

—
마이크로인터랙션

Microinteractions

Microinteractions

by Dan Saffer

© 2015 Insight Press

Authorized Korean translation of the English edition of MICROINTERACTIONS ISBN 9781449342685 © 2013 Dan Saffer.

This translation is published and sold by permission of O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

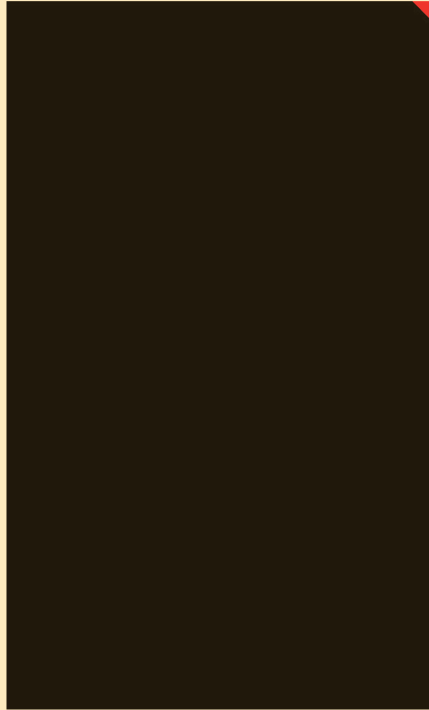
이 책의 한국어판 저작권은 에이전시 원을 통해 저작권자와의 독점 계약으로 인사이트에 있습니다. 저작권법에 의해 한국 내에서 보호를 받는 저작물이므로 무단전재와 무단복제를 금합니다.

마이크로인터랙션

디테일에 강한 인터랙션 디자인

전자책 1쇄 발행 2022년 12월 8일 (종이책 초판 2쇄 반영) 지은이 덴 새퍼 옮긴이 정승녕 펴낸이 한기성 펴낸곳 (주)도서출판인사이트
편집 조은별 등록번호 제2002-000049호 등록일자 2002년 2월 19일 주소 서울시 마포구 연남로5길 19-5 전화 02-322-5143 팩스
02-3143-5579 블로그 <https://blog.insightbook.co.kr> 이메일 insight@insightbook.co.kr ISBN 978-89-6626-381-3

UX
insight



UX
insight

마이크로인터랙션

디테일에 강한 인터랙션 디자인

맨 새퍼 지음 | 정승병 옮김

차례

	추천의 글	viii
	옮긴이의 글	xiv
	머릿글	xviii
1	마이크로인터랙션 디자인	1
	작지만 중요한 요소	6
	작지만도 않은 마이크로인터랙션	7
	마이크로인터랙션의 숨겨진 역사	12
	마이크로인터랙션의 구조	18
	디자인 철학으로서의 마이크로인터랙션	26
	요약	30

2	트리거	31
	수동 트리거	35
	정보를 앞당겨 표시하기	37
	트리거의 구성 요소	38
	시스템 트리거	55
	시스템 트리거의 동작규칙	59
	요약	61
3	동작규칙	63
	동작규칙 설계	68
	동작규칙 결정	73
	동사와 명사	76
	화면과 상태	78
	동작규칙의 제약점	80
	백지에서 시작하지 말 것	81
	복잡성을 포용하기	85
	제한된 선택 항목과 세심한 기본값	87
	조작 장치와 사용자 입력	92
	오류 예방	95
	마이크로카피	98
	알고리즘	101
	요약	105

4	피드백	107
	피드백은 동작규칙을 부각시킨다	111
	사람을 위한 피드백	116
	적은 편이 더 낫다	119
	개성을 표현하는 수단으로서의 피드백	120
	피드백의 방법	123
	시각적 피드백	123
	음향 피드백	131
	햅틱 피드백	136
	피드백의 동작규칙	137
	요약	139
5	순환과 모드	141
	모드	145
	용수철 모드와 일회성 모드	147
	순환	149
	순환의 특성	150
	장기적 순환	153
	요약	157
6	마이크로인터랙션 프로젝트의 사례	159
	사례1: 모바일 앱	162
	사례 2: 재생 목록의 온라인 공유	169
	사례 3: 식기세척기 조작 패널	174

		180
		183
		184
		186
		188
부록A	마이크로인터랙션의 검증	191
		194
		196
		199
		201

추천의 글

내가 마이크로인터랙션에 대한 댄 새퍼의 관심을 처음으로 알게 된 것은 브라질에서 열린 학회에서였다. 나는 그 개념에 바로 매료되었다. 댄은 관현악단의 연주회 중에 울린 휴대폰의 사례로 이야기를 시작했다. (이 사례는 이 책의 1장에서 자세히 소개된다.) 작은 것에 주의를 기울임으로써 뭔가 아주 커다란 것을 발견한 것이다.

그 후 나는 애플의 사진 정리/편집 어플리케이션인 Aperture를 사용하다가 느낀 당혹스러운 순간을 통해서, 디자인에는 디테일이 중요하다는 사실을 다시 느낄 수 있었다. 새로 출판할 책에 들어갈 그림을 모아 정리하던 중에 컴퓨터가 멈춰버려서 리부팅을 할 수밖에 없었는데, 다시 Aperture를 열려고 하니까 데이터베이스에 문제가 있다는 메시지만 표시하고는 그대로 종료되는 것이다. 아니, 아무런 해결법을 알려 주지 않을 거라면 오류 메시지가 무슨 소용이 있는가? 내가 뭘 어떻게 하면 되는 거지?

프로그램의 도움말 파일을 뒤져 봤지만 별 도움이 되지 않았다. 애플의 고객 지원 웹사이트를 검색해도 도움되는 내용은 없었다. 난 짜증이 나고 걱정도 되기 시작했다. 도대체 어떻게 하면 내 사진들을 되

찾을 수 있는 거지? 프로그램은 도통 열리지 않았다. 다른 컴퓨터에서 백업 파일을 찾아봤지만, 내가 쓰던 파일 동기화 프로그램이 너무 효과적으로 동작하는 바람에 문제가 있는 데이터베이스를 그대로 다른 컴퓨터들에도 모두 덮어써버린 후였다.

결국 한참을 고생한 끝에, 인터넷 검색을 통해서 해결 방법을 찾을 수 있었다. 애플에서 잘 정리해 놓은 일련의 방법을 찾아낸 것이다. 지시된 내용을 한 15분 정도 그대로 따라하고 나서야 모든 사진을 복구할 수 있었다. (내가 그 방법을 애플 웹사이트에서 찾은 게 아니라는 점은 말해줄 만하다. 애플 웹사이트 구석에 올라와 있던 내용을 누군가 온라인 토론 모임에 올려둔 것을 찾은 것이다.)

내가 왜 이 이야기를 하고 있냐고? 그건 애플의 프로그래머가 이 책을 읽었더라면 내가 그 고생을 하지 않아도 됐을 것이라고 생각하기 때문이다. 마이크로인터랙션, 즉 디테일을 제대로 만드는 게 중요하다.

