

Андрей Маркелов

Путеводитель по кнопкам

История интерфейсов от первых компьютеров до виртуальной реальности.
В поисках универсальной дизайн-системы



Москва, 2026

УДК 004.5
ББК 16.32г
М26

В книге упоминаются социальные сети Instagram и/или Facebook, принадлежащие компании Meta Platforms, Inc., деятельность которой по реализации соответствующих продуктов на территории Российской Федерации запрещена.

Маркелов А.

М26 Путеводитель по кнопкам. История интерфейсов от первых компьютеров до виртуальной реальности. В поисках универсальной дизайн-системы. — М. : Альпина ПРО, 2026. — 264 с. : ил.

ISBN 978-5-206-00610-0

Данная книга представляет собой всеобъемлющее руководство по созданию компьютерных интерфейсов. В отличие от типичных сборников правил и стандартов, она построена в формате исторического нарратива и раскрывает причинно-следственные связи, лежащие в основе современных интерфейсных решений. Это помогает читателю глубже понять принципы работы интерфейсов, а не просто запомнить готовые шаблоны. Книга состоит из трех частей. Первая посвящена настольным интерфейсам — от Xerox Alto и Apple Lisa до современных macOS и Windows. Вторая охватывает мобильные устройства — от Apple Newton до iPhone и Android. Третья рассматривает будущее интерфейсов: работу в виртуальной и дополненной реальности на примере Meta¹ Quest и Apple Vision.

В издание включены сотни иллюстраций, схем и скриншотов, а также примеры из 180 популярных приложений. Благодаря последовательному изложению, книга будет полезна как начинающим дизайнерам, так и профессионалам.

УДК 004.5
ББК 16.32г

Все права защищены. Никакая часть этой книги не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами, включая размещение в интернете и в корпоративных сетях, а также запись в память ЭВМ для частного или публичного использования, без письменного разрешения владельца авторских прав. По вопросу организации доступа к электронной библиотеке издательства обращайтесь по адресу mylib@alpina.ru.

О книге

Эта книга адресована широкому кругу читателей, интересующихся дизайном интерфейсов, цифровыми технологиями и историей компьютерной техники. Она будет полезна как новичкам, так и профессионалам.

Начинающим дизайнерам будет особенно интересно проследить, как привычные элементы интерфейса — кнопки, текстовые поля, чекбоксы, выпадающие меню — возникли из аналогов в физическом мире: электрических звонков, радиоприемников, бюллетеней для голосования и других устройств. Это не просто исторический экскурс, а способ по-новому взглянуть на повседневные вещи.

Профессиональным дизайнерам книга даст повод для глубоких размышлений о природе интерфейсных решений. Она помогает найти ответы на вопросы, которые кажутся простыми, но на практике вызывают затруднения: почему компонент устроен именно так? подсвечивать ли кнопку при наведении или затемнять? как правильно отмечать выбранный пункт — галочкой или цветом? чем отличается выпадающий список от выпадающего меню? как оптимально выровнять поля в форме?

В книге рассмотрены 180 интерфейсов популярных приложений — как десктопных, так и мобильных. В качестве иллюстраций, счет которых идет на сотни, используются как классические, так и современные популярные программы: Google Docs, Notion, Airbnb, Uber, Яндекс.Карты, Т-Банк, Telegram и многие другие. Дизайнеры часто сталкиваются с тем, что не могут найти хороший пример из популярного интерфейса, — книга решает эту проблему, формируя настроенность.

«Путеводитель по кнопкам» — не просто история развития интерфейсов, но и практическая инструкция по созданию современной дизайн-системы. Каждая глава посвящена одному компоненту и рассматривает его во всех деталях: от истории и эволюции до анатомии и универсальной матрицы всех видов, стилей и состояний.

Книгу делают уникальной несколько особенностей.

Во-первых, в книге содержится 35 QR-кодов со статьями, видеоуроками и дополнительными материалами. Это настоящая интерактивная энциклопедия. Читателю теперь не обязательно открывать поисковик, чтобы узнать значение термина или увидеть, как рассмотренный интерфейс работает вживую.

Во-вторых, все описанные в книге наработки собраны в реально существующую дизайн-систему, которая опубликована в сообществе Figma и используется в крупных международных проектах, над которыми работал автор.

В-третьих, структура книги такова, что сложные разделы — такие как матрицы компонентов — можно пропустить без ущерба для понимания и просто получать удовольствие от увлекательной истории и визуального материала.

Наконец, это единственная в мире книга с интермиссией.

Содержание

Часть 1. Настольные интерфейсы	8		
<i>Глава 1. Изобретение кнопки</i>	10	Поиск по списку	82
Скевоморф атакует	13	Анатомия списка	82
Античные кнопки	15	Матрица компонента	85
Состояния кнопок	18		
Главные и вторичные кнопки	19	<i>Глава 7. Выпадающий список</i>	86
Входит Фотошоп	22	Миф о дропдаунах	87
UI-киты и дизайн-системы	24	Правила выпадающих списков	89
Фотошоп уходит. Входит Фигма	26	Поиск по списку	92
Анатомия кнопки	27	Комбо-бокс	92
Основа	28	Прокрутка списков	95
Левая иконка	30	Список с мультивыбором	95
Правая иконка	31	Матрица компонента	98
Счетчик	32		
Состояния кнопки	32	<i>Глава 8. Выпадающее меню</i>	99
Режим фокуса	34	Способы вызова меню	100
Размеры	35	Бумажные и компьютерные меню	102
Матрица компонента	36	Анатомия меню	104
<i>Глава 2. Западающая кнопка</i>	39	Классификация секций меню	109
Матрица компонента	43	Трехмерные меню	109
<i>Глава 3. Чекбокс</i>	46	Левые и правые	111
Статусы	48	Поиск по меню	111
Стили	51	Матрица компонента	112
Анатомия чекбокса	52		
Матрица компонента	54	<i>Глава 9. Текстовое поле</i>	117
<i>Глава 4. Сегмент кнопок</i>	57	Летающий цирк Материал-дизайна	120
Сегмент с мультивыбором	61	Анатомия текстовых полей	122
Анатомия сегмента	62	Плейсхолдеры	123
Матрица компонента	65	Иконки	124
<i>Глава 5. Радиокнопка</i>	66	Пароли	125
Анатомия радиокнопки	70	Маска ввода	127
Инструкция по применению	72	Сегмент полей	128
Радиочекбокс	75	Подсказки и комбо-боксы	129
Матрица компонента	76	Текстовая область	131
<i>Глава 6. Список</i>	78	Матрица компонента	132
Список с мультивыбором	81		
		<i>Глава 10. Вкладки</i>	137
		Вкладки первого уровня	140
		Вкладки второго уровня	142
		Вкладки третьего и четвертого уровня	144
		Сегмент кнопок в роли табов	145

Вертикальные вкладки	147	Списки с мультивыбором	208
Необычные решения	148	Компоненты	208
Анатомия вкладок	150	<i>Глава 17. Текстовое поле</i>	210
Матрица компонента	151	Подсказки и комбо-боксы	212
<i>Глава 11. Другие компоненты</i>	153	Текстовые области	213
Переключатель	153	Компонент	214
Сплит-кнопка	155	<i>Глава 18. Перечень, меню и шторка</i>	215
Наборщик тегов	156	Способы вызова меню	217
Слайдер и диапазон	156	Редакт-меню	218
Календарь	157	Шторки	218
Интермиссия	160	Компоненты	219
Дизайн форм	163	<i>Глава 19. Вкладки</i>	221
Эквивалентность компонентов	166	Панель вкладок	221
Ловушка адаптации	168	Заминированный интерфейс	223
Часть 2. Мобильные интерфейсы	170	Вкладки первого уровня	224
<i>Глава 12. История мобильных интерфейсов</i>	172	Вкладки второго уровня	225
Ньютон	174	Вкладки третьего и четвертого уровня	226
Все остальные	179	Боковое меню	228
iPod	181	Компоненты	229
<i>Глава 13. Зоны досягаемости экрана</i>	183	<i>Глава 20. Ячейки. Интерфейс «Бенто»</i>	230
<i>Глава 14. Мобильный список</i>	186	Плитка	230
Анатомия пункта списка	191	Горизонтальный список	231
Матрица компонента	193	Назад к спискам	232
<i>Глава 15. Мобильная кнопка</i>	194	«Бенто»	233
Кнопка как пункт списка	196	Часть 3. Смешанная реальность	236
Сочетания кнопок	197	<i>Глава 21. Виртуальная реальность</i>	238
Западающие кнопки	198	Дополненная реальность	241
Плавающие кнопки	198	Смешанная реальность	245
Анатомия кнопки	200	<i>Глава 22. Виртуальный интерфейс</i>	246
Матрица компонента	200	Пространство и окружение	246
<i>Глава 16. Чекбокс, радиокнопка, тоггл, сегмент, список</i>	202	Виртуальные окна	248
Выпадающие списки	205	Взаимодействие с реальностью	252
<i>Глава 23. Кнопка в смешанной реальности</i>			260

До нашей эры

Наша эра



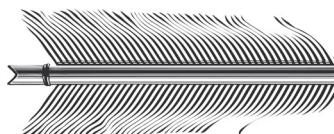
1911

Основание IBM



1976

Рождение Apple



Доисторический период

1973
Выход
Xerox Alto

1968
Первая
мышь

1985
Распад
империи

1993
Выход
Newton

Релиз
Windows 1.0



Чарльз Беббидж
1791...1871



Алан Тьюринг
1912...1954



Джеф Раскин
1943...2005



Наша эра



Часть первая

Настольные интерфейсы



Изобретение кнопки



Дверной звонок, XIX век

Про появление первой электрической кнопки мы знаем так же мало, как и про изобретение колеса, хотя кнопку придумали в 19 веке в Европе, а колесо изобрели примерно в - 3500 году в Месопотамии.

Рейчел Плотник в своей книге *Power Button* («Кнопка включения») пишет, что кнопки появились «не позднее 1880 года» и использовались для замыкания электрической цепи. И хотя люди к тому времени уже почти 50 лет пользовались азбукой Морзе, штуковину в его аппарате правильнее называть рычагом, а не кнопкой. Так что Плотник, наверное, права.

