

모바일 스타일링 자료조사 수집

목차

모바일 스타일링 자료조사 수집.....	1
목차	2
1. 목적	3
2. 스타일링 성능 테스트 방법.....	3
3. 화면전환(Page Transition).....	5
4. Bootstrap 최적화 대상 요소 파악 및 방법	13
5. 참조	15

1. 목적

- Mobile에서 jQuery Mobile보다 향상된 Page Transition 방안 조사
- PC(IE9++)에서도 가능한 Page Transition 효과 조사
- Mobile보다 PC UI에 중심적인 Bootstrap을 Mobile 환경에서 최적화 방법
- 스타일링에서 전체적인 성능 향상을 위한 조사

2. 스타일링 성능 테스트 방법

체감적으로 테스트하는 것은 정확한 결과를 얻기 어렵다는 판단으로 보다 객관적이고 정확한 결과를 얻기 위해 Google Speed Tracer를 통해 테스트를 진행한다.

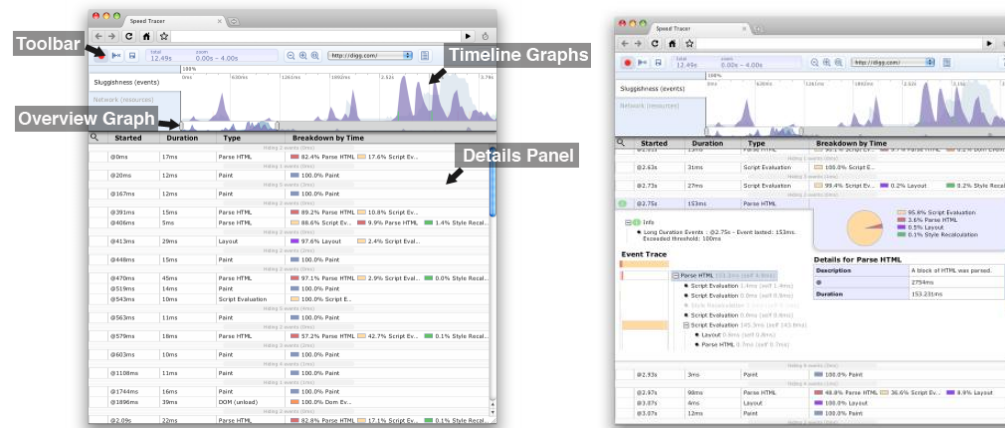


그림. 2-1

Speed Tracer(스피드 트레이서)는 웹 애플리케이션에서 성능 문제를 해결하는데 필요한 도구이다. 이 것은 브라우저의 내부 로우 레벨 수준에서 애플리케이션을 분석하고 통계를 시각적으로 보여준다. Speed Tracer(스피드 트레이서)는 크롬을 설치할 수 있는 OS에서 크롬 플러그인 형태로 쉽게 사용할 수 있다.

※ 스피드 트레이서 사이트 : <https://developers.google.com/web-toolkit/speedtracer/>

※ 크롬 플러그인 : https://chrome.google.com/webstore/detail/ognampngfcbddbfemdapefohjjobgbd?utm_source=chrome-ntp-icon

아래 표. 2-1은 애플리케이션 성능에 영향을 끼칠 수 있는 항목들이다.

항목	설명
JavaScript callback	JavaScript was run in an event dispatch.
Layout	The browser's rendering engine performed layout calculations.
CSS style recalculation	The renderer recalculated CSS styles.
DOM event handling	A top level DOM event fired, such as mousemove or DOMContentLoaded fired.
XMLHttpRequest	The handler for an XMLHttpRequest ran. Check the state field to see if this is an intermediate state or the last call for the request.
Painting	The browser's rendering engine updated the screen.
Parse Html	A block of HTML was parsed.
Garbage Collection	The JavaScript engine ran its garbage collector to reclaim memory.

표. 2-1

※ Speed Tracer(스피드 트레이서) 도움말 링크 : <https://developers.google.com/web-toolkit/speedtracer/get-started?hl=ko-KR>

3. 화면전환(Page Transition)

화면전환 테스트는 1차적으로 jQuery Mobile, jqMobi, Custom를 개별 테스트하고 2차적으로 상대적 테스트로 진행한다.