왜 처음 나왔던 오류 메시지에서는 문제 뿐만 아니라 해결 방법까지도 전달해주지 않은 걸까? 무엇보다 애플은 그 문제에 대한 상세한 설명은 물론 해결 방법까지도 갖고 있었다. 오류 메시지가 “데이터베이스에 문제가 있습니다. 문제를 해결하려면 다음 지시사항을 따라 하세요.”라는 내용과 함께 다음 단계로 바로 실행할 수 있는 버튼을 제공해 줄 수도 있었을 것이다. 왜 그러지 않았을까? 프로그램의 이 부분을 개발한 프로그래머들이 그런 메시지를 표시하는 게 자기 책임이 아니라고 생각했기 때문일까? 그 프로그래머들이 데이터베이스를 관리하는 부서와는 다른 부서에 소속되어 있어서, 문제가 있는 건 알았지만 해결할 방법은 몰랐던 걸까? 아니면 오류 메시지를 작성하던 사람에게 오류에 대한 해결 방법도 함께 제공해야 한다는 생각이 없었던 것일까? (내 생각으로는 이 세 가지 요인이 모두 작용한 것으로 보인다.) 이

유가 뭐든 간에, 결과적으로 나온 것은 열악한 사용자 경험이다. 이 경험은 내가 그 제품을 극단적으로 싫어하게 만들었고, 이를 대체할 다른 소프트웨어를 찾게 했다. 이런 반응은 애플이 고객들에게 얻고자 한 것이 아닐 것이다. 애플에서 Aperture를 개발할 때 이 책이 없었던 사실은 아쉬울 뿐이다.

마이크로인터랙션이 디테일인가? 물론이다. 마법은 모두 디테일 안에 있다.

마이크로인터랙션에는 ‘마이크로’라는 말이 들어 있어서 뭔가 작은 것에 대한 개념이라는 것을 암시한다. 마이크로인터랙션은 실제로 작은 것에 대한 개념이지만, 작다는 게 중요하지 않다는 뜻은 절대로 아니다! 마이크로인터랙션은 친근한 경험과 끔찍하고 불안한 경험을 구분 짓는 중요한 디테일에 대한 이야기다. 댄 새퍼가 머릿글에서 지적 하듯이, 디자이너는 큰 그림을 제대로 정립하고 싶어 한다. 큰 그림을 확고하게 잡아가는 것은 멋진 일이다. 그 어떤 큰 문제도 해결할 수 있을 것 같은 기분도 든다. 하지만 큰 그림을 올바르게 잡아냈다고 하더라도, 디테일을 제대로 다루지 않는다면 그 문제에 대한 해결안은 실패하게 된다. 순간순간의 경험을 조정하는 것은 바로 디테일이다. 사람들이 제품을 막힘 없이 사용할 수 있게 해주는 것도 시의적절하게 제공되는 디테일이다. 이와 반대로, 그런 디테일에 충분한 주의를 기울이지 않은 제품은 사용자에게 혼란과 짜증을 유발하며, 결국은 그 제품을 정말로 싫어하게 만든다. 큰 그림은 물론 중요하다. 하지만 작고 세부적인 그림도 그만큼 중요하다. 주의 깊게 디자인된 디테일이야말로 사용자에게 매끄러운 성취감을 느끼게 해준다.

훌륭한 마이크로인터랙션을 만드는 데에는 몇 가지 단계가 있다. 우선은 상황을 적절하게 파악하는 것으로, 많은 개발자가 가장 어려워하는 부분이다. 이 단계는 상당한 관찰력을 필요로 한다. 다른 사람이나

자기 자신이 제품을 사용하는 방식을 관찰하면서 문제점을 찾아내고, 그 논리적인 흐름을 이해함으로써 어떻게 마이크로인터랙션을 구성할 수 있을지를 결정하는 것이다. 흔하게 생각할 수 있는 것은 오류 메시지나 대화창이다. 이런 요소들은 보통 어떤 정보를 보여 주고, 그를 통해서 다음에 해야 할 행동을 암시한다. 다음 단계를 지금 단계에 아예 포함시켜서 제공하는 건 어떨까?

훌륭한 마이크로인터랙션 디자인은 그 제품을 사용하는 사람들에 대한 이해를 필요로 한다. 사람들이 하고자 하는 일이 무엇인가? 어떤 단계를 거쳐야 하는가? 대상 인터랙션의 맥락도 이해해야 한다. 사용자에 대한 공감을 형성하고, 그들의 행동을 제대로 관찰함으로써 제품이 갖는 다양한 측면(아마도 여러 개발 팀 혹은 여러 부서로부터의 산출물일 것이다)을 적절하게 조합하여 하나의 부드러운 마이크로인터랙션으로 만들어 내기 위해서는 다른 무엇보다도 맥락을 알아야 한다. 이런 작업의 진행 방식에 대한 원칙이 이 책의 1장에 잘 소개되어 있다. 이 책 전반에 걸쳐 인용되는 수많은 사례들은 마이크로인터랙션이 제공하는 기회를 일깨워준다. 그리고 나서 끊임없는 관찰을 통해서 새로운 마이크로인터랙션의 기회를 발견하는 것은, 이 책을 읽는 독자의 몫이다. 어떤 해결안이 회사의 여러 부서를 관여시켜야 하는 경우가 생기더라도, 그 이유로 마이크로인터랙션을 포기해서는 안 된다. (애플의 개발자들은 분명 그랬던 것 같다.) 사용자를 위해서 올바른 결정을 내리는 일이 결국은 훌륭한 제품을 만드는 방법인 것이다.

훌륭한 마이크로인터랙션을 위한 두 번째 단계는 그 구현에 있다. 마이크로인터랙션에는 많은 디자인 과제가 있다. 트리거, 동작규칙, 피드백, 순환과 모드. 이 주제에 대해서는 책에서 각각 하나의 장을 할애하여 잘 설명하고 있다.

마이크로인터랙션이 중요한가? 내가 최근에 구입한 자동차를 예로 들어 보자. 이 자동차에 가까이 가서 문 손잡이 근처에 손을 갖다 대면, 손잡이 주변에 불이 들어오고 실내등이 켜진다. 차 문을 열면 운전석 위로 조명이 켜진다. 다른 승객이 차에 타려고 문을 열면 그쪽 좌석 위의 조명이 켜진다. 운전석에 타면 좌석, 거울, 그리고 심지어 라디오 프로그램까지도 내가 좋아하는 설정으로 맞춰진다. 내 아내와 내가 교대로 운전을 하면, 그때그때 운전하는 사람에 맞춰 설정을 바꿔준다. 자동차 디자이너는 어떻게 이 일련의 반응들을 정했을까? 어떤 부분을 조정하고 어떤 부분을 하지 않을지는 어떻게 정했을까? 바로 지능적으로 면밀하게 디자인된 마이크로인터랙션이다. 자잘한 것들이라고? 그렇다. 사람이 직접 조정할 수도 있다고? 물론이다. 하지만 자동차가 사용자를 위해 이런 일을 해주면, 차 안에는 즐거운 분위기가 감돈다. 심지어 그 차를 소유하고 있다는 것에 대한 자부심마저 느끼게 된다. 모든 제품의 제조사에서 그들의 고객이 느끼기를 바라는 것이 바로 이런 기분 아닐까?