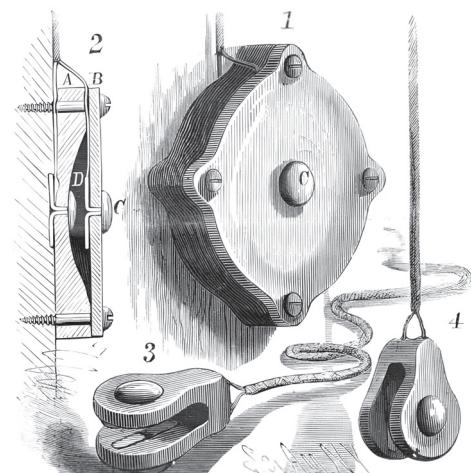
Первые кнопки делались в основном из бронзы и обрамлялись широкими панелями с огромным числом деталей — как и в любом предмете Викторианской эпохи. Чаще всего кнопки использовали для дверных звонков и вызова персонала в отелях. Сейчас такие звонки еще можно встретить в некоторых городах Европы или купить современную стилизацию на eBay.

Интересно, что поначалу кнопки, как и само электричество, вселяли настоящий ужас в современников. Люди 19 века боялись электричества куда сильнее, чем мы сегодня боимся «искусственного интеллекта». И их можно понять. В конце концов, автору ничего неизвестно про стул с ChatGPT, а вот электрический стул уже тогда использовали для казни.

Что касается самих кнопок, то, хотя они довольно быстро завоевали популярность, в конце 19 века даже образованные люди часто опасались, что, если «пустить дело на самотек», кнопки уничтожат способность нового поколения ориентироваться в мире и все так сильно привыкнут к кнопкам, что разучатся что-либо делать руками. И хотя в пользу нового изобретения мало кто сомневался, все-таки считалось, что человек сначала должен детально изучить принцип работы электрических цепей, затем внутреннее устройство самих кнопок — и только потом уже их нажимать.

SIMPLE PUSH BUTTON FOR ELECTRIC BELLS.

Mr. Gosonko, of Kozloff, Russia, sends us the following simple device for a key or push button for electric



GOSONKO'S PUSH BUTTON FOR ELECTRIC BELLS.

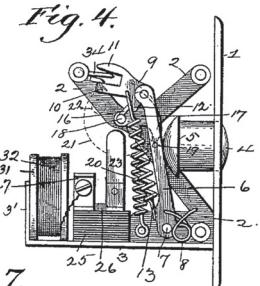
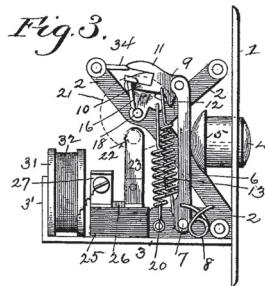
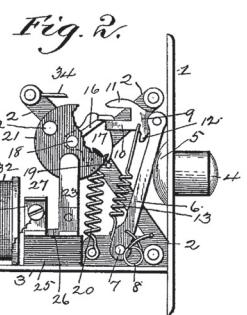
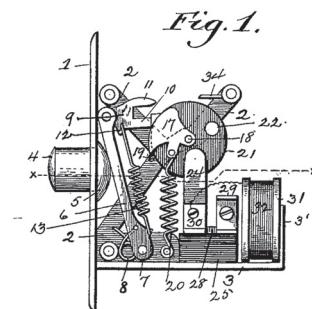


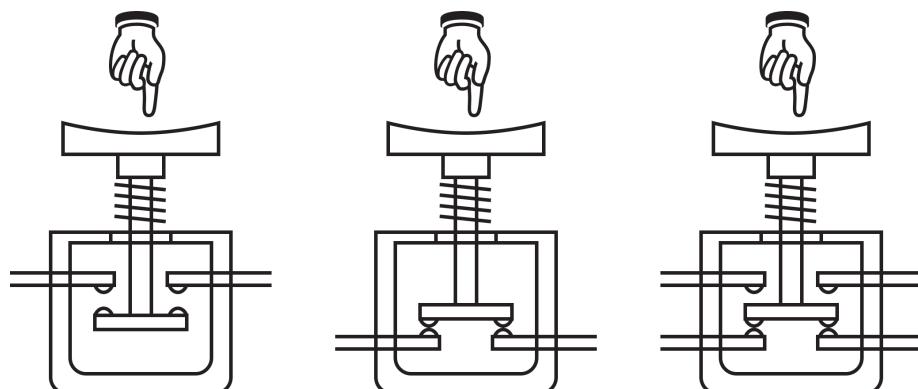
Fig. 7.

Давайте и мы внимательно изучим, как же работает кнопка.

Ее самая сложная часть таится вовсе не в нажатии, а в механизме возвращения в исходное состояние. Какие только конструкции не предлагались, чтобы «отжать» кнопку! Самый простой вариант — использовать пружину или посадить кнопку на упругую мембрану из резины, как предлагал в 1885 году некий мистер Госонко из города Козлова, приславший свое изобретение в журнал *Scientific American*. Были механизмы и гораздо сложнее, как, например, запатентованная в 1894 году американцем Маклафлином кнопка дверного звонка, состоявшая из тридцати трех деталей.

Слева: чертеж кнопки мистера Госонко из города Козлова, журнал *Scientific American*, № 26, 1885 год

Справа: чертеж кнопки американского инженера Маклафлина, патент № 521808 от 1894 года



Разные кнопки имеют разные механизмы, но суть их работы одинакова

Как бы там ни было, но все кнопки — что из трех, что из тридцати трех деталей, — по сути, работали одинаково. Они просто замыкали электрическую цепь, соединяя между собой два куска провода. По замкнутому проводу проходил ток, который активировал звонок. Все просто.

Удовлетворив страсть к дверным звонкам, к концу 19 века кнопки начали использовать для совершенно разных целей. Они стали массово внедряться во всю аппаратуру, которая существовала в то время. Так, компания Kodak в 1888 году запустила знаменитую рекламу новых кнопочных фотоаппаратов, слоган которой заявлял: «Вы нажимаете кнопку — мы делаем все остальное». Другая компания, Eveready, в 1899 году получила патент и выпустила первые кнопочные фонарики, работавшие от батареек собственного производства, которые сегодня выпускаются под брендом Energizer.

Реклама кнопочных фотокамер Kodak: «Вы нажимаете кнопку — мы делаем все остальное», 1888 год

New Kodak Cameras.

*“ You press the button,
we do the rest.”*

(OR YOU CAN DO IT YOURSELF.)

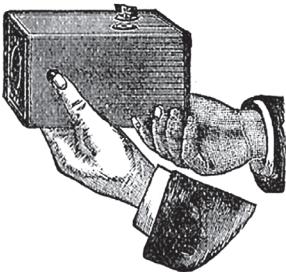
Seven New Styles and Sizes

ALL LOADED WITH

Transparent Films.

For sale by all Photo. Stock Dealers. Send for Catalogue.

THE EASTMAN COMPANY, ROCHESTER, N. Y.



Кнопки покоряли и более серьезные вещи. Например, в конце 19 века американские дома начали массово обзаводиться компактными выключателями света взамен громоздких настенных рычагов. Тогда же кнопки начали использовать в лифтах, которые до этого управлялись рычагами, а работали на гидравлике или паровой тяге. Чуть позже, уже в 1910-х годах, кнопками стали заводить автомобили. Спустя еще двадцать лет кнопки облюбовали радиоприемники. В 1950-х годах начался бум кнопочных приборных панелей на автомобилях, а уже в 1960-х кнопки захватили вообще всю электронику: телевизоры, телефоны, стиральные машины, плееры, игровые автоматы и даже самолеты.

Триумфальное шествие электрических кнопок по планете практически не встречало сопротивления вплоть до середины 1970-х. На пути кнопок к мировому господству оставался ровно один, последний бастион, который казался маленьким темным облачком на совершенно ясном небе: недавняя новинка, последний писк электрической моды — персональный компьютер. Однако в метре от пьедестала произошло непредвиденное. Кнопки столкнулись с обстоятельством непреодолимой силы, после встречи с которым начался их медленный, но верный закат.

Первого марта 1973 года исследовательский центр Xerox PARC представил первый в истории графический интерфейс.

Скевоморф атакует

С тех пор прошли уже десятки лет, и мы можем сказать точно: настоящих, физических кнопок за это время сильно поубавилось. Конечно, и компьютеры до сих пор используют больше сотни кнопок на клавиатуре, но все-таки это не сравнится с тем числом виртуальных кнопок, которые мы каждый день нажимаем на экране. Кнопкам не удалось захватить персональный компьютер. Вместо этого компьютер поглотил кнопки, забрав их души в виртуальный рай под названием графический интерфейс.

Интересно, что сегодня мы даже не замечаем подмены. Однако кнопки на экране — вовсе не то же самое, что кнопки в жизни. Дело в том, что виртуальная кнопка полностью лишена обратной связи. Физическую кнопку можно нашупать и нажать с закрытыми глазами, почувствовав при этом ее упругий ход или услышав щелчок. Виртуальную кнопку можно нажать только смотря в монитор, а ощутить ее нажатие нельзя.

При этом кнопки на экране изо всех сил пытаются вести себя как настоящие. У них бывают тени, подсветка, объем, а иногда даже звук — особенно если дизайнеру попался упрямый клиент. И хотя часто дизайнеры злоупотребляют оформлением, все эти спецэффекты появились не просто так.

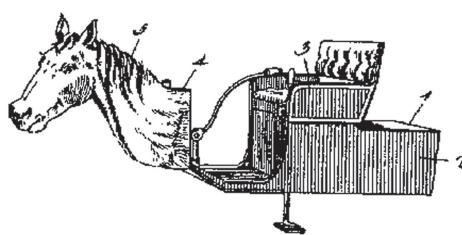
Еще в конце 19 века изобретатели поняли, что дивный новый мир кнопок и электричества слишком необычен для людей, привыкших ездить на лошадях и ужинать при свече. Лампы, электрические дверные звонки — все это пугало людей, потому что было для них незнакомым. Чтобы сделать новые устройства не такими пугающими, им пытались придать форму знакомых предметов. Например, электрические люстры выполнялись в виде пышных свечных канделябров, а лампочкам иногда придавали форму пламени.