1) 개별테스트

A. 테스트 방법 : Flip 효과를 기준으로 1분간 20~30회 시도

B. jQuery Mobile (테스트 URL : <http://dropbox.jnwlab.com/page-transition/compare/jquery-mobile/docs/pages/page-transitions.html>)

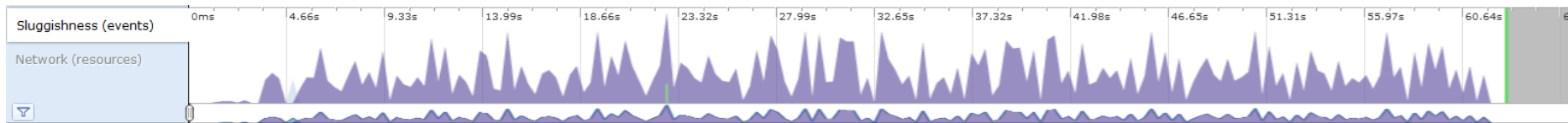


그림. 3-1

Summary Report for Selection: 0ms - 65.96s (65.96s)



71.4%	UI Thread Available (47.11s)
8.8%	Style Recalculation (5.79s)
7.2%	Paint (4.73s)
6.2%	JavaScript Callback (4.06s)
5.9%	Layout (3.92s)
0.2%	Garbage Collection (144ms)
0.2%	Dom Event (103ms)
0.1%	Parse HTML (95ms)
0.0%	XMLHttpRequest (5ms)
0.0%	Timer Fire (3ms)

Hints

All Rule **Severity**

Info

Time	RuleName	Description
22.75s	Long Duration Events	Event lasted: 129ms. Exceeded threshold: 100ms

그림. 3-2

그림. 3-1은 1분간 jQuery Mobile에서 Flip효과를 사용한 경우 Sluggishness 얼마나 성능에 영향을 주는지를 보여주는 차트이다.

x축은 시간, y축은 Sluggishness이며 jQuery Mobile은 y축이 높은 편이므로 성능이 좋지 않다는 결과를 얻을 수 있다.

그림. 3-2에선 표. 2-1에서의 항목들 중 어떤 항목이 1분간 jQuery Mobile의 성능에 영향을 끼쳤는지를 순위, 점유율(%) 및 시간(s, ms)으로 확인 할 수 있다

C. jqMobi (테스트 URL : <http://dropbox.jnwlab.com/page-transition/compare/jqmobi/index.html#jqmtransitions>)

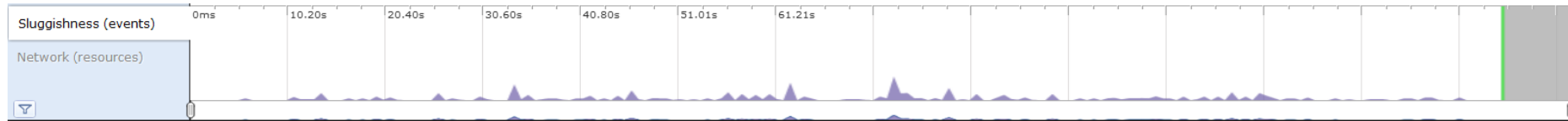


그림. 3-3

Summary Report for Selection: 0ms - 66.10s



92.3%	UI Thread Available (61.02s)
4.2%	Style Recalculation (2.80s)
1.7%	Paint (1116ms)
1.2%	JavaScript Callback (770ms)
0.4%	Layout (283ms)
0.1%	Dom Event (95ms)
0.0%	Parse HTML (12ms)
0.0%	Garbage Collection (7ms)
0.0%	Timer Fire (2ms)

Hints

All Rule **Severity**

그림. 3-3은 1분간 jqMobi에서 Flip효과를 사용한 경우 Sluggishness 얼마나 성능에 영향을 주는지를 보여주는 차트이다.

x축은 시간, y축은 Sluggishness이며 jqMobi은 jQuery Mobile에 비해 y축이 높지 않으므로 성능이 좋다는 결과를 얻을 수 있다.