우리가 삶의 대부분을 함께 보내는 작은 것들에 환호를 보낸다. 물건을 사용하다가 다음 단계로 넘어가려면 어떻게 해야 하는지, 조잡하게 제시된 맥락 때문에 느꼈던 당혹스러움의 그 모든 순간에게도 환호를 보낸다. 그리고 댄 새퍼의 친근한 글이 방대한 사례를 만나 의미를 더한 이 책에도 환호를 보낸다. 나는 스스로 사람들이 기술을 사용하는 모습을 관찰하는 데에 일가견이 있다고 생각했다. 하지만 이 책을 읽고 나서 내 그런 능력은 한층 향상되었다. 이제 나는 디테일을 좀 더 면밀하게 보고, 이를 좀 더 개선할 수 있는 방법을 고민한다. 나는 또한 제품이 제대로 동작하는 순간을 알아보게 됐다. 보는 법을 배우는 것은, 뭔가를 개선하는 첫 번째 단계다.

이제 여러분의 차례다. 세상에 나아가 문제를 해결하라. 마이크로인

터랙션을 이해하고 실무에 적용함으로써, 우리의 삶을 보다 단순하고 즐겁게 만들어 보자.

도널드 노먼

(실리콘 벨리에 위치한 노먼 그룹의 대표이며,
『디자인과 인간심리』를 저술했다. don@jnd.org)

움긴이의 글

사용자 경험 디자인 분야는 지난 수년간 큰 변화를 겪어왔다. 특히 모바일 앱이나 웹사이트를 다루는 실무 디자이너라면, 최근 많은 사람들이 이야기하는 린 UX(혹은 린 스타트업) 방식을 프로젝트에 적용해 보거나 애자일 소프트웨어 개발 방법론을 채용한 팀과 협업하면서 약간의 의구심을 가졌을 법도 하다.

소위 말하는 ‘최소 요건 제품’^{Minimum Viable Product, MVP}, 즉 단기간에 핵심 기능에 대한 시장의 반응을 볼 수 있도록 최소한도로 정의된 제품을 개발자나 프로젝트 관리자 등 다른 사람들과 함께 정의하고 디자인하고 개발해서 출시하고, 실제 사용 데이터를 참고해서 이를 올바른 방향으로 개선해 나간다. 이런 방식은 분명 기존에 없던 새로운 제품을 하나 만들어 성공시키는 데에는 가장 확실한 접근 방법이다. 또한 보다 큰 조직의 경우에도 같은 방식을 채용하면 대규모의 개발 과제를 적은 규모로 나누고, 각 단계에서의 검증을 통해서 효율적으로 제품을 개발할 수 있다.

그런데, 이런 프로젝트에 참여하다 보면 디자이너로서의 역할과 그 범위를 다시 생각해보지 않을 수 없다. 이런 업무 환경에서의 디자인

이런 다양한 전문분야의 팀원들이 모여 아이디어를 모으는 브레인스토밍에 가깝고, 이를 통해 골라낸 아이디어는 따로 시간을 들여 ‘디자인’되고 문서화되기보다 그렇게 논의를 통해 합의한 대로 프로토타입으로 만들어지는 경우가 대부분이다. 프로토타입의 디자인 자체를 검증하는 ‘반복적 디자인 프로세스’를 내부적으로 적용할 수 있는 기회는 거의 주어지지 않고, 기본적인 기능이 완성되는 대로 우선 출시한 후 사용자의 반응과 사용 데이터를 참고하여 개선점에 대한 가설을 세우고 제품을 개선하게 된다.

위와 같은 새로운 업무 방식에서 디자이너는 모종의 디자인 방법론을 통해서 팀에 디자인 산출물을 제공하는 것을 넘어서, 팀 구성원 모두가 참여하는 디자인 토론을 주도적으로 이끌어야 한다. 다양한 배경과 전문성을 지닌 팀원들의 논의가 산발적인 난상토론으로 결론 없이 끝난다거나, 목소리가 크거나 직위가 가장 높은 사람이 원하는 대로 흘러가지 않게 하려면, 디자인에서 고려해야 할 모든 측면을 체계적으로 검토할 수 있는 통찰을 제시해야 하는 것이다.

상대적으로 적은 규모의 조직에서라면, 디자이너는 시장 분석이나 제품 기획에서 시작해서 세부적인 디자인과 구현 그리고 개발 후의 검증까지에도 기여하면서 그야말로 팔방미인이 되어야 한다. 반대로 다양한 전문성을 갖춘 여러 부서와의 협업을 통해 제품을 만들어야 하는 보다 큰 규모의 기업에서는, 디자이너의 직접적인 책임은 실제 디자인을 제공하는 영역에 대부분 집중되게 된다. 그렇지만 그 어떤 경우라 할지라도, 디자이너의 책무는 언제나 결국은 ‘좋은 디자인’을 제공하는 것이다.

이 책은 우리 디자이너들이 어떤 환경에서 일하든 잊지 말아야 할 그 구심점을 되새기게 해준다. 좋은 사용자 경험 디자인은 주어진 정보와 기술을 최대한 활용하여 사용자가 필요한 기능은 물론, 사용자가

미처 모르고 있던 부분까지 깊이 배려함으로써 만족과 감동을 준다. 디자이너의 역할과 책임의 범위를 어떻게 정의하든 간에, 디자인의 세부적인 영역 즉 ‘마이크로인터랙션’은 온전히 디자이너의 책임으로 남는 부분이며, 최종적으로 제품이 출시되었을 때 그 디자인이 좋은 디자인인지 아닌지를 말하게 되는 근거가 되는 것이다.

마이크로인터랙션은 언뜻 생각하면 분석적인 접근이나 체계적인 디자인 방법론의 대상이라기보다 디자이너의 재치에서 나오는 번뜩이는 아이디어로 여겨지기 쉽다. 하지만 이 책의 저자는 마이크로인터랙션의 구조를 정리하고, 그 구성요소 하나하나를 다루는 데 있어서의 고려사항과 접근 방법을 다양한 제품의 사례와 함께 설명하면서 마이크로인터랙션 디자인 또한 체계적이고 분석적인 접근이 가능하다는 것을 보여 준다.