Часто такая имитация действительно помогала человеку адаптироваться к новому миру, но иногда это доходило до абсурда. Самый известный пример — знаменитая Horsey Horseless (*англ. лошадь-безлошадь*). Один американский изобретатель в попытках сделать автомобиль менее пугающим для лошадей и жителей деревни додумался приделать к нему макет... лошадиной головы. Хорошо, что его изобретение не ушло дальше патента: вместо сияющих капотов автомобили Генри Форда вполне могли бы снабжаться громадной бронзовой конской головой.



Калькулятор в первых версиях iOS имитировал старый калькулятор Braun ET66 в рамках скевоморфизма

Fig.3.



Слева: чертеж патента № 30551 на автомобиль с лошадиной головой

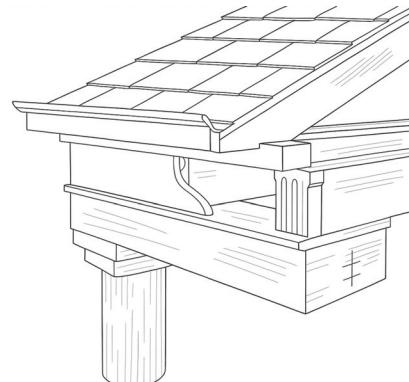
Справа: иллюстрация из журнала *Historical Motor Scrapbook*

Возможно, читатель уже догадался, о чем идет речь. Такое подражание дизайна новых изобретений старым вещам называется *скевоморфизмом*, и хотя это сложное слово известно далеко не всем, само явление известно каждому, кто хоть раз держал в руках телефон.

Обычно, когда говорят про скевоморфизм, имеют в виду дизайн первых версий систем OS X и iOS, в которых каждая деталь интерфейса стилизовалась под реальную вещь. Например, в приложении для записей блокнот состоял из нарисованных листов, прошитых декоративной пружиной, камера при съемке имитировала сужение диафрагмы, а на каждую кнопку был наложен десяток теней, градиент и текстура.

Это действительно хорошие примеры скевоморфизма, но на самом деле скевоморфизм — это не только интерфейс старого iPhone, а *вообще любое* подражание новой вещи старой с целью сохранения привычности. И хотя это слово появилось в конце 19 века, само явление с нами уже тысячи лет. Например, триглиф — узор из трех полосок, идущий вдоль ордера древнегреческих храмов, — не имеет никакого архитектурного смысла, а подражает более старым, деревянным храмам, где он получался просто потому, что так выпирали концы деревянных балок.

Триглиф в греческих храмах — элемент скевоморфизма, подражающий балкам в более старых деревянных храмах



Графический интерфейс, представленный Xerox PARC в 1973 году, стал грандиозной вехой в истории техники. Этот скачок был гораздо более невероятным, чем переход от деревянных храмов к каменным и чем переход от огненных свечей к электрическим лампочкам. Наверное, поэтому почти каждый элемент графического интерфейса является — или раньше был — скевоморфом, имитирующим какой-то объект реального мира.

Инженеры Xerox воплотили в жизнь нечто, что до этого существовало только в воображении и паре голливудских фильмов. Графический интерфейс настолько отличался от всего ранее виденного, что он просто должен был подражать реальному миру, чтобы его мог понять простой человек.

Книга, которую вы держите в руках, рассказывает о внутренней логике графического интерфейса, вскрывая самые глубокие механизмы его работы. Эта книга своего рода реверс-инжиниринг от мира дизайна. Шаг за шагом восстанавливая историю каждого элемента — кнопок, чекбоксов,

выпадающих списков, — мы поймем причины, по которым они выглядят и работают именно так, а не иначе. Подобно первым пользователям электрических кнопок, мы просто обязаны разобраться во внутренних механизмах кнопок виртуальных.

Автор уверен, что спроектировать действительно сложный, логичный и расширяемый интерфейс возможно, только досконально понимая его глубинную логику. А понять ее можно, только подняв со дна истории изначальные проблемы и их решения, которые сделали современные интерфейсы такими, какими они существуют сегодня.

Античные кнопки

Мы начнем с кнопок — самого простого элемента интерфейса.

Эпоха Античности для них началась в 1973 году с выходом Xerox Alto, первого персонального компьютера с графическим интерфейсом. Вместе с ним появилась и первая интерфейсная кнопка, ознаменовав конец первобытной эпохи кнопок физических. Выглядела она максимально просто: это была короткая надпись на английском языке, обведенная клетчатой рамочкой.

Очевидно, дизайнеры срисовали виртуальную кнопку с настоящей, физической кнопки, которые часто имели прямоугольную форму. А вот идея поместить надпись внутрь кнопки — это уже ноу-хай. В реальном мире надпись редко делают на самой кнопке, ведь простая краска быстро стирается от нажатия пальцем. Поэтому кнопки принято подписывать внизу или сбоку. Представляете, как бы выглядел интерфейс, если бы кнопки были круглыми, а подписывались сбоку? К счастью, такие интерфейсы встречаются только в компьютерных играх и симуляторах. Впрочем, чекбоксы и радиокнопки именно так и выглядят. Но об этом позже.



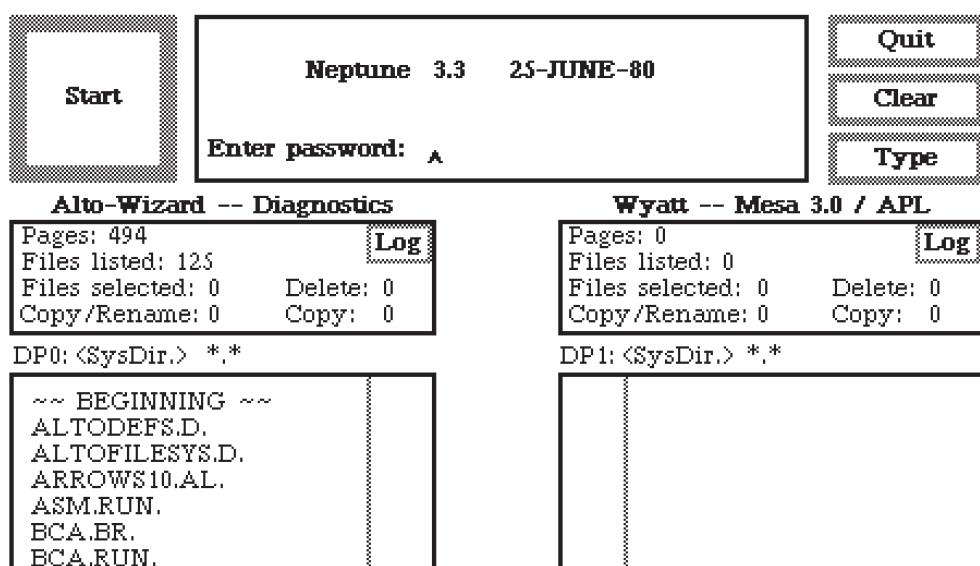
Кнопки в игровых интерфейсах и стилизациях могут быть круглыми, а подписываться сбоку

Слева: лифт во Франции
Справа: главное меню Fallout

Несмотря на свою примитивность, античные кнопки с самого появления следовали одному из главных правил дизайна: поскольку кнопка выполняет *действие*, надпись на ней должна быть в форме глагола.

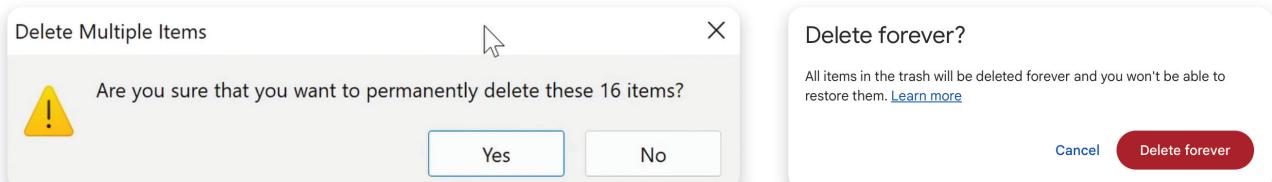
Взглянем на один из первых в истории файловых менеджеров, приложение Neptune для компьютера Xerox Alto. Это античный аналог проводника Windows, хотя своим двухколоночным интерфейсом он скорее напоминает Norton Commander. Работа с файлами нас сейчас не интересует; а вот что интересно — так это верхняя панель управления, на которой размещено несколько кнопок: Start, Quit, Clear и Type. Об их значении легко догадаться из названия, и даже если убрать обводку, по надписи-глаголу все равно будет понятно, что в это место можно нажать и ожидать выполнение действия.

Интерфейс файлового менеджера Neptune 3.3 для компьютеров Xerox Alto



Диалог очистки корзины в Windows 11 заставляет читать ненужный текст перед выбором кнопки. В аналогичном диалоге Google Drive выбор очевиден

Хотя это правило установилось еще в 1973 году, дизайнеры часто забывают о нем и сегодня, чем серьезно ухудшают интерфейс. Например, чтобы выбрать между кнопкам *Да* и *Нет* в диалоговом окне удаления файла, пользователь вынужден читать длинный и часто бесполезный текст. При этом выбор между кнопками *Удалить* и *Отменить* был бы очевиден без пояснений.

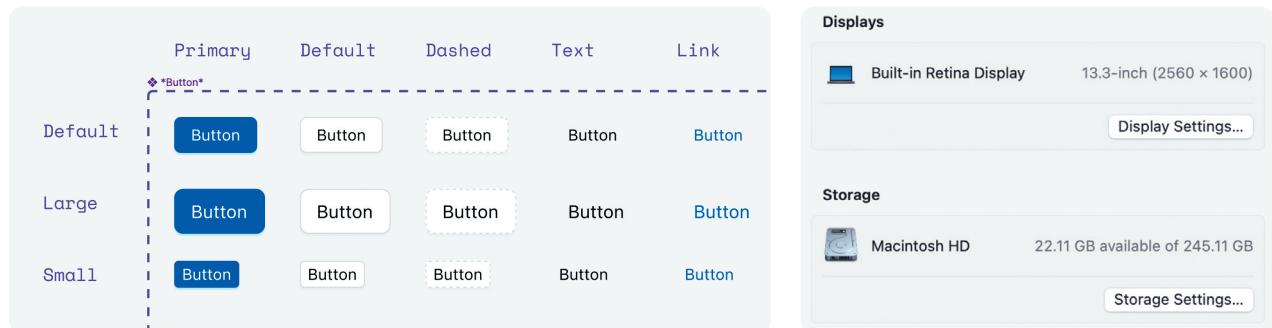


С другой стороны, дизайнеры сайтов иногда впадают в другую крайность и используют глаголы в гиперссылках, хотя гиперссылка может быть только существительным, поскольку просто ведет на другую страницу и не выполняет никаких действий.