그림. 3-4에선 표. 2-1에서의 항목들 중 어떤 항목이 1분간 jqMobi의 성능에 영향을 끼쳤는지를 순위, 점유율(%) 및 시간(s, ms)으로 확인 할 수 있다

그림. 3-4

D. Custom (테스트 URL : <http://dropbox.jnwlabs.com/page-transition/compare/index.html>)

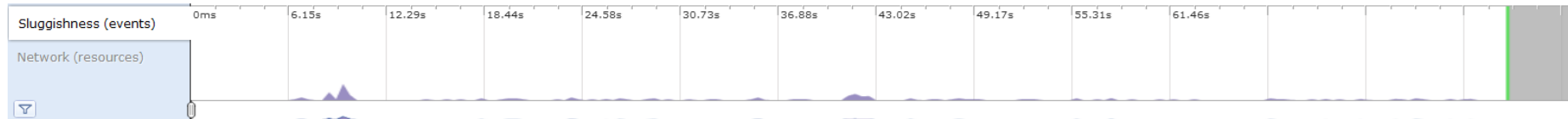


그림. 3-5

Summary Report for Selection: 0ms - 64.78s



96.4% UI Thread Available (62.43s)
2.3% Style Recalculation (1518ms)
0.6% JavaScript Callback (416ms)
0.4% Paint (275ms)
0.1% Layout (61ms)
0.1% Garbage Collection (41ms)
0.0% Dom Event (28ms)
0.0% Parse HTML (10ms)

Hints

All Rule **Severity**

그림. 3-5은 1분간 Custom에서 Flip효과를 사용한 경우 Sluggishness 얼마나 성능에 영향을 주는지를 보여주는 차트이다.

x축은 시간, y축은 Sluggishness이며 Custom은 jQuery Mobile 그리고 jqMobi보다도 y축이 높지 않으므로 성능이 좋다는 결과를 얻을 수 있다.

그림. 3-6에선 표. 2-1에서의 항목들 중 어떤 항목이 1분간 Custom의 성능에 영향을 끼쳤는지를 순위, 점유율(%) 및 시간(s, ms)으로 확인 할 수 있다

그림. 3-6

E. 비교

항목	jQuery Mobile	jqMobi	Custom
Sluggishness	상 (UI Thread Avail. 80%미만)	하 (UI Thread Avail. 90%이상)	하 (UI Thread Avail. 90%이상)
UI Thread Available	71.4%	92.3%	96.4%
Style Recalculation	8.8%	4.2%	2.3%
Paint	7.2%	1.7%	0.4%
Javascript Callback	6.2%	1.2%	0.6%
Layout	5.9%	0.4%	0.1%
Garbage Collection	0.2%	-	0.1%
Dom Event	0.2%	0.1%	-
Parse HTML	0.1%	-	-
XMLHttpRequest	-	-	-
Timer Fire	-	-	-

표. 3-1

개별테스트의 경우 각 라이브러리에서 제공하는 데모페이지를 기본으로 테스트를 했기 때문에 비교 테스트가 큰 의미는 없다. 개별테스트에선 각 항목의 수치가 어떤 의미를 갖는지 간단히 설명하며, 비교테스트는 다음 장인 상대테스트에서 자세히 설명하도록 한다.

표. 3-1에서 Sluggishness(y축)는 낮을수록 성능이 좋고, 이하 항목들 중 UI Thread Available의 비중이 높을수록 성능이 좋다. 상대적으로 UI Thread Available보다 아래 항목들에 비중이 높아지면 성능이 좋다고 볼 수 없다. jQuery Mobile과 jqMobi를 비교해봐도 그 차이를 확인 할 수 있다.

2) 상대 테스트

A. 테스트 방법 : Flip 효과를 기준으로 1분간 20~30회 시도

B. No Transition

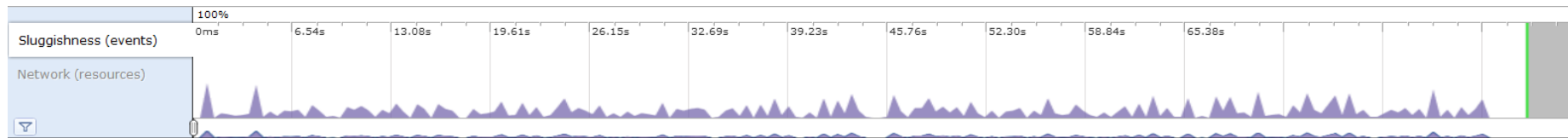


그림. 3-7

Summary Report for Selection: 0ms - 64.89s (64.89s)

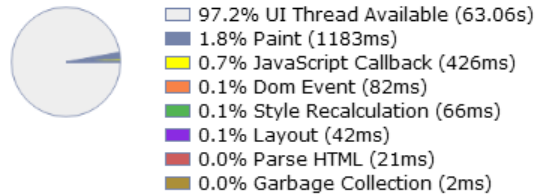


그림. 3-8

그림. 3-7은 1분간 Custom에서 화면전환 효과 없는 상태에서 Home과 Subpage를 이동하면서 UI 자원을 얼마나 사용했는지 기준점을 찾기 위한 차트이다.

