

マイクロインタラクション

[デザイン]

デザイン

■マイクロインタラクション

単一のシナリオに基づいて一つの作業だけをこなす最小単位

- ・細部こそが製品を作り上げる
- ・製品購入の理由にはならない
- ・ちょっとした特徴や補助機能
- ・さまざまなユーザーエクスペリエンスを生み出す
- ・信じられないくらい重要
 - ・こよなく愛する製品
 - ・許容範囲の製品

最適な用途

- ・単一タスクの処理
- ・機器同士の接続
- ・単一データの取得
- ・現在進行中プロセス制御
- ・特定設定の変更
- ・小さなコンテンツの表示や生成
- ・特定機能のオン・オフ

製品の一部であることも、製品全体であることもある

■フィーチャー

特徴的機能だがマイクロインタラクションとは呼べない

マクロインタラクション

- ・優れたデザインが伴う
 - ・UX デザインのツボ

■構造

4つのパート

- ・トリガー
 - ・開始させる
 - ・ユーザーが何をいつ、どれくらいの頻度でやりたいかを理解

シグニファイア

. アフォーダンス

. 知覚可能なデザイン上の手がかかり
アクセシビリティ
継続時間

- ・コントロール

手動トリガーで最も重要

- ・システムトリガー

条件充足検知でシステムが起動

- ・ルール
 - ・動作を定義
 - ・トリガーが起動する条件
 - ・ユーザーはルールを知りたい
- ・フィードバック
 - ・ルールをユーザーに明示
 - ・ユーザー理解を助けるメタファ

ファイルをフォルダに入れる
メールが郵便受けに届く

- ・進行中のインタラクション

視覚的、聴覚的、触覚的要素

- ・ループとモード
 - ・メタルール

■ 3つの手法

マイクロインタラクションを個別に検討

- ・マイクロインタラクションのスコープを限定
 - ・難しい

フィーチャーにしたがる人が多い
マイクロインタラクションは、抑制の行為
可能な限り小規模、多くこなせる
Less is More(省略の美)
. ミース・ファン・デル・ローエ

ひとつのマイクロインタラクションを中心にしてひとつの製品を作り上げる

- ・無駄をそぎおとし、本質、真髄だけ残す
- ・究極の簡素な製品が正当化、促進

多数のマイクロインタラクションをひとつの製品にまとめあげる

- ・アプリや機器
 - ・数多くのマイクロインタラクションの協調動作の結果
 - ・細部は単なる細部にとどまらない。細部こそが製品を上げる

チャールズ・イームズ

- ・小規模、アジャイルな手法と相性がいい
- ・弱点
 - ・全体を見失う恐れ
 - ・時間も労力もかかる

それでもそれだけの価値はある

トリガー

■手動トリガー

原則

- ・原則 1: コンテキスト (周囲の状況・文脈) から

トリガーであると認識できるものをトリガーとする

- ・もたせる

いかにも見えそうな外見
何かしてくれそうな雰囲気

- ・起動

ユーザーの要求

- . いつ、どのような状況かを理解
- .. いつ、どこに置くかが決まる
- .. デザインに関するリサーチ
- ... インターフェースデザインの実践教室
- .. 直観や深い理解
- .. 製品テストや出荷後

- ・ユーザーが自ら起動
- ・原則 2: 常に同じことが始まるようにする
- ・不可視トリガー

メニューに埋め込まれ見えない

- ・原則 3: データを前面に出す
 - ・ マイクロインタラクションに含まれるデータをトリガーに反映させ提示も可能
 - ・ 表示できるデータのうち最も価値があるもの

マイクロインタラクションが何のために使われるか

- ・原則 4: 視覚的なシグニファイアを破壊しないこと
 - ・ 外見がボタンであれば、ボタンのように振る舞うべき
- ・原則 5: マイクロインタラクションの使用頻度が

