캔버스를 사용하고 있으며 캔버스의 성능개선은 웹 개발자에게 매우 중요한 과제가 되었습니다.

본 세션에서는 캔버스의 성능을 개선할 수 있는 새로운 방법들에 대해서 다룹니다. 특히 최근 Chrome 69에 추가 된 `OffscreenCanvas API`를 사용하여 렌더링에 걸리는 부하를 메인 스레드에서 Web Worker로 위임함으로써 캔버스를 개선할 수 있는 방법과 내부 구현 원리에 대해 상세히 설명합니다. 나아가 국내에서는 잘 알려지지 않았지만 브라우저 내부에 구현되었던 몇몇 API들과 앞으로 캔버스가 나아갈 방향 (웹 표준과 브라우저 구현 관점에서)를 공유합니다.

목차

- 1. Motivation
  - HTML5 Canvas의 등장
  - 웹 분야에서 Canvas의 위치
  - 많은 개선에도 불구하고 Canvas가 느린 이유
- 2. DOM과 Canvas 애니메이션의 차이점
  - Retained Mode VS Immediate Mode
  - DOM animation VS Canvas animation
  - 장점과 단점
- 3. 기존의 성능 개선 사례 및 문제점 분석
  - 기존 사례1. < canvas >와 drawImage를 이용한 성능 개선
  - 기존 사례2. Canvas Proxy와 Canvas In Workers
  - 기존 사례3. WebGL in Web Worker
  - 기존의 문제점은 무엇인가?
- 4. 새로운 OffscreenCanvas API 적용하기
  - 이미지 복사 최소화 (Zero-copy image)
  - Trasnferrable ImageBitmap을 활용한 Main thread 동기화 방식
  - Compositor에 직접 commit하는 방식
  - 기존 대비 개선 된 성능 비교

- 5. 사례 연구
  - Three.js in a worker
  - Multiple WebGL views
  - W3C WebXR spec
  - Google Map
- 6. 그 밖에 알려지지 않은 API 및 연구과제들
  - Path2D
  - HitRegion
  - Color Space
  - Input Events in Workers
- 7. 브라우저 구현 현황

대상

- 성능 향상이 필요한 Canvas2D 개발자
- WebGL / WebXR (AR + VR) 개발자
- Web Front-End 개발자
- 웹 표준 및 브라우저 기술에 관심 많은 개발자
- 기타 모든 개발자

웹,, 모바일,, 성능

speaker

- 방진호삼성전자 / Web Platform Developer
- 삼성전자에서 삼성 인터넷 브라우저를 개발 중인 평범한 개발자입니다. 오픈소스를 사랑하여 Chromium/Blink를 비롯해 W3C, Node.js 다양한 웹 플랫폼 관련 프로젝트에 기여를 하고 있습니다.

관련 문서

- Deveiw 2018

From:
http://jace.link/ - **Various Ways**

Permanent link:
http://jace.link/open/%EC%B1%85%EC%97%90%EC%84%9C%EB%8A%94-%EB%A7%9B%EB%B3%BC-%EC%88%98-%EC%97%86%EB%8A%94-html5-canvas-%EC%9D%B4%EC%95%BC%EA%B8%B0

Last update: **2020/06/02 09:25**