이제까지 세부적인 부분에 깊은 고려가 배어 있는 ‘좋은 디자인’의 제품을 발견할 때마다 “와 이거 정말 기발하네!”라고 감탄했다면, 이 책에서 제시하는 마이크로인터랙션의 구조를 숙지함으로써 그런 디자인을 보다 분석적으로 이해할 수 있을 것이다. 직접 디자인 업무를 진행함에 있어서도, 디자인하고 있는 제품을 마이크로인터랙션의 구성요소에 비추어 조목조목 뜯어보고 그에 대한 고려사항을 하나씩 반영해 본다면 단지 기본 기능을 실험하기 위해 최소한도로 개발된 제품이 아니라 주어진 제약하에서 세심하게 완성된 디자인을 가진 제품을 출시하고 그 성공률을 높이는 데에 기여할 수 있으리라 생각한다.

사용자 경험 디자인은 그 역할과 범위에 대한 대내외적인 논의가 끊이지 않는 분야지만, 그럼에도 변하지 않은 진리는 서양 속담에서 말하듯 “악마는 사소해 보이는 데에 숨어 있다.”^{Devil lies in detail} 거나 광고에 등장했던 “작은 차이가 명품을 만든다”는 것이다. 이 책이 그런 사소해 보이는 부분을 찾아내고 체계적으로 개선하여 꾸준히 ‘좋은 디자인’을

만들어 내는 디자이너가 되는 데 기여할 수 있다면 옳긴이로서 더 바랄 나위가 없겠다. 그렇게 좋은 디자인을 약속할 수 있는 디자이너가, 결국 ‘좋은 디자이너’이기 때문이다.

2015년 4월 에딘버러에서 정승녕

머릿글

이 책이 담고 있는 내용

우리가 가전제품의 전원을 켜는 순간에서부터 온라인 서비스에 로그인하거나 모바일 앱으로 날씨 정보를 받는 순간에 이르기까지, 마이크로인터랙션은 주변 어디에서나 찾아볼 수 있다. 마이크로인터랙션은 딱 한 가지 일을 수행하는 하나의 사용 목적을 가지며, 독자적인 앱이 될 수도 있고 복잡한 기능의 일부가 될 수도 있다. 잘 만들어진 마이크로인터랙션은 효율성, 재치, 스타일 그리고 사용자의 니즈와 목표에 대한 이해를 바탕으로 기능을 수행한다. 사람들이 사랑하는 제품과 그 저 참가하며 사용할 뿐인 제품의 차이도, 그 제품에 담긴 마이크로인터랙션에 달려 있는 경우가 많다.

이 책은 독자의 디자인 작업에 도움을 주기 위해서 마이크로인터랙션을 분석한다. 먼저 마이크로인터랙션의 모델을 정의하고 그 모델을 각 부분별로 상세하게 분석하고 나서, 이를 바탕으로 마이크로인터랙션을 제대로 활용할 수 있는 지침과 원칙을 제공한다. 이런 관점을 디자인 업무에 적용하게 되면, 제품은 점차 나아지고 사용자들은 그 제품을 더욱 좋아하게 되어 (브랜드에 대한) 고객의 충성도가 높아질 것이다.

대상 독자

이 책은 보다 나은 제품, 특히 디지털 제품을 만드는 데 관심이 있는 모든 사람을 위한 내용을 담고 있다. 모든 분야의 디자이너들은 물론 개발자, 연구자, 제품 관리자, 비평가, 기업가도 이 책에서 한번쯤 생각하고 적용해 보거나 시도할 만한 내용을 찾아내기 바란다.

이 책은 특히 세밀한 부분에 신경을 쓰는 것이 정말 중요하며, 그런 세부사항이 제품을 훨씬 더 좋게 만든다는 사실을 함께 일하는 클라이언트, 개발자, 제품 관리자, 프로젝트 관리자에게 설득시키기 힘든 사람을 위해 쓰여졌다. 이제 그런 세밀한 부분들에 ‘마이크로인터랙션’이라는 이름을 부여함으로써, 그 중요성을 보다 효과적으로 주장할 수 있게 되었다.

책의 구성

이 책은 작지만 중요한 주제인 마이크로인터랙션에 대해서 간략하게 소개하고 있다.

■ 1장. 마이크로인터랙션 디자인

이 장에서는 마이크로인터랙션을 소개하고 아주 하찮아 보이는 것들이 왜 그렇게 중요한지를 설명한다. 모든 마이크로인터랙션이 따르는 일반적인 유형을 개괄적으로 파악하면서 그 구조를 논의한다. 끝으로, 이 장은 어떻게 하면 마이크로인터랙션을 프로젝트에 포함시킬 수 있는지를 고찰한다.

■ 2장. 트리거

마이크로인터랙션이 시작되는 순간인 ‘트리거¹’를 소개한다. 사용자에게 의해 시작되는 수동 트리거와 시스템 트리거를 둘 다 검토

¹ (옮긴이) 저자는 일련의 마이크로인터랙션이 일어나게 되는 계기가 되는 입력을 은유적으로 방아쇠, 즉 트리거라고 명명하고 있다. 인터랙션 디자인이나 프로그래밍 분야에서 종종 사용되는 용어이므로 우리말로 번역하지 않고 사용했다.

하고, ‘정보를 앞당겨 표시할 것’ 원칙을 설명한다.

- 3장. 동작규칙

이 장에서는 마이크로인터랙션을 정의하는 특성과 숨겨진 변수들, 즉 ‘동작규칙’에 대한 논의를 제공한다. 동작규칙은 어떻게 만들어지며 어떤 내용이 포함되어야 하는가? ‘백지에서 시작하지 말 것’ 원칙도 함께 소개한다.

- 4장. 피드백

사용자가 동작규칙을 이해하게 되는 방법인 ‘피드백_{feedback}’에 대해서 이야기한다. 피드백을 사용해야 하는 순간과 시각, 청각, 촉각의 세 가지 피드백 유형을 설명하고, ‘인간을 고려할 것’과 ‘간과하기 쉬운 요소를 활용할 것’ 원칙을 소개한다.

- 5장. 순환과 모드

이 장에서 논의되는 ‘순환_{loops}과 모드_{modes}’는 마이크로인터랙션의 구조에서 상위에 있는 개념이다. 순환과 모드의 다양한 유형과 함께, 오랜 기간에 걸친 순환을 활용하는 방법도 설명한다.

- 6장. 마이크로인터랙션 프로젝트의 사례

이제까지 설명한 마이크로인터랙션의 요소들을 모두 합쳐 마이크로인터랙션의 세 가지 사례를 디자인해 본다. 각각의 사례는 모바일 앱, 온라인 서비스, 가전제품을 다룬다. 마이크로인터랙션들을 한데 모아서 기능으로 구성하는 일도 이 장에서 논의한다.

- 부록 A. 마이크로인터랙션의 검증

끝으로, 마이크로인터랙션을 검증하는 절차에 대해서 간략하게 설명한다.

왜 마이크로인터랙션에 대한 책을 쓰게 됐나?