Читатель может подумать, что такие ошибки совершают только новички, однако в Figma можно найти десятки профессиональных дизайн-систем, в которых ссылка оформляется не в виде отдельного компонента, а в качестве одного из стилей кнопки: например, под названием Link.

Слева: дизайн-система Ant Design — пример смешения кнопки и гиперссылки

Справа: если кнопка ведет на другой экран, то вместо глагола используется существительное с многоточием



Это говорит о системном непонимании незначительных, почти неуловимых логических отличий, которые лавинообразно приводят к грубым ошибкам в сложных интерфейсах. Ссылку и кнопку нельзя соединить в один компонент хотя бы потому, что у ссылки другой набор состояний: кроме Hover (при наведении) и Pressed (при нажатии) ссылка бывает Visited (посещенной), но такие «мелочи» сегодня принято игнорировать.

Sign in as IAM user

Account ID (12 digits) or account alias

IAM user name

Password

Remember this account

Sign in

[Sign in using root user email](#)

[Forgot password?](#)

IAM user sign in

Account ID (12 digits) or account alias

IAM username

Password

Show Password

[Having trouble?](#)

Sign in

[Sign in using root user email](#)

Remember this account

Старый и новый диалог авторизации Amazon AWS. Дизайнеры не могут определиться: делать Sign in using root user email ссылкой или кнопкой

Состояния кнопок

Кажется, я впервые употребил термин *состояние*.

У физических кнопок есть два положения: исходное и нажатое, в котором кнопка обычно находится доли секунды, вдавливаясь внутрь панели и затем отпружинивая в исходное положение. Виртуальные кнопки нельзя нажать пальцем, им некуда вдавливаться и у них нет нажатого положения. На самом деле мы и не нажимаем кнопки на экране. Мы нажимаем кнопку мыши, а нажатие кнопки на экране — всего лишь иллюзия.

При нажатии на кнопку мыши мы получаем от нее обратную связь в виде щелчка. Если при этом сфокусироваться на экранной кнопке и забыть про существование мыши, наше восприятие создаст иллюзию нажатия кнопки на экране. Этот же эффект позволяет человеку не задумываясь управлять велосипедом и машиной: мы не ощущаем руля и педалей, как будто срастаясь с механизмом в единое целое.

Идиллию нарушает лишь один недостаток: кнопка на экране все-таки не нажимается. И хотя человек получает тактильную и слуховую обратную связь при клике мышкой, эта связь не дополняется по зрительному каналу. Поэтому, чтобы иллюзия заработала на полную мощность, необходимо заставить кнопку визуально реагировать на нажатие.

Невероятно, но те самые античные дизайнеры, из ниоткуда явившие миру первый графический интерфейс, с самого начала позаботились об этой неуловимой детали. При нажатии на кнопку в Xerox Alto ее фон на долю секунды становился черным. Этот прием был не чем иным, как *имитацией тени*, которая накрывает физическую кнопку при нажатии пальцем.

Подобную имитацию нажатия неправильно называть положением, поэтому используется другой термин: *состояние* кнопки. Таким образом, мы можем говорить, что кнопка в интерфейсе Xerox Alto имела два состояния: исходное и нажатое, которые в современных дизайн-системах называются Default и Pressed.

Кнопка в Xerox Alto:
состояния Default
и Pressed



Черный цвет был выбран для имитации нажатия просто потому, что других вариантов не было: все дисплеи тогда были черно-белыми. Позже, когда дисплеи стали поддерживать градации серого, идею с тенями развили и начали имитировать освещение во всех деталях.

Эпоха Античности была в самом расцвете, когда 1 апреля 1976 года родилась компания Apple, ознаменовав начало нашей эры. Вопреки стереотипу, ранние компьютеры компании не умели работать с гра-

фикой и использовали текстовый интерфейс. Первым компьютером Apple с графическим интерфейсом стал Lisa, который появился только в 1983 году — всего за год до выпуска Macintosh, дальнейшего распада Империи и заката эпохи Античности.

Однако пришедшие затем Средние века не были такими темными, как иногда рассказывают историки. Хотя интерфейсы Средневековья выглядят устаревшими, они были хорошо проработаны для своего времени. Например, дизайнеры Mac OS и Windows научились придавать объем кнопкам двумя простыми приемами: нижней и правой границе они добавляли темную обводку, а верхней и левой — светлую. Таким образом дизайнеры хотели показать, что кнопка немного выступает на зрителя относительно плоскости, а из верхнего левого угла на нее падает свет. При нажатии на кнопку светлая обводка становилась темной, а на Mac OS — дополнительно затемнялся фон.

Эти примитивные тени на кнопках стали первым шагом к развитому скевоморфизму, который поначалу не удавалось создать из-за технических ограничений. В конце 1990-х годов началось бурное развитие компьютерной графики, которое продлилось все нулевые годы и потянуло за собой графический интерфейс.



Ранние Mac OS и Windows имитировали освещение из левого верхнего угла, накладывая на кнопки светлые и темные участки

Главные и вторичные кнопки

Эпоха Средневековья подарила нам множество изобретений, хотя ни одно из них нельзя назвать революционным. В период с 1984 по 2001 год компьютерная индустрия словно накапливала потенциал, чтобы реализовать его на полную мощность в эпоху Возрождения, которая началась с выходом десятой версии Mac OS.

Новая версия операционной системы от Apple представила интерфейс Aqua, основанный на технологии графики OpenGL. Привлекательный интерфейс стал главной фишкой новой системы. На ее презентации Стив Джобс сказал, что одной из задач было создать настолько вкусный дизайн, чтобы пользователи захотели лизнуть монитор. Новые кнопки были действительно хороши. Они напоминали капли воды, блестящие на солнце, — отсюда и название Aqua.

«Одной из задач дизайна было вызвать желание лизнуть его»

Стив Джобс на презентации OS X 5 января 2000 года

Был и еще один момент, который мало кто заметил. В той же презентации Джобс обратил внимание на новый вид кнопок. На одном из экранов справа от привычной серой кнопки *Сбросить* находилась синяя кнопка

Применить, которая плавно меняла цвет, как будто имитируя неоновую подсветку. Стив объяснил, что синяя кнопка — это основная кнопка, которую можно нажать при помощи клавиши Enter.

Это был первый случай разделения элементов интерфейса на несколько типов согласно их важности. Если до этого все кнопки были одинаковыми серыми прямоугольниками, то теперь одна из кнопок — основная — могла быть синей. Это позволяло мгновенно понять, какое действие является основным или рекомендованным.

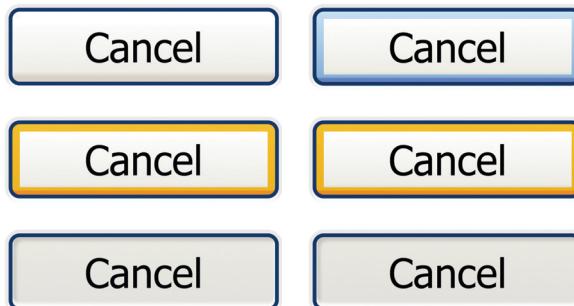
Разделение кнопок на два типа прочно вошло в дизайн интерфейсов. Сегодня сложно найти интерфейс, в котором не было бы основных и вторичных кнопок. Дизайн-системы обычно называют такие кнопки Primary и Secondary, а также могут содержать дополнительные типы, которые мы изучим позже.

Кнопки OS X 10.0
Слева: Secondary button
Справа: Primary button



Одновременно с Apple решила заняться своим интерфейсом компания Microsoft и в том же 2001 году выпустила Windows XP. В этой системе также появилось деление кнопок на основные и вторичные. Оба типа кнопок имели белый фон, но основная кнопка получала дополнительную синюю обводку. Кроме того, система привнесла в дизайн собственную инновацию, добавив кнопкам третье состояние. Теперь кнопка не только затемнялась при нажатии, но и подсвечивалась оранжевой обводкой при наведении мыши.

Кнопки Windows XP
Слева: Secondary buttons
Справа: Primary buttons
Сверху вниз: состояния
Default, Hover, Pressed



Интересно, что третье состояние, которое сегодня называется Hover, так и не появилось в Mac OS. Эта система до сих пор использует только два состояния кнопок: обычное и нажатое. Возможно, дизайнеры Apple правы: в конце концов, никакого «наведения» в физическом мире не существует, так зачем же добавлять его виртуальным кнопкам?