대체로 Paint와 Javascript Callback

C. jQuery Mobile

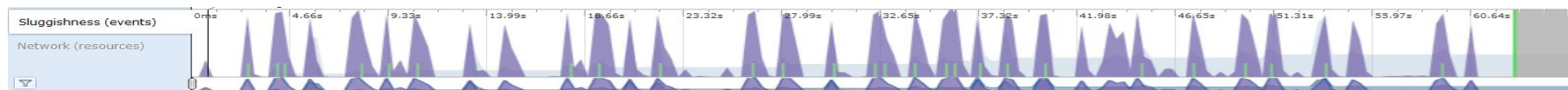


그림. 3-9

Summary Report for Selection: 0ms - 65.96s

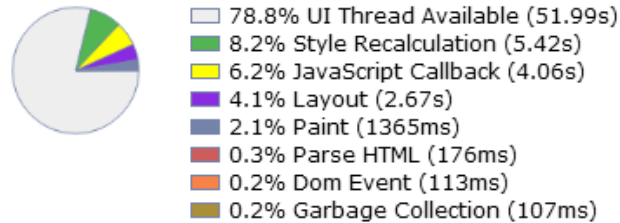


그림. 3-10

D. Custom(Flip)

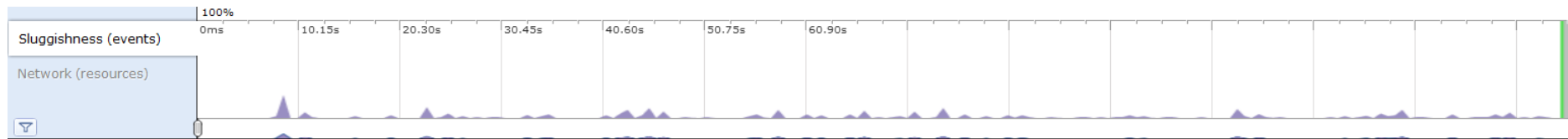


그림. 3-11

Summary Report for Selection: 0ms - 65.77s (65.77s)

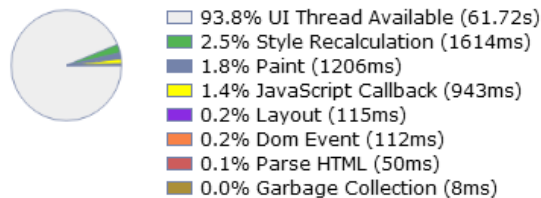


그림. 3-12

그림. 3-9은 jQuery Mobile의 경우 각 페이지마다 레이아웃 뿐만 아니라 Button같은 Component까지도 상당히 많은 수의 클래스가 추가된다 클래스가 많다는 것은 그만큼 styling을 재해석한다는 것을 뜻하며 그림. 3-10에서 얻은 결과에서도 styling 재해석 부분이 가장 많은 UI 자원을 사용하고 있다는 것을 알 수 있다.

그림. 3-11은 1분간 Custom에서 Flip 화면전환 효과로 Home과 Subpage를 이동하면서 UI 자원을 얼마나 사용했는지 보여주는 차트이다.

차트에서 알 수 있는 것처럼 그래프의 y축이 높지 않아 효율적이라는 것을 알 수 있다. 좌측에 그림. 3-12에서 UI Thread Available가 93%이상 남은 것을 확인 할 수 있다.

E. Custom(Slide)

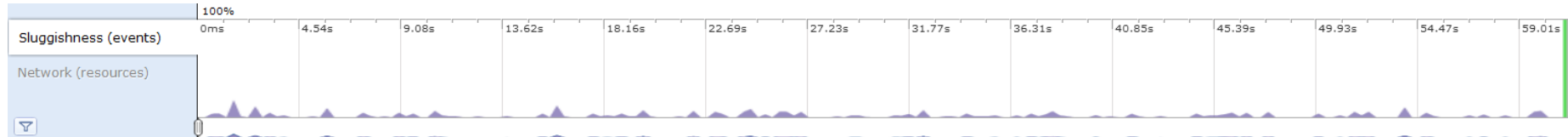


그림. 3-13

Summary Report for Selection: 0ms - 64.18s (64.18s)