지난 십 년 동안 디자이너들은 ‘불명확한 문제’를 해결하고 기업이나 정부 시스템의 거대하고 조직적인 문제를 다루기 위해서, ‘디자인 사고 방식’^{Design Thinking}을 적용하고 큰 관점으로 생각하라는 소리를 들어왔다. 세상의 어떤 큰 문제도 디자인의 방법론을 적용할 수 있고, 디자인의 역할은 조직의 구조조정부터 도시 관리에 이르기까지 모든 것과 관련될 수 있다.

이렇듯 디자인의 접근 방식을 재조명하는 것에 대한 효과는 명확하지 않지만, 그렇게 큰 규모로 작업하다보면 디자인에 있어서 중요한 부분(사용자를 기쁘게 하는 세부사항)을 잊게 되기도 한다. 우리가 사랑하는 제품은 아름다운 곡선, 딸깍하는 기분 좋은 소리, 이해하기 쉬운 구조 등 세밀한 부분까지 주의 깊게 디자인되어 있다.

마이크로인터랙션은 새로운 작업 방식이다. 대규모의 디자인 프로젝트처럼 큰 열개부터 시작하는 게 아니라, 작은 부분부터 사랑하듯이 조심스럽게 빚어 나가면서 전체를 이뤄가는 것이다. 이 방식은 디자이너들이 제법 잘 하는 방식으로, 즉각적이고 손에 잡히는 결과를 낼 수 있다. 하찮아 보이는 순간을 사용상의 즐거움으로 만들으로써 세상을 바꾸는 또 다른 방법이 될 수도 있다.

아름답고 제대로 동작하는 작은 것들은 즐거움을 준다. 그런 즐거움은 사용자의 것이기도 하지만 만든 사람의 즐거움이기도 하다. 그것을 이루기 위해서 많은 기술과 시간과 고민을 들여야 하더라도 마찬가지다. 이런 접근은 힘든 작업이며, ‘큰 문제’를 다루는 것만큼이나 훌륭한

일이다. 삶에 더 많은 즐거움을 원하지 않는 사람이 어디 있겠는가?

감사의 글

이 책에 담겨 있는 대부분의 그림은 'Little Big Details'라는 훌륭한 블로그에서 인용된 것이다. 'Little Big Details'의 편집자이자 수집가인 플로리스 데커와 앤드류 맥카시에게 깊은 감사 인사를 보낸다. 그렇게 잘 정리되어 있는 사례들이 없었더라면 이 책을 쓸 생각도 하지 못했을 것이다. 블로그의 운영자들은 물론, 그 웹사이트에 온갖 사례를 보내 준 수많은 분들에게도 큰 신세를 졌다. 각각의 화면을 찾아내 올린 사람을 알 수 있는 경우에는 이름을 함께 표시했다.

'Design A Day' 블로그의 운영자인 잭 포렛에게도 감사드리고 싶다. 그 블로그의 'In the Details' 분류에서 많은 예시를 인용했을 뿐만 아니라, 그런 예시들을 분석하는 방식 역시 오랫동안 나에게 영감을 줬으며, 이 책을 쓰는 간접적인 계기가 됐다.

로버트 라이만, 크리스토퍼 파에이, 대니 말리크, 닉 레미스, 데이브 호퍼, 빌 스콧, 스콧 엔슨 등 감수를 맡아 준 분들의 격려와 지혜, 빈틈 없는 검토도 이 책을 좀 더 좋게 만드는 데에 큰 도움이 됐다.

내가 지난번에 출판한 책은 그다지 눈부신 성과를 거두지 못했다. 그럼에도 불구하고 O'Reilly 출판사의 편집자인 매리 트레슬러와 다른 모든 분들이 한 번 더 책을 출판하기로 결정하고, 이 괴상한 주제에 대한 얇은 책에 일관적으로 성원을 보내주신 것에 대해서 감사드린다.

언제나 그렇듯이, 내가 함께 지내는 여성들(인간과 강아지)의 참여 성도 간과할 수 없다. 나는 TV가 있는 방의 한가운데 안락의자에 앉아서 글을 쓰기 때문에, 이 책은 우리 집의 인내심을 평가하는 기준이 되었다. 이 책을 그들에게 헌정한다.

끝으로, 그동안 함께 일하고 가르침을 준 여러 선생님들과 디자이너

들에게 감사의 표시를 하고자 한다. 그들은 나에게 제품의 세밀한 부분에 집중하는 것이 왜 중요한지를 (가끔은 반강제적으로) 알려 주었다. 그분들이 고안하거나 나에게 만들어 보라고 했던 작지만 독창적인 부분들이 우리가 만들던 제품을 실현하는 데에 큰 도움이 되었으며, 그런 개념이 바로 내가 이 책을 통해 전달하고자 하는 내용이라고 할 수 있다.

2013년 2월 샌프란시스코에서 댄 새퍼

1 마이크로인터랙션 디자인

“덩치가 크면 제대로 움직이지 않는 법이다.”

빅터 파파넥

결국 지휘자는 연주를 멈췄고, 사람들은 화를 내며 소리를 지르기 시작했다. 뉴욕 필하모닉 오케스트라는 말리 교향곡 9번의 느리고 잔잔한 아다지오 대목 막바지를 한창 연주하던 중이었다. 이 특별한 기회를 위해 수백 달러를 지불했을 대부분의 청중들은 온 신경을 집중하고 앉아서 한 시간이 넘는 연주를 마무리하는 정적이고 섬세한 대단원에 귀를 기울이고 있었다.

그때 일어난 일이다. 앞 줄 어딘가에서 누구나 알 수 있는 아이폰의 ‘마림바’ 알림음(실로폰 높은 음을 두드리는 소리)이 반복적으로 들리기 시작한 것이다. 알람 소리였다. 그 소리는 몇 번이고 반복되었다. 지휘자 알란 길버트가 오케스트라의 연주를 중지시켰지만, 알람은 계속 반복됐다. 결국 청중들이 그 휴대폰의 주인에게 소리를 지르기 시작했는데, 그 사람은 후에 그 관현악단의 오랜 고객이자 어느 회사의 나이 든 중역으로 밝혀졌고 악단으로부터 고객 ‘X’라는 칭호를 얻게 되었다.

2012년 뉴욕 타임스의 보도¹에 따르면, 고객 X의 회사에서는 그 일이 있기 하루 전에 원래 그가 쓰던 블랙베리 휴대폰을 아이폰으로 교체해 줬다. 고객 X는 공연이 시작되기 전에 음소거 스위치를 켜서 매너 모드를 실행시켰지만, 아이폰의 동작규칙에 따르면 매너 모드에서도 알람 소리가 울린다는 것을 몰랐던 것이다. 그렇기 때문에 그는 알람이 울리고 한참이 지날 때까지 그게 자신의 휴대폰에서 나오는 소리라는 것을 알아채지 못했다. 결국 자기 휴대폰이 울리고 있다는 것을 알고 알람을 껐을 때에는, 이미 연주를 망쳐버린 후였다.