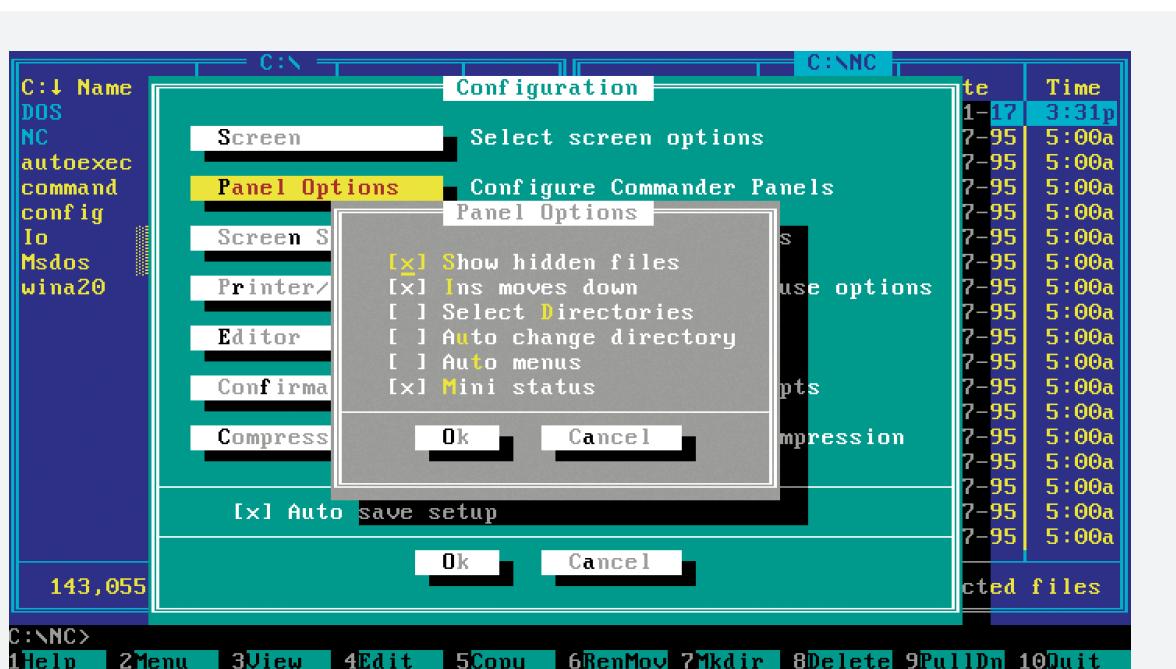
Тем не менее в большинстве современных интерфейсов используется состояние наведения, и автор не видит причин его не использовать. Легкое изменение цвета кнопки при наведении придает интерфейсу шарм, а мимолетное затемнение в момент нажатия создает ощущение глубокой проработки и отзывчивости интерфейса.

Этот эффект настолько важен, что может даже компенсировать недостатки логической части. Например, интерфейс Facebook*, являясь весьма неудобным, выглядит довольно привлекательным и дружелюбным. Это называется **эффектом «эстетика-юзабилити»**. Экспериментально показано, что из двух эквивалентных интерфейсов более удобным кажется тот, который выглядит более привлекательно.



Эффект «эстетика-юзабилити»

* Деятельность Meta Platforms, Inc. (в т.ч. по реализации соцсетей Facebook и Instagram) запрещена в Российской Федерации как экстремистская. — Здесь и далее прим. ред.



Текстовый реваншизм

Параллельно с прогрессом графического интерфейса развивался странный мутант — текстовый интерфейс. Он сочетал в себе элементы командной строки и полноценного интерфейса, рисуя плоские кнопки, галочки и даже выпадающие меню при помощи специальных символов и одноцветной заливки. Самый знаменитый пример такого интерфейса — файловый менеджер Norton Commander, завоевавший мировую известность в 1990-х годах. Им часто пользовались профессионалы, привыкшие вводить команды с клавиатуры и не считавшие графический интерфейс удобным.

Так что же такое состояние при наведении? В поисках аналогии в реальном мире автор вышел на две интересные метафоры.

Представьте, что вы ищете нужную кнопку на панели управления в комнате с плохим освещением. Вы используете маленький фонарик с узким лучом, который наводите на кнопки по очереди. Кнопка, на которую направлен фонарик, становится светлее. Возможно, именно такой эффект имитировали дизайнеры Windows XP.

Впрочем, некоторые дизайн-системы, наоборот, слегка затемняют кнопку при наведении. И такой подход тоже оправдан. Представьте, что вы поднесли палец к кнопке, которую хотите нажать. В таком случае на нее будет падать тень от пальца, которая сделает кнопку чуть темнее.

Какой подход выбрать — решать читателю. Автору всегда нравилась аналогия с фонариком, и я привык делать кнопки чуть светлее при наведении. Однако пока я писал эту книгу, мне пришлось пересмотреть свой подход. Он оказался логически неверным. Кнопки без заливки просто невозможно сделать светлее: они и так абсолютно белые.

Входит Фотошоп

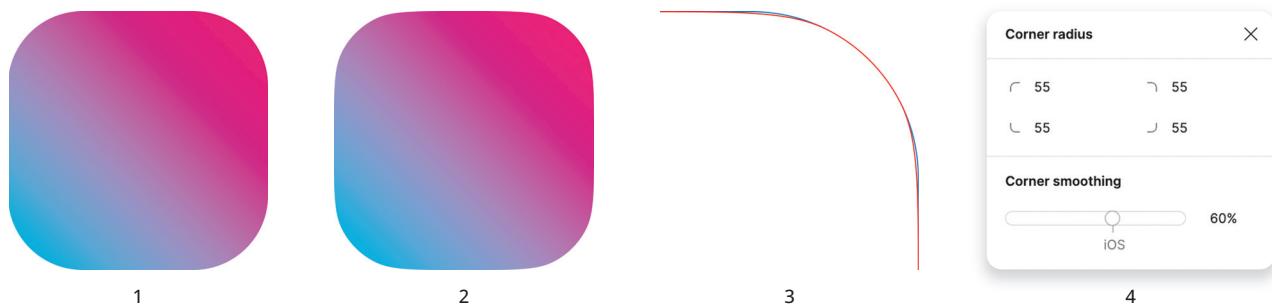
Скевоморфизм быстро развивался все нулевые годы. Дизайнеры тех времен соревновались, кто наложит больше теней, градиентов, текстур и анимаций на свои кнопки и при этом не вызовет у пользователя приступ эпилепсии. Типичная кнопка в то время включала: тень снизу, блеск сверху, свечение снизу, градиентную заливку, обводку, текстуру или шум, а также дополнительные эффекты для надписи.

Кнопка, стилизованная под натуральную кожу, от graphicsfuel.com — типичный представитель эпохи расцвета скевоморфизма, 2012 год



Верхом изящества считались контурные тени, которые были секретной фишкой Photoshop. Вместо обычной линейной тени можно было сделать тень, убывающую по кривой Безье. Контурная тень считалась таким же

признаком внимания к деталям, каким сегодня считается скругление углов при помощи суперэллипсов, принятное в iOS. И точно так же не поддерживалось ни одним браузером. Впрочем, это не было проблемой: в то время кнопки делали при помощи картинок.



Пик развития скевоморфизма пришелся на конец нулевых и начался в 2007 году с выпуска iPhone — первого телефона с управлением пальцами. Многие дизайнеры до сих пор с теплотой вспоминают интерфейсы первых версий системы iOS, в которых каждое приложение имело уникальный дизайн, проработанный до мелочей. Записки были выполнены в стиле блокнота, камера при съемке показывала анимацию диафрагмы, а книги в приложении-читалке стояли в шкафу на деревянных полках.

Квадрокруг со скруглением углов по суперэллипсу (2) едва отличим от скругленного квадрата (1) и является частью эстетики iOS. На рисунке 3 показано сравнение границ, рисунок 4 — настройка в Figma



Любопытно, что все это время дизайнеры рисовали интерфейсы в Photoshop — графическом редакторе для обработки фотографий, который был для этого совершенно непригоден.

Приложение календаря для iPad, работающего на iOS 6 — последней версии системы со скевоморфизмом

Первым нормальным инструментом для дизайна интерфейсов стало приложение Sketch, которое появилось в 2010 году и набрало некоторую популярность к концу декады, но так и не покорило сердца дизайнеров. Скорее всего, Sketch появился слишком рано. В то время скевоморфизм был еще популярен, но Sketch работал только с векторной графикой. Кроме того, программа поддерживала только Mac OS.

Поэтому дизайнеры, хотя и понимали преимущества Sketch, переходили на него медленно и неуверенно, в тайне запуская Photoshop и дорисовывая то текстуру, то пунктирную обводку. До выхода благословенной Figma оставалось еще несколько лет, а до того как она обзаведется всем нужным функционалом — почти десяток.

Спасение пришло откуда не ждали: скевоморфизм убила сама же Apple.

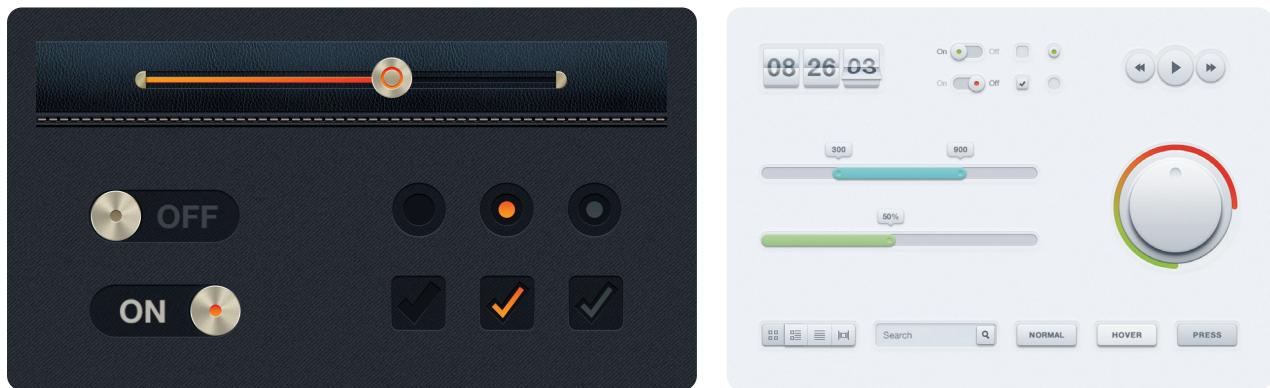
UI-киты и дизайн-системы

Ко временам скевоморфизма относится появление *UI-китов* — наборов элементов интерфейса, выполненных в одном стиле (*англ. kit — набор*).

Работали они следующим образом. Допустим, дизайнеру нужно было создать интерфейс какого-либо приложения, например читалки книг. После подготовки концепции будущего интерфейса следовало выбрать подходящий стиль и нарисовать интерфейс в полноценном виде: со всеми деталями, шрифтами, цветами, текстурами и прочим оформлением.

Эта часть всегда была самой сложной. Среди творческих людей не просто так существует понятие «страх белого листа». Выглядело это так: дизайнер открывал Photoshop, смотрел на белый лист и не мог понять, с чего начать работу. Ведь чтобы нарисовать даже простейшую кнопку, нужно знать хотя бы какого она будет цвета и каким шрифтом делать надпись. Но чтобы это понять, нужно сначала увидеть весь интерфейс целиком, а затем спуститься на уровень кнопки. Для решения этой проблемы и были придуманы UI-киты. Дизайнер шел на Dribbble или Envato и скачивал готовый набор кнопок, например в стиле дерева (ведь книги стоят на деревянных полках), после чего находил рядом подходящую текстуру, шрифт, дорисовывал недостающие элементы и таким образом постепенно собирал все экраны приложения.