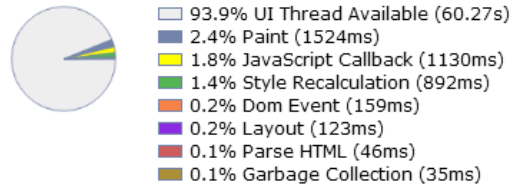


그림. 3-14

D. Custom(Flip)과 Garbage Collection을 제외하곤 동일한 효율을 보여주고 있다.

F. 비교

항목	None Transition	jQuery Mobile	Custom(Flip)	Custom(Slide)
Sluggishness	기준점	상	하	하
UI Thread Available	97.2%	78.8%	93.8%	93.9%
Style Recalculation	0.1%	8.2%	2.5%	1.4%
Paint	1.8%	2.1%	1.8%	2.4%

Javascript Callback	0.7%	6.2%	1.4%	1.8%
Layout	0.1%	4.1%	0.2%	0.2%
Garbage Collection	-	0.2%	-	0.1%
Dom Event	0.1%	0.2%	0.2%	0.2%
Parse HTML	0.1%	0.3%	0.1%	0.1%

표. 3-2

위 표. 3-2에서 None Transition 열의 데이터를 기준으로 jQuery Mobile과 Custom(Flip/Slide)의 UI 자원 사용량을 확인 할 수 있다.

기준이 되는 None Transition은 화면전환 효과가 없으므로 UI 대기 자원이 97% 이상 많이 남은 것을 알 수 있다. Flip / Slide 효과의 경우 화면전환 효과를 주기 위해 None일 때 보다 다소 UI 대기 자원이 줄긴 했지만 둘 역시 93% 이상의 높은 수치를 보여주고 있다. 이처럼 jQuery Mobile을 사용하지 않고 오직 CSS3 -webkit-animation 기능만으로도 좋은 반응 속도를 얻을 수 있다고 판단된다.

위와 같은 결과를 얻기 위해 CSS3 webkit-animation에 다음과 같은 Tip을 적용함.

- 1) 화면전환시 깜박임 현상 : -webkit-backface-visibility: hidden;
- 2) 안드로이드에서 보다 부드럽게 노출 : -webkit-transform: translate3d(0, 0, 0);
- 3) CSS3만으로 animation 구현 위해 keyframes 또는 transition 사용 :

```
@-webkit-keyframes slide-in{
  from {-webkit-transform: translateX(100%);}
  to {-webkit-transform: translateX(0);}
}
```

Transition 테스트 사이트 : http://www.w3schools.com/css3/tryit.asp?filename=trycss3_transition2

CSS Transition과 jQuery Animation 비교 사이트 : <http://css3.bradshawenterprises.com/demos/speed.php>

4. Bootstrap 최적화 대상 요소 파악 및 방법

분류	최적화 대상요소	최적화 방안
성능	Hover 문제 : 클릭 / 터치 이벤트를 발생 시키는 모든 태그에서 생기는 문제로 hover가 된 후 반응하므로 체감상 무겁고, 느리다는 느낌을 받음.	터치기반 디바이스의 경우 Hover 부분을 제거.
성능	Touch 기반에서 반응속도가 느린 문제	방안1) Creating Fast Buttons for Mobile Web Applications

- Hover 및 Tutch 반응속도

```

Thu Jul 05 2012 16:46:34 GMT+0900 (KST) Thu Jul 05 2012 16:46:34 GMT+0900 (KST) "소요시간 = 0.26s"
hover
mousedown
mouseup
click
Thu Jul 05 2012 16:46:36 GMT+0900 (KST) Thu Jul 05 2012 16:46:36 GMT+0900 (KST) "소요시간 = 0.267s"
hover
mousedown
mouseup
click
Thu Jul 05 2012 16:46:39 GMT+0900 (KST) Thu Jul 05 2012 16:46:39 GMT+0900 (KST) "소요시간 = 0.306s"
hover
mousedown
mouseup
click
Thu Jul 05 2012 16:46:41 GMT+0900 (KST) Thu Jul 05 2012 16:46:41 GMT+0900 (KST) "소요시간 = 0.269s"