다음날 이 소식이 퍼지면서, 인터넷은 신랄한 비판과 농담으로 넘쳐났다. 지휘자 대니얼 도르프는 트위터에 “만일을 대비해서, 휴대폰 벨 소리를 말려 교향곡 9번으로 바꿨습니다.”라는 글을 올리기도 했다. 여러 블로그에서 주장과 토론이 벌어졌고, 어떤 사람들은 매너 모드를 실행시켰을 때에는 모든 종류의 소리가 꺼져야 한다고 했다. 기술 분야 칼럼니스트인 앤디 이넷코는 2012년 1월에 올린 「아이폰 음소거 스위치의 기능에 대하여」라는 글²에서, “기계가 나름의 판단으로 잘못된 결정을 내려서 화가 나는 것보다, 사람이 뭔가 바보 같은 짓을 해서 스스로에 대해 화를 내는 편이 훨씬 낫다.”고 언급했다.

또 다른 사람들은 매너 모드를 실행시켰을 때에도 알람은 켜져야 한다고 생각한다(나도 이 편이 옳다고 생각한다). 애플 전문가인 존 그루버가 지적했듯이, “만일 음소거 스위치가 모든 소리를 나지 않게 한다면, 휴대폰이 매너 모드로 되어 있다는 사실을 잊고 잠들었다가 아침에 늦잠을 자는 사람이 매일 수천 명은 될 것이다.”

애플의 iOS 휴먼 인터페이스 가이드라인³ iOS Human Interface Guidelines에서는 이러한 매너 모드 동작 방식에 대하여 다음과 같은 근거를 대고 있다.

1 「Ringin' Finally Ended, but There's No Button to Stop Shame」(2012. 01. 12, Daniel J. Wakin, The New York Times)

2 (옮긴이) 원 글의 제목은 「Daring Fireball: On the Behavior of the iPhone Mute Switch」이며, 다음 웹사이트에 찾아볼 수 있다. <http://goo.gl/MEuKA>

예를 들어 극장에 간 사용자는 함께 있는 다른 사람들을 방해하지 않기 위해서 휴대기기를 음소거 모드로 설정할 수 있습니다. 이 상황에서도 사용자는 여전히 기기에 설치된 앱을 사용하고자 하지만, 전화벨이나 메시지 알림음 같이 기대하지 않았거나 직접 요청하지 않은 소리 때문에 놀라기를 바라지는 않습니다.

음소거 스위치를 켜 놓아도, 사용자가 분명한 의도를 가지고 직접 설정한 소리는 그대로 재생됩니다.

다시 말해서 아이폰의 동작규칙은 휴대폰을 매너 모드로 했을 때에는 사용자가 직접 설정한 소리는 그대로 두고, 그렇지 않은 소리(예: 문자 메시지 알림음, 전화 수신 벨소리)만 들리지 않게 하는 것이다. 다른 많은 규칙과 마찬가지로 이런 동작규칙도 감춰져 있으며, 스위치에 붙어 있는 조그마한 주황색 표시를 제외하면 매너 모드인지 아닌지를 말해 주는 화면 표시 같은 것도 없다. 만일 애플이 이것을 다른 동작규칙(음소거 스위치를 켜면 모든 소리가 꺼진다.)으로 바꾼다면 또 다른 규칙과 피드백 방식이 설계되어야 한다. 알람이 울릴 때에는 휴대폰이 진동해야 할까? 휴대폰이 매너 모드일 때 휴대폰을 켜면 화면이나 하드웨어의 LED 표시등에 어떤 표시가 계속 나타나야 할까? 휴대폰 매너 모드를 디자인하는 방법에도 여러 가지가 있다.

휴대폰 매너 모드는 마이크로인터랙션의 한 가지 사례다. 마이크로인터랙션은 제품에서 한 가지 사용 사례에 해당하는 짧은 순간, 혹은 한 가지만 수행하는 작은 기능이다(일례로 그림 1-1과 같은 경우를 들 수 있다). 마이크로인터랙션은 하나의 앱이나 기기 전체가 될 수도 있지만, 대부분의 경우엔 보다 큰 제품과 함께 혹은 그 일부로 존재한다. 마이크로인터랙션은 지루하고 잊혀지기 쉬운 순간이 될 수도 있고, 즐겁고 매력적인 경험이 될 수 있다. 설정을 하나 바꾸거나, 데이터나 기기를 동기화하거나, 비밀번호를 설정하거나, 전원을 켜거나, 로그인하

거나, 상태 메시지를 설정하거나, 어떤 것을 ‘좋아요’라고 표시하는 등의 행위를 할 때마다 우리는 마이크로인터랙션을 경험한다. 마이크로인터랙션은 모든 곳에 있다. 우리가 들고 다니는 기기, 집에 있는 가전 기기, 휴대폰이나 PC에 설치된 소프트웨어, 심지어 우리가 살고 일하고 있는 환경 속에도 포함되어 있다.

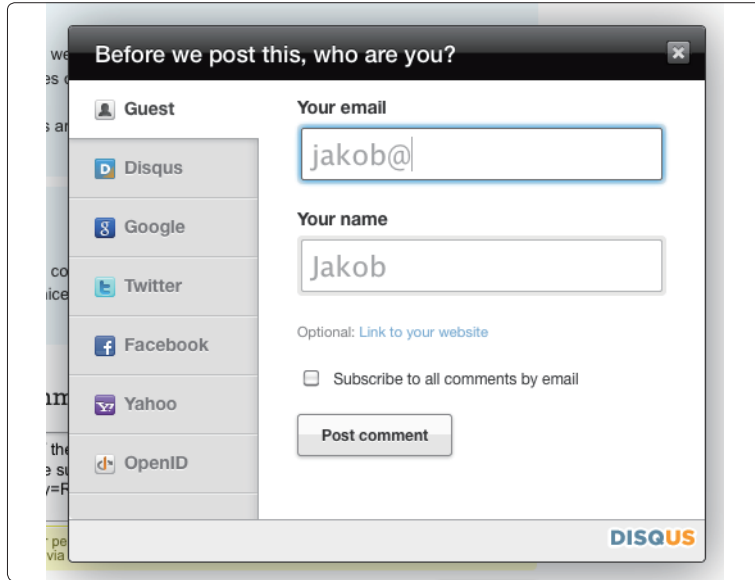


그림 1-1 일상적인 마이크로인터랙션의 사례: 회원 등록. Disqus의 회원 등록 양식은 입력된 이메일 주소를 바탕으로 이름을 추측해 낸다. (제공: Jakob Skjarning, Little Big Details)

마이크로인터랙션은 제품에서 기능적이고 인터랙티브한 세부사항들이다.³ 또한 찰스 임스(Charles Eames)의 명언⁴에도 나오듯이, 디테일들은 그저 디테일이 아니라 디자인 그 자체다. 사람들이 의식적으로는 기억하지 못하더라도, 디테일은 좀 더 쉽게 제품에 매력을 느끼고 즐겁게 사용하도록 만들 수 있다. 어떤 마이크로인터랙션은 사실상 (혹은 실제로) 눈에 보이지 않으며, 마이크로인터랙션 때문에 제품을 사는 경우도 드물다. 일반적으로 마이크로인터랙션은 기능의 일부거나, 소위 ‘완벽을 기하기 위해서’ 보조적으로 들어간 기능이다. 이를테면 음소

3 (윤건이) 디자인의 세부 사항(detail)은 업계에서 영문 그대로 ‘디테일’이라고 일컫는 경우가 많으며, 때로는 번역으로 전달하기 어려운 의미를 갖고 있어서 문맥에 따라 번역을 하지 않고 사용했다.