Хотя UI-киты были полезным источником вдохновения, они слабо помогали в проектировании сложных интерфейсов, в которых помимо внешней стороны интерфейса (*UI, англ. user interface — пользовательский интерфейс*) требовалась глубокая проработка его внутренней логики (*UX, англ. user experience — пользовательский опыт*). UI-киты были слишком примитивны для больших задач. Обычно они включали пару простых кнопок, радиокнопку, галочку, переключатель и ползунок. Этого было достаточно, чтобы нарисовать приложение для книг, но явно не хватало для большого приложения вроде Microsoft Excel.



Была и другая проблема: скевоморфизм. Хотя многие кнопки действительно выглядели настолько вкусно, что хотелось лизнуть монитор, такая стилизация серьезно мешала развитию дизайна.

Интерфейс крупного проекта состоит из десятков компонентов, каждый из которых имеет несколько стилей и состояний. Взять хотя бы кнопку. Ее можно сделать с заливкой или с обводкой, а можно вообще выполнить в виде текста. Слева и справа от надписи можно добавить иконку, а также часто нужны кнопки разных цветов: белого, синего и красного (для удаления). И каждый из этих вариантов может иметь как минимум четыре состояния: обычное, при наведении, при нажатии и неактивное.

Даже простейшую кнопку крайне сложно нарисовать во всех этих стилях, видах и состояниях, работая в жанре скевоморфизма. Что уж говорить о десятках других компонентов! Именно поэтому в 2013 году Apple убила скевоморфизм. Принеся в жертву красоту, компания открыла дорогу *дизайн-системам*, которые стали следующим шагом в развитии дизайна.

Дизайн-система — это серьезно продвинутая версия UI-кита, которая включает десятки компонентов с множеством состояний, стилей и типов. Компоненты бывают элементарными: кнопка, чекбокс, поле ввода; или составными: календарь, форма, панель управления. При этом составные элементы строятся из элементарных: так, панель управления — это не более чем набор кнопок и выпадающих списков, объединенных в горизонтальный ряд. Такой принцип называется *атомарным дизайном*. Он был предложен в том же 2013 году дизайнером Брэдом Фростом. Наконец, хорошая дизайн-система кроме компонентов всегда предлагает библиотеку цветов (для дневной и ночной темы) и типографские стили для заголовков и надписей.

Эта книга посвящена поискам универсальной дизайн-системы. Проведя анализ десятков дизайн-систем и подробно изучив историю развития интерфейсов, автор заметил, что все они построены по универсальным принципам. С другой стороны, каждая дизайн-система так или иначе их нарушает. Изучению этих принципов и разбору ошибок дизайн-систем посвящены многие главы этой книги.

Типичные UI-киты были красивыми, но содержали крайне мало компонентов и не годились для сложных интерфейсов



Атомарный дизайн

Фотошоп уходит. Входит Фигма

Эпоха скевоморфизма подошла к концу в 2013 году, когда Джони Айв, директор по дизайну Apple, представил плоскую iOS 7, а через год и плоскую Mac OS Yosemite. Называть дизайн этих систем плоским стали за полное отсутствие теней, градиентов и текстур: Айв ликвидировал даже отдаленные намеки на скевоморфизм.

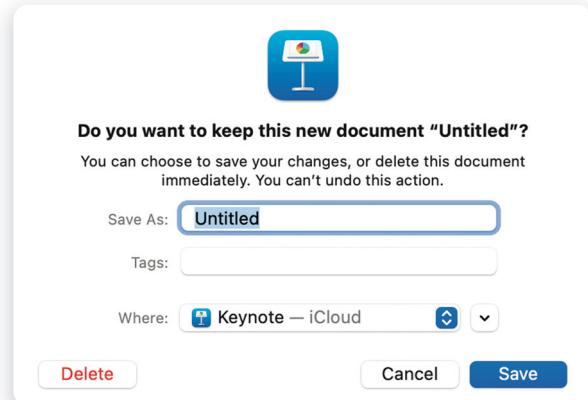
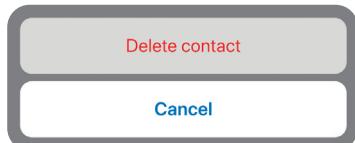
Пользователи восприняли новый интерфейс в штыки. Первые версии плоского дизайна выглядели очень невзрачно по сравнению с ярким, живым интерфейсом старой iOS. Однако со временем дизайн отполировали, и он стал весьма привлекательным. Сам Джони Айв объяснял убийство старого дизайна двумя причинами. Во-первых, он говорил, что скевоморфизм был полезен только в начале эпохи смартфонов, когда пользователи еще не привыкли к новинкам. Во-вторых, Айв считал, что лишние детали отвлекают внимание, ставя стиль приложений выше их функционала.

Несмотря на все ошибки и шквал критики, в целом мистер Айв оказался прав и весьма дальновиден. Немаловажно, что именно конец скевоморфизма дал дорогу великолепной программе Figma, пришедшей на замену Sketch, от которой сегодня без ума все дизайнеры. Но еще важнее, что плоский дизайн открыл дорогу дизайн-системам, которые до сих пор были практически неприменимы за пределами больших корпораций. Как следствие, в середине 2010-х годов в дизайне произошел свой «кембрийский взрыв», породивший целый веер новых кнопочек, чекбоксиков и выпадашечек.

Одной из первых новых кнопок, которая появилась в плоской Mac OS, стала *опасная кнопка* (англ. *Danger button*). На самом деле она появилась еще в первых версиях iOS, но до настольных интерфейсов дошла лишь спустя десятилетие. Опасная кнопка выделяется красным цветом и обычно выполняет опасное действие, например удаление файла.

Вслед за опасной кнопкой появились самые разные варианты этого компонента: с обводкой; без обводки, но с подсветкой при наведении; без обводки и без подсветки; а также в виде простого текста. Все эти стили мы изучим и включим в нашу дизайн-систему.

Опасная кнопка в интерфейсе iOS и Mac OS



Помимо стилистических изменений, большим прогрессом после отказа от скевоморфизма стало появление обвески — вспомогательных элементов, которые обычно размещаются у краев компонента. Современная кнопка состоит не только из подложки и надписи. Она также может иметь иконку слева или справа от надписи, а также дополнительную мини-надпись, которую обычно используют как счетчик.

Нельзя сказать, что ничего этого не было при скевоморфизме. Конечно, были и опасные кнопки, и кнопки с иконками. Однако создать целый набор кнопок с десятками комбинаций было просто невозможно. Во-первых, Photoshop не умел работать с библиотекой компонентов. Если нужно было поменять дизайн кнопки на десяти разных страницах сайта, нужно было открывать каждый файл отдельно и заменять слои вручную. Во-вторых, даже если решить эту проблему, поддержка дизайн-системы в жанре скевоморфизм с учетом всех теней, текстур и блесток была бы чрезвычайно трудоемка.

Sketch стал большим шагом вперед: он позволял кое-как работать с библиотеками и свойствами компонентов, однако настоящую революцию совершила Figma, нарастив количество полезных фишек до немыслимого уровня.

Анатомия кнопки

Пора перейти от теории к практике. Давайте возьмем в правую руку скальпель, а в левую руку — пинцет, и препарируем одну из кнопок, чтобы разобраться в ее анатомическом устройстве.

Через все тело кнопки, которое мы будем называть основой, проходит ее позвоночник — надпись. Обычно ширина кнопки подстраивается под длину надписи; сама же надпись не переносится на вторую строку и делается как можно короче. Слева от надписи находится иконка в виде пиктограммы, подобранной по смыслу под выполняемое действие. Справа от надписи также может быть иконка. Например, это может быть стрелка **>** вправо, которая часто встречается на кнопке «Далее». Наконец, справа от надписи иногда вклинивается дополнительная мини-надпись, которую автор привык называть *счетчиком*.





Как сделать кнопку в Figma. Свойства компонентов

Разумеется, не обязательно использовать все эти элементы. Смысл в том, чтобы создать универсальный компонент кнопки, в котором эти элементы предусмотрены в виде опций. В дизайн-системах эти опции оформляются в виде *свойств компонента* и могут включаться или выключаться дизайнером по необходимости.

Основы

Хотя кнопки бывают исключительно текстовыми, обычно кнопка имеет цветную подложку, которая помогает визуально отделить ее от фона. Такая подложка, называемая телом или основой, чаще всего представляет собой прямоугольник со скругленными углами. Однако в Figma основу никогда не делают при помощи инструмента «прямоугольник». Вместо этого используется фрейм в режиме Auto Layout, в который и помещаются все элементы кнопки, включая надпись и иконки.

Поскольку ширина кнопки должна автоматически адаптироваться под длину надписи, вместо указания ширины в пикселях ей присваивается свойство Hug contents (обхват содержимого). Что касается визуальной части, то два самых популярных стиля основы — Fill (с заливкой) и Outline (с обводкой).

Примеры простейших кнопок с заливкой и обводкой. Ширина кнопки автоматически подстраивается под длину надписи

Accept

Accept

Very long button label

Very long button label

Пожалуй, самый спорный параметр основы — это степень скругления углов. Система Mac OS вплоть до версии Snow Leopard использовала сильно скругленные кнопки, по форме напоминающие пилюлю или капсулу. С тех пор Apple отказалась от таких кнопок и перешла на прямоугольники со скругленными углами. При желании мы можем легко получить кнопку-пилюлю, скруглив углы до предела в Figma. Такая форма подойдет для стилизации, но в целом ее следует избегать: пилюльная форма сегодня ассоциируется с тегами и бейджиками.