```

그림. 5-1 HOVER(O) CSS된 경우(0.2~0.3초)

```

hover
mousedown
mouseup
click
Thu Jul 05 2012 16:49:08 GMT+0900 (KST) Thu Jul 05 2012 16:49:07 GMT+0900 (KST) "소요시간 = 0.052s"
hover
mousedown
mouseup
click
Thu Jul 05 2012 16:49:09 GMT+0900 (KST) Thu Jul 05 2012 16:49:09 GMT+0900 (KST) "소요시간 = 0.073s"
hover
mousedown
mouseup
click
Thu Jul 05 2012 16:49:11 GMT+0900 (KST) Thu Jul 05 2012 16:49:11 GMT+0900 (KST) "소요시간 = 0.119s"
hover
mousedown
mouseup
click
Thu Jul 05 2012 16:49:13 GMT+0900 (KST) Thu Jul 05 2012 16:49:13 GMT+0900 (KST) "소요시간 = 0.056s"
hover
mousedown
mouseup
click
Thu Jul 05 2012 16:49:15 GMT+0900 (KST) Thu Jul 05 2012 16:49:15 GMT+0900 (KST) "소요시간 = 0.048s"

```

그림. 5-2 HOVER(X) CSS된 경우(0.05~0.1초)

```

touchstart
onTouchStart
hover
mousedown
mouseup
click
onClick
click
Thu Jul 05 2012 16:55:29 GMT+0900 (KST) Thu Jul 05 2012 16:55:29 GMT+0900 (KST) "소요시간 = 0.032s"
touchstart
onTouchStart
hover
mousedown
mouseup
click
onClick
click
Thu Jul 05 2012 16:55:31 GMT+0900 (KST) Thu Jul 05 2012 16:55:31 GMT+0900 (KST) "소요시간 = 0.028s"
touchstart
onTouchStart
hover
mousedown
mouseup
click
onClick
click
Thu Jul 05 2012 16:55:32 GMT+0900 (KST) Thu Jul 05 2012 16:55:32 GMT+0900 (KST) "소요시간 = 0.056s"
touchstart
onTouchStart
hover
mousedown
mouseup
click
onClick
click
Thu Jul 05 2012 16:55:34 GMT+0900 (KST) Thu Jul 05 2012 16:55:34 GMT+0900 (KST) "소요시간 = 0.032s"

```

그림. 5-3 HOVER(O) CSS + FastButton된 경우(0.03~0.5초)

```

hover
mousedown
mouseup
click
onClick
click
Thu Jul 05 2012 16:52:09 GMT+0900 (KST) Thu Jul 05 2012 16:52:09 GMT+0900 (KST) "소요시간 = 0.029s"
touchstart
onTouchStart
hover
mousedown
mouseup
click
onClick
click
Thu Jul 05 2012 16:52:11 GMT+0900 (KST) Thu Jul 05 2012 16:52:11 GMT+0900 (KST) "소요시간 = 0.045s"
touchstart
onTouchStart
hover
mousedown
mouseup
click
onClick
click
Thu Jul 05 2012 16:52:13 GMT+0900 (KST) Thu Jul 05 2012 16:52:13 GMT+0900 (KST) "소요시간 = 0.035s"
touchstart
onTouchStart
hover
mousedown
mouseup
click
onClick
click
Thu Jul 05 2012 16:52:16 GMT+0900 (KST) Thu Jul 05 2012 16:52:16 GMT+0900 (KST) "소요시간 = 0.033s"

```

그림. 5-4 HOVER(X) CSS + FastButton 된 경우(0.03~0.5초)

	BTS hover(O)	BTS hover(X)	FastButton	평균 소요시간	체감 의견
그림. 5-1	O	X	X	0.2 ~ 0.3초	Hover 후에 click 발생으로 상당히 느리다고 느껴짐.
그림. 5-2	X	O	X	0.05 ~ 0.1초	반응 속도가 그림5.1에 비해 상당히 빨라짐.
그림. 5-3	O	X	O	0.03 ~ 0.5초	그림 5.2와 체감상 차이는 크지 않다.
그림. 5-4	X	O	O	0.03 ~ 0.5초	그림 5.3와 체감상 차이는 크지 않다.

※ Highlight 제거 부분(-webkit-tap-highlight-color:rgba(255, 255, 255, 0))은 적용 전 후 성능의 차이는 없다.

표. 5-1 테스트 결과 비교 테이블

5. 참조

1) 서적

- *Javascript Web Application*
- *jQuery Mobile: Up and Running by Maximiliano Firtman*
- *Pro jQuery Mobile by Brad Broulik*

2) 도구

- *Google Speed Tracer*
- *Adobe Shadow*