4 「100 Quotes by Charles Eames, Charles Eames,」 (Eames Office, 2007) / 「찰스 임스의 명언 100개」(임스 오피스, 2007)

거 기능을 보고 휴대폰을 사는 사람은 없지만, 이는 당연히 기대하는 기능이다. 그리고 앞서 소개한 사례에서도 알 수 있듯이, 마이크로인터랙션은 다양한 경험(좋은 경험이든 나쁜 경험이든)을 유발할 수 있다. 어떤 마이크로인터랙션은 당황스러울 수도 있고, 어떤 마이크로인터랙션은 지루하고 잊혀질 수 있다. 하지만 좋은 마이크로인터랙션은 매력적이고 독창적이다. 이 책에서 제시할 디자인 접근법은 바로 그런 종류의 마이크로인터랙션을 만들기 위한 것이다.

고객 X의 사례는 마이크로인터랙션이 기사화되는 흔치 않은 경우 중 하나다. 우리는 매일같이 마이크로인터랙션에 둘러싸여 살고 있지만, (고객 X의 경우에서 볼 수 있듯이) 뭔가가 엄청나게 잘못되기 전까지는 보통 그 존재를 인지하지 못한다. 하지만 그 작은 규모와 존재감에도 불구하고, 마이크로인터랙션은 엄청나게 중요하다. 사람들이 좋아하는 제품과 그저 별다른 수가 없어서 사용하는 제품의 차이는 대부분 그 제품이 제공하는 마이크로인터랙션에 있다. 잘 디자인된 마이크로인터랙션은 우리의 삶을 좀 더 쉽고, 재미있고, 흥미진진하게 만들어 줄 수 있다. 마이크로인터랙션을 잘 디자인하는 방법, 바로 이것이 책에서 다루고자 하는 주제다.

이 장에서는 마이크로인터랙션을 기능으로부터 구분하는 방법을 설명하고, 마이크로인터랙션의 역사를 간략하게 돌아본다. 그리고 나서 이 책의 나머지 부분에서는 마이크로인터랙션 구조에 대해서 자세히 알아볼 것이다. 마이크로인터랙션 모델은 마이크로인터랙션을 이루고 있는 모든 구성요소들을 분석하고 고찰할 수 있는 방법을 제공하며, 이를 통해 우리는 마이크로인터랙션을 디자인하거나 개선할 수 있는 방법을 알게 된다. 끝으로, 마이크로인터랙션을 실제 업무 과정의 일부로 포함시키는 방법에 대해서 이야기하게 될 것이다.

작지만 중요한 요소

잘 디자인된 마이크로인터랙션과 매크로인터랙션(macro-interaction, 즉 ‘기능’)의 조합은 강력한 힘을 가지며, 경험 디자인이 진정으로 추구하는 방향이기도 하다. 사용자가 제품을 쓰면서 좋은 경험을 할 수 있도록 하려면 큰 그림뿐만 아니라 세부사항에도 주의를 기울여야 한다(그림 1-2 참조).

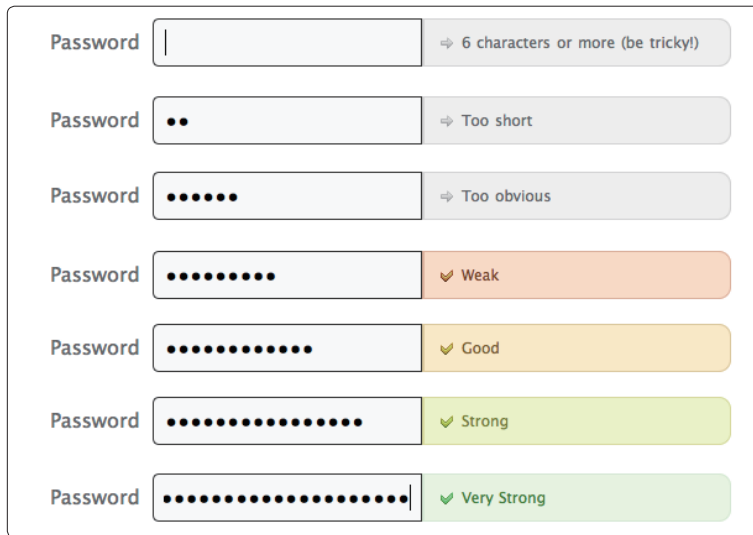


그림 1-2 트위터의 비밀번호 입력 양식은 피드백을 섬세하게 제공함으로써 일상적인 비밀번호 선택 마이크로인터랙션을 훌륭하게 바꾼 사례다. (제공: Little Big Details)

마이크로인터랙션은 그 규모로 보나 범위로 보나 기능과는 차이가 있다. 기능은 대체로 여러 사용 사례를 포괄하기 때문에 복잡하고, 오랜 시간 동안 사용하게 되며, 쓰면서 생각을 많이 해야 한다. 반면에 마이크로인터랙션은 짧고 단순하며 별다른 노력 없이 쓸 수 있다(그림 1-3 참조). 음악 플레이어는 하나의 기능이고, 음량을 조절하는 것은 그 기능에 포함된 마이크로인터랙션 중 하나다.

그림 1-3 페이스북 페이지에 올라온 글이 설정된 기본 언어로 적혀 있지 않으면, 자동으로 번역 기능이 표시된다. (제공: Marina Janeiko, Little Big Details)



마이크로인터랙션은 다음과 같은 경우에 적용하면 좋다.

- 하나의 작업을 수행할 때
- 기기들을 서로 연결할 때
- 단순한 정보를 주고 받을 때 (예: 주식 가격, 날씨)
- 진행 중인 기능을 조정할 때 (예: TV 채널 조정)
- 설정을 바꿀 때
- 간단한 내용을 작성하거나 확인할 때 (예: 상태 메시지)
- 어떤 기능을 켜거나 끌 때

작지만도 않은 마이크로인터랙션

마이크로인터랙션은 제품의 일부가 될 수도 있고 그 제품 전체가 될 수도 있다. 토스터를 예로 들어 보자. 토스터는 빵을 굽는 단 한 가지 동작만을 수행한다. 사용자가 구울 식빵을 토스터 안에 집어넣고 시작 버튼을 누른다. 토스터가 식빵을 굽는다. 잠시 후 다 구워진 식빵이 튀어나오고 끝난다. 물론 이 과정은 여러 가지로 달라질 수 있지만(이를 테면, 식빵 대신 베이글을 구울 수도 있다), 일반적으로 토스터라는 기기는 온전히 하나의 마이크로인터랙션으로 이루어졌다고 할 수 있다.