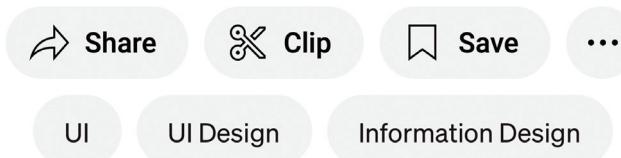
Кнопки с максимальным скруглением углов называются пилюлями. Сегодня такая форма чаще используется для тегов и бейджиков, чем для кнопок

Accept

Accept

Very long button label

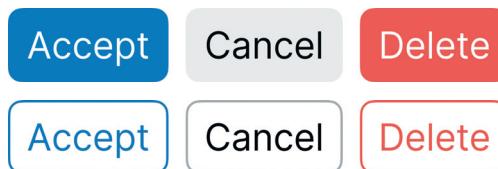
Very long button label



Кнопки-капсулы YouTube (сверху) выглядят как теги на Medium (снизу) и узнаются за счет иконок и размера

Цвет основы может быть каким угодно. Большинство интерфейсов обходится тремя цветами: синий используется для основных кнопок, серый — для вторичных, а красный — для опасных. В дизайн-системах цвет кнопки редко задается напрямую. Обычно он увязывается с типом кнопки и управляется свойством `Type`, которое принимает три значения: `Primary`, `Secondary` и `Danger`.

Обратите внимание, что меняется не только цвет основы, но и цвет текста. Очевидно, что на синем фоне читать черный текст тяжело, поэтому для главной и опасной кнопок с заливкой используется белая надпись.



За цвет кнопки отвечает свойство, определяющее ее тип: Primary-кнопки обычно синие, Secondary — серые, а кнопки типа Danger — красные

Двух стилей — с заливкой и обводкой — недостаточно для сложных интерфейсов, поэтому в дизайн-системах почти всегда есть еще два стиля.

Один из них называется `Ghost button`, что дословно переводится как «кнопка-привидение». В русском языке их часто называют пустыми, прозрачными или невидимыми кнопками. Конечно, такие кнопки вполне видимые, просто у них отсутствует как заливка, так и обводка; видны же только надпись и иконки. При этом такие кнопки подсвечиваются легкой заливкой как при наведении, так и при нажатии. В этот момент становится понятно, что кнопка по-прежнему имеет основу и внутренние отступы, просто ее цвет сделан прозрачным.

	Ghost	Text	
Default	Accept	Accept	We will soon take action because Program policies. Fix violations as Go to Policy status
Hover	Accept	Accept	We will soon take action because Program policies. Fix violations as Go to Policy status
Pressed	Accept	Accept	We will soon take action because Program policies. Fix violations as Go to Policy status
Disabled	Accept	Accept	We will soon take action because Program policies. Fix violations as Go to Policy status

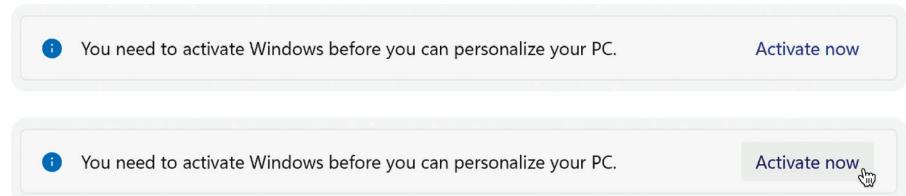
Слева: Невидимые кнопки имеют основу, поэтому подсвечиваются при наведении и нажатии. Текстовые кнопки не имеют основы и внутренних отступов

Справа: Пример правильного выравнивания невидимой кнопки в Google Play Console. Кнопка смещается влево, чтобы границы надписи и текста совпадали

Второй стиль называется Text button. Это текстовая кнопка. В отличие от невидимой кнопки, текстовая кнопка при наведении и нажатии не подсвечивается заливкой. Меняется только цвет текста, что роднит ее с гиперссылкой. Именно этот стиль дизайнеры ошибочно называют Link. Текстовая кнопка не имеет основы и внутренних отступов.

Невидимые кнопки можно часто найти в продуктах Google и других крупных корпораций. Они часто используются в панелях навигации и инструментов, а также в качестве вторичной кнопки на экранах авторизации.

Невидимые кнопки имеют основу, внутренние отступы, которые становятся видимыми за счет подсветки при наведении



Текстовые кнопки используются в тех же целях. Зачастую вторичная кнопка настолько маловажная, что хочется уделить ей как можно меньше внимания. В этом случае текстовая кнопка подойдет как нельзя лучше. Кроме того, у нее нет внутренних отступов, что здорово экономит место.

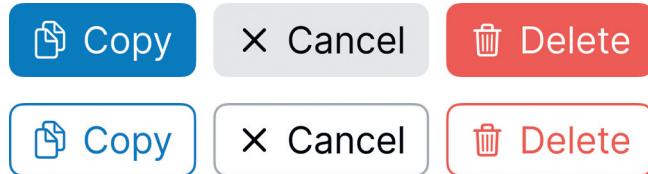
Текстовые кнопки не имеют основы, поэтому они особенно компактны и не отвлекают внимание



Левая иконка

Слева от надписи на кнопке часто размещают иконку. Ее задача скорее декоративная. Иконка дополняет надпись и привлекает внимание, поэтому не следует размещать ее на каждой кнопке. Обычно иконку добавляют на ту кнопку, которую хотят дополнительно выделить. Если несколько кнопок идут в ряд, то иконки часто добавляют на каждой из них, чтобы ряд таких кнопок не выглядел как строка текста.

Иконка слева от надписи хорошо выделяет кнопку среди других



Теперь, когда мы добавили на кнопку иконку, никто не запрещает удалить из нее надпись и получить кнопку-иконку. Такую кнопку нужно использовать с осторожностью. Иконка должна быть понятна пользователю без надписи и подсказки, чего не так-то просто добиться.

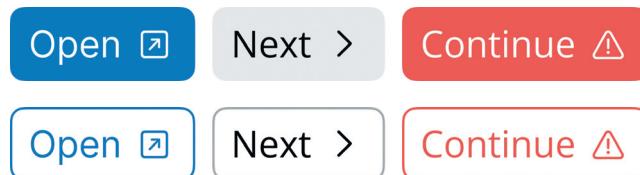
Список однозначно понятных иконок очень короткий и включает не больше десятка пиктограмм. Так, корзина  однозначно понимается как удаление, а шестеренка , скорее всего, откроет настройки. В то же время такой примитивный знак как крестик  может означать что угодно: закрыть, удалить, скрыть или отменить. То же самое касается и двух файлов  друг на друге. Без контекста невозможно понять, копирует ли эта иконка выбранный файл в буфер обмена или создает его дубликат.



Если теперь максимально скруглить углы квадратной кнопки-иконки, как мы ранее делали с прямоугольной кнопкой, то получится вполне симпатичная круглая кнопка. В отличие от кнопок-капсул, которые легко перепутать с тегом или бейджиком, такая круглая кнопка-иконка — неплохой прием, который, например, часто использует компания Google.

Правая иконка

Иконка может быть не только слева от надписи, но и справа. Чаще всего используется стрелка  вправо, которая хорошо подходит к серой кнопке с надписью Next. Для разнообразия подберем к ним другую иконку, например  стрелку открытия в новом окне. Аналогичных примеров для красной кнопки автор не встречал, однако можно себе представить такую кнопку с надписью Continue и иконкой  восклицательного знака.



Круглые кнопки-иконки Amazon Prime Video и Facebook (деятельность Meta Platforms, Inc. (в т.ч. по реализации соцсетей Facebook и Instagram) запрещена в Российской Федерации как экстремистская)

Кнопки-иконки без подписи стоит использовать с осторожностью, ведь даже простейшие пиктограммы могут пониматься по-разному

Круглые кнопки-иконки могут быть частью фирменного стиля компании

Иконка справа уточняет надпись на кнопке

Разница между левой и правой иконкой довольно тонкая. Иконка слева *подтверждает* надпись. Например, если на кнопке написано «Скачать», то иконка  облака располагается слева, потому что она изображает само действие скачивания, выступая в роли иллюстрации. Справа же от надписи размещается иконка, которая *уточняет* надпись. Это хорошо видно на примере кнопки Open: стрелка в квадрате не иллюстрирует действие открытия, а уточняет, что ссылка откроется в новом окне.

Кнопки с иконкой справа:
Trello, Google Material Design, Google Images

 Watching ✓

Start exploring Material 3 →

Visit >

Кнопки в GitHub для создания ветки и добавления в избранное показывают, сколько людей это уже сделали

Счетчики на кнопках помогают понять, над сколькими объектами будет совершено действие

Счетчик

Заключительный элемент кнопки — счетчик. Это маленькая вторичная надпись, которая размещается между надписью и правой иконкой. Автор называет эту надпись счетчиком, поскольку есть множество примеров кнопок, которые что-то считают. Например, на кнопке «Скачать файлы» можно указать, сколько всего файлов будет скачано; кнопка «Удалить» со счетчиком поможет понять, сколько файлов будет удалено; и так далее.

 Download 3  See results 15  Delete files 2

 Download 3  See results 15  Delete files 2

Конечно, счетчик можно использовать и в других целях. В следующих главах мы увидим, что мини-надпись можно использовать не только для счетчика.

Состояния кнопки

В ходе исторического обзора мы выяснили, что кнопки реагируют на действия пользователя: подсвечиваются при наведении курсора и затемняются при нажатии, имитируя физический мир. Чтобы учесть реакцию компонента на подобные события, каждый компонент дизайн-системы создается в нескольких состояниях. В Figma для этого используются Variants — *варианты компонента*.

Состояния компонента оформляются в виде свойства, обычно под названием State, которое можно переключать так же, как переключаются



Варианты компонентов в Figma

типы и стили. Исходное состояние компонента называется Default, а состояние при наведении — Hover. Состояние в момент нажатия дизайнеры часто называют Active, но это неверно: кнопка не «активируется», а на доли секунды *нажимается*, поэтому автор предпочитает называть его Pressed.

Четвертое состояние называется Disabled и используется для неактивных кнопок. На самом деле это не совсем состояние. Мы называем состоянием реакцию кнопки на действия *пользователя*, но неактивными кнопки становятся по воле программиста.