高ければ高いほど、見つけやすいものでなければならない

- ・利用

ある程度のユーザーがかなりの頻度で利用

. 簡単に見つけられなければならない

ほとんどのユーザーが頻繁に利用

. すぐ見つけられなければならない

あまり利用されない

. 探さなければ見つからないようにする

- ・ どうやって見つけるか

ものが動いたり音をたてる

目的をもって探す

. 視覚優位探索

.. 音に対する反応のほうが速い

.. 何かを探すときは視野が狭まる

... 通常の 1% 未満しか見えない

. 対象認識

.. ジオン

... 馴染みのある形状

.... 四角形、三角形、立方体といった単純な図形や立体

... 目はジオンを探す

... 脳はジオンを組み合わせてその物体が何かを把握

... トリガー、特にアイコンのようなものには、幾何図形を使うとよい

... 単一の特徴に基づいて探す方が目的物を見つけやすい

.... 視覚的に単純な図形とするのが最善

- ・原則 6: 偽のシグニファイアが生成されないようにする
 - ・ 認知負荷を小さく

トリガーの使い方を考えさせない

発明は最後の手段

. 可能な限り標準コントロールを使用

- ・原則 7: トリガー自体では表せない情報を提供するときに限ってラベルを付加する
- ・実用本位

曖昧さ解消

短い言葉

しっかりとした説明

首尾一貫

利用者の言葉

構成要素

- ・コントロール
 - ・少なくとも一つ必要
 - ・カスタムコントロール

存在すると製品のマイクロインタラクションが際立つ
コントロール自体が記憶に残る瞬間

- ・適切なデフォルト値

マイクロインタラクションの目的は選択の負荷を減らすこと

- ・コントロールの状態
- ・文字あるいはアイコン化されたラベル

不可視トリガー

- ・要素を隠すことにより機能を削除しなくても見かけがすっきりする
 - ・見えなくすることは、マイクロインタラクションの目標ではない
 - ・学習可能でなければならない
 - ・優先度の高いマイクロインタラクションに割り当ててはならない

コントロールの状態

- ・デフォルト
 - ・待ち
- ・ホバー
 - ・ツールチップなどでクリック可能を示す
 - ・内部データを一部表示出来ると効果的
- ・ロールオーバー
 - ・画像入れ換えなどでコントロールを変化
- ・起動中
- ・アクティブ
 - ・処理が進行中

バックグラウンド処理があれば、内容を示すのが効果的

- ・トグル

- ・現在の設定を表示

ラベル

- ・トリガーによっては重要
- ・名前付け
 - ・何が行われるのか
 - ・本当にやりたいことなのか
- ・視覚的負担が増す
 - ・曖昧な場合に付ける

■システムトリガー

実はユーザー起動より多い

起動条件要因

- ・エラー
- ・位置情報
- ・データの到着
- ・内部データ
- ・その他のマイクロインタラクション
- ・他人

ルール

- ・起動頻度
- ・既知のユーザー情報
 - ・現在夜中ならシステムトリガー頻度をさげる等
- ・トリガー起動を見えるようにする必要性
- ・システムエラーが起きるとどうなるか

ルール

■ルールのデザイン

マイクロインタラクションの要となる

目的

- ・最も重要な要素
- ・マイクロインタラクションの目的を単純明快に定義すること
- ・理解可能かつ達成可能
- ・最終的な状態を定義
- ・目的に絞られているほどよいものとなる
- ・ルールの原動力

ルールは、ユーザーの行動を制約

- ・ルールと感じさせないことが大切

- ・自然
- ・うまくいっているという感覚

決定する必要

- ・起動されたときどう応えるか
- ・進行中にどう制御出来るか
- ・操作の順番とそのタイミング
- ・どこから取得するどのようなデータが使われるか
- ・使われるアルゴリズムとパラメータ
- ・いつどのようなフィードバックが返されるか
- ・どのモードにあるか
- ・繰り返されるのか、頻度はどれくらいか
- ・終わると何が起こるか

ルールを作る