이와 비슷한 경우로, 간단한 앱이 단 하나의 마이크로인터랙션으로 이루어져 있을 수 있다. PC와 휴대기기에서 실행할 수 있는 수천 개의 앱들은 Convertbot처럼 수치를 다른 단위로 변환하거나(그림 1-4 참조), 계산을 하거나, 날씨 정보를 보여 주는 등 한 가지 단순 작업만을 제대로 수행하도록 되어 있다.



그림 1-4 Tapbot사의 Convertbot은 수치를 다른 단위로 변환해 주는 간단한 하나의 마이크로인터랙션을 수행하기 위해 만들어진 앱이다.

흔히 마이크로인터랙션은 제품을 디자인하고 개발하는 데 있어서 마지막으로 고려되며, 그러다 보니 종종 간과되곤 한다. 하지만 마이크로인터랙션을 무시하는 것은 큰 실수다. 모바일 운영체제인 안드로이드가 처음 나왔을 때 그렇게 덜 다듬어진 것처럼 느껴진 이유는 바로 마이크로인터랙션이 거칠었기 때문이었다. 이는 특히 아이폰과 비교되면서 뚜렷하게 드러났는데, 이를테면 항목을 삭제하는 방법이 일관적이지 않다든가, 어떤 앱에서는 검색 버튼⁵을 눌러도 아무 일이 생기지 않았다든가 하는 문제가 있었다. 마이크로인터랙션이 허술하다는

5 (옮긴이) 초창기 안드로이드 기기에는 하드웨어 자체에 검색 버튼이 달려 있어서 사용 맥락에 따른 검색 기능을 실행할 수 있었지만, 앱에서 검색 기능을 제공하지 않으면 아무 동작도 하지 않는 경우가 있었다.

것은 결국 핵심기능을 잘 만들어 놓고 불편함과 당혹스러움으로 그 기능을 덮어버리는 격이다. 제품의 디자인이 얼마나 잘 되어 있는지는 가장 작은 부분의 디자인에 달려 있다.

PC든 휴대기기든 상관없이 모든 운영체제가 하는 일은 애플리케이션을 설치하거나 실행하고, 파일을 관리하고, 소프트웨어와 하드웨어를 연결하고, 실행 중인 애플리케이션과 윈도우를 관리하는 일 등 기본적으로 동일하다. 하지만 사용자의 관점에서 봤을 때, 운영체제 사이의 차이는 쓰면서 매일, 혹은 심지어 매 시간 겪게 되는 마이크로인터랙션에 있다(그림 1-5, 1-6 참조).

그림 1-5 필자가 사용하는 OS X의 메뉴 바는 온갖 아이콘으로 가득 차 있으며, 각각의 아이콘을 클릭할 때마다 마이크로인터랙션이 시작된다.



물론 기능 중에는 아주 유용하거나 강력해서 (혹은 지적재산권으로 철저히 보호되어 있어서) 마이크로인터랙션이 별로 의미 없는 경우도 있다. 발전 단계의 초기에 있는 기술이나 의료기기의 경우에는 그 물건을 쓰는 방법보다 무슨 기능을 수행할 수 있는지가 더욱 의미 있는 부분이며, 따라서 마이크로인터랙션의 중요성이 상대적으로 적다. 예를 들어, 2002년 로봇청소기 ‘룸바^{Roomba}’가 처음 나왔을 때에는 방의 크기를 계산하거나 장애물과 먼지를 감지하는 기능이 없었다. 하지만 로봇청소기가 그 당시에는 새로운 기술이었기 때문에 그 자체로도 충분한 의미가 있었다. 그 이후의 모델들은 (무엇보다 시장에 경쟁 제품이 나온 다음이었기 때문에) 인간과 로봇 사이의 마이크로인터랙션에 좀 더 집중되어 있다.

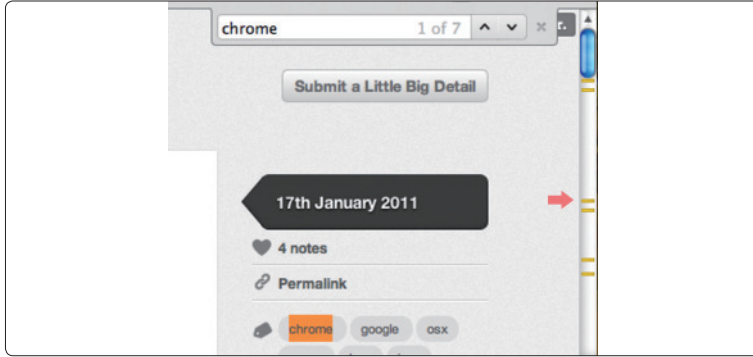


그림 1-6 크롬 웹 브라우저에서는 페이지 안에서 어떤 단어를 찾을 때, 그 단어가 나타나는 위치가 스크롤바에 표시된다. (제공: Saul Cozens, Little Big Details)

경쟁이 심한 시장의 경우, 마이크로인터랙션은 더욱 더 중요해진다. 기능상 동일한 제품이 있을 때, 소비자가 제품을 받아들이고 그 브랜드를 좋아하게 만드는 것은 제품과 관련된 경험이며, 어떤 제품에 대한 경험은 대체로 그 제품이 제공하는 마이크로인터랙션에 달려 있다. 제품을 보고 느낀다고 할 때는 ‘느낌’이 관여되는 법이다. 구글플러스^{Google+}가 페이스북^{Facebook}과의 경쟁에서 처참하게 실패한 이유는 마이크로인터랙션에 있었다. 친구들을 여러 개의 서클^{circles}로 분류하는 방식이 처음에는 흥미로울지 몰라도, 곧 피곤하고 쓸데없게 느껴졌던 것이다.

마이크로인터랙션에 주의를 기울여야 하는 또 다른 이유는, 여러 개의 정보 기기를 사용하는 상황에 아주 잘 들어맞기 때문이다. 마이크로인터랙션은 휴대용 기기, TV, PC와 노트북 컴퓨터, 가전 기기, 웹 등에서 따로따로 제공하는 기능을 하나의 맥락으로 엮어 주는 접착제 역할을 한다. 어떤 기술을 통해 제공되느냐에 따라 근본적인 차이는 있지만, 마이크로인터랙션은 크기가 작기 때문에 사용상의 일관성을 비교적 쉽게 유지할 수 있게 해준다. 이와 비교하면 서로 다른 기술을 통해서 구현된 기능들이 일관성을 갖기란 어려운 일이다. 한편, 화면이 작거나 아예 없는 가전제품과 휴대폰은 마이크로인터랙션을 위해 만들어진 것처럼 보일 정도다. 작은 인터랙션은 작은 기기에 매우 적합하다.