Споры насчет пользы неактивного состояния ведутся с его появления в 1980-х годах. Кажется, нет никакого смысла запрещать нажатие кнопки: если она не нужна, почему бы просто ее не скрыть? Дело в том, что кнопка может быть в целом нужна, но в ряде случаев нажатие на нее было бы нежелательным. Например, кнопка «Удалить» может быть доступна только для тех файлов, которые пользователь создал сам, но не для системных файлов. Если в таком случае не показывать кнопку совсем, то новый пользователь, покликав пару файлов и не увидев кнопки удаления, может решить, что файлы вообще нельзя удалять. Чтобы избежать подобной ситуации, кнопки принято все-таки показывать, но делать полупрозрачными и запрещать нажатие.

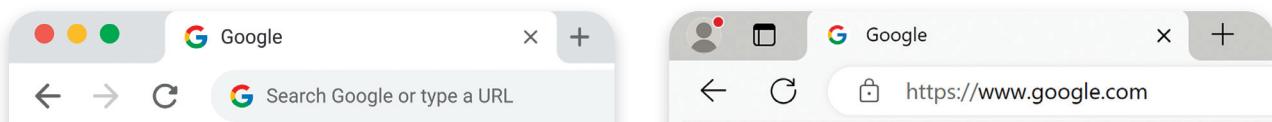
Есть и третий подход. Его сторонники говорят, что лучше всегда показывать и разрешать нажимать кнопку, но при нажатии отображать ошибку. В пример обычно приводится какая-нибудь форма, в которой кнопка «Отправить» становится доступной только после заполнения всех полей. В этом случае действительно лучше позволить нажать кнопку, даже если форма не заполнена, а затем подсветить пустые поля красным цветом. Однако такой подход вряд ли подойдет, например, для кнопки → «Вперед» в браузере, так что этот способ не универсален.

Выбор между двумя основными вариантами — запрещать или скрывать ненужную кнопку — сводится к следующему. Если действие, выполняемое кнопкой, недоступно временно или в конкретной ситуации, то кнопку следует запрещать. Если действие недоступно в принципе, то есть из-за постоянных ограничений, то кнопку не следует показывать вообще. Например, если пользователь включил режим «Не беспокоить», то это временная ситуация, поэтому кнопку звонка напротив его имени нужно показать в неактивном состоянии. Если же звонки запрещены администратором системы, то это постоянное ограничение и кнопку показывать не нужно.



Вопросы UX: скрывать или запрещать?

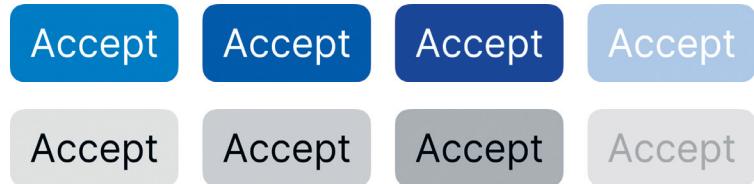
Даже кнопку → в браузере можно как запретить (Google Chrome), так и просто скрыть (Microsoft Edge)



Итак, мы имеем четыре состояния кнопки: исходное, при наведении, нажатое и неактивное. Как мы договорились, при наведении мы будем

делать цвет кнопки чуть темнее, а при нажатии — затемнять сильно. Для неактивной кнопки мы будем использовать полупрозрачный фон и текст.

Состояния кнопок, слева направо: исходное, при наведении, нажатое и неактивное



Режим фокуса

Почти всегда дизайнеры ошибочно включают в дизайн-систему пятое состояние под названием Focus. Оно используется для кнопки, которая находится в так называемом «фокусе». Такая кнопка снабжается дополнительной обводкой, которая в системе Mac OS имеет вид толстой синей линии, а в Windows — тонкой пунктирной линии. Это состояние показывает, что данную кнопку можно нажать с клавиатуры.

Скажем прямо: это не самая важная вещь в современном дизайне. Нажатие кнопок с клавиатуры было популярным в 1990-х годах, когда в Windows 98 постоянно что-то ломалось и беспроводная мышь могла перестать работать после обновления драйверов. В таком случае приходилось откликаться на предыдущую версию драйвера, пробираясь в настройки системы при помощи клавиатуры.

Сегодня такие случаи чрезвычайно редки. Например, в системе Mac OS навигация с клавиатуры по кнопкам выключена и работает только для текстовых полей. Но даже если ее включить, она будет вести себя странно: кнопка в фокусе нажимается только пробелом, а при нажатии Enter всегда нажимается синяя Primary-кнопка (даже если она не в фокусе).

Автору кажется, что дизайнеры давно «забили» на это состояние и сделали это не без оснований. Во-первых, навигацией с клавиатуры действительно почти никто не пользуется. Если только вы не проектируете сугубо системный, безотказный интерфейс, то можно вообще не рисовать это состояние для компонентов, за исключением текстовых полей. Что касается людей с ограниченной подвижностью рук, то большинство из них используют сенсорный экран или специальные плагины. Во-вторых, если навигация с клавиатуры включена, операционная система сама добавит обводку кнопке в фокусе — если, конечно, она оформлена как элемент button.

Есть и другая проблема. На самом деле фокус — это не состояние, а режим. Разница в том, что состояния взаимоисключают друг друга, то есть кнопка может принимать только одно из состояний в один момент времени и не может быть в двух состояниях сразу. Но кнопка в любом состоянии — исходном, при наведении и при нажатии — может при всем этом быть еще и в фокусе!



Операционные системы сегодня не особо заботятся о навигации с клавиатуры. В данном примере нажатие Enter приведет к нажатию кнопки Save, хотя в фокусе находится кнопка Don't Save

Accept >

Accept >

Accept >

Кнопка в фокусе может принимать любое состояние, поэтому фокус — это не состояние, а режим

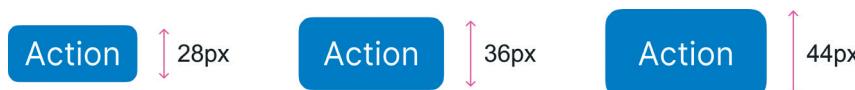
Это один из примеров логической путаницы, которая укоренилась в головах многих дизайнеров, включая профессионалов. Во всех известных автору дизайн-системах Focus оформляется как состояние наряда с Hover и Pressed, хотя правильнее было бы сделать его отдельным свойством.

В связи со всем вышесказанным, мы добавим режим фокуса в нашу дизайн-систему только в целях совместимости и не будем уделять ему много времени. Он будет оформлен при помощи слоя с синей обводкой и включаться отдельным свойством-переключателем.

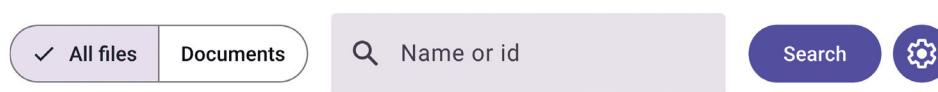
Размеры

Многие дизайн-системы, в том числе Windows и Mac OS, используют всего один размер компонентов. Другие системы, такие как Bootstrap и Ant Design, используют три размера: маленький, средний и большой. Учитывая, что ширина компонентов обычно подстраивается под их содержимое, говоря о размерах мы будем подразумевать только высоту. В отличие от ширины, высота компонентов задается фиксированно, при помощи значения Fixed.

Автор не нашел никакой закономерности в определении размера кнопок, которую можно было бы взять за правило. Каждая дизайн-система использует свои размеры, и все они случайны. Личный вкус и опыт автора подсказывают, что оптимальные размеры должны быть 28, 36 и 44 пикселя.



Читатель может выбрать другие размеры. Главное, чтобы высота всех компонентов совпадала, ведь они могут размещаться в линию. Это правило кажется очевидным, но оно часто нарушается. Например, в Material Design высота текстового поля составляет целых 56 пикселей, хотя высота кнопки и других компонентов — всего 40. Из-за этого текстовое поле выдается из стройного ряда кнопок на панели управления.



Текстовое поле в Material Design выделяется из ряда других компонентов из-за чрезмерно большой высоты

Создавая нашу дизайн-систему, мы постараемся не допускать таких грубых ошибок. Мы также будем применять визуальную компенсацию: большие кнопки требуют больших внутренних отступов и более сильного скругления углов, чтобы выглядеть сбалансированно.

Матрица компонента. Сводим все воедино

Итак, мы обсудили все параметры кнопки. Теперь наша задача — свести их воедино и оформить кнопку в виде компонента с набором свойств.

Важно понять следующее. Весь этот набор типов, стилей, размеров и состояний порождает огромное число комбинаций. Если быть точным, то 144 варианта одной и той же кнопки. Такое количество вариантов невозможно представить в виде списка, поэтому в дизайн-системах компоненты хранятся в виде таблиц. По вертикали откладываются стили и состояния, а по горизонтали — размеры и типы.

Для начала опробуем этот подход на одном размере и одном стиле. Выпишем состояния кнопок в столбик, а типы — в строку. Получим таблицу 3×4 , в которой представлено 12 кнопок.

Комбинации параметров компонента формируют таблицу, в которой столбцы соответствуют типам, стилям и видам, а строки — состояниям

	Primary	Secondary	Danger
Default	Accept >	Cancel >	Delete >
Hover	Accept >	Cancel >	Delete >
Pressed	Accept >	Cancel >	Delete >
Disabled	Accept >	Cancel >	Delete >

Автор называет полученную таблицу *матрицей видов и состояний*, или просто *матрицей компонента*.

Выше приведен самый простой случай матрицы для одного типоразмера кнопки. Чтобы получить полную матрицу всех вариантов, размножим таблицу так, чтобы по горизонтали получилось три, а по вертикали — четыре таких таблицы. Для удобства сгруппируем кнопки по цвету. По горизонтали подпишем группы согласно размерам, а по вертикали — согласно стилям кнопок. Получится огромная матрица компонента, состоящая из 144 вариантов кнопки.

Small		Medium		Large	
Primary	Secondary	Primary	Secondary	Primary	Secondary
Fill					
Default	Action 3 >				
Hover	Action 3 >				
Pressed	Action 3 >				
Disabled	Action 3 >				
Outline					
Default	Action 3 >				
Hover	Action 3 >				
Pressed	Action 3 >				
Disabled	Action 3 >				
Ghost					
Default	Action 3 >				
Hover	Action 3 >				
Pressed	Action 3 >				
Disabled	Action 3 >				
Text					
Default	Action 3 >				
Hover	Action 3 >				
Pressed	Action 3 >				
Disabled	Action 3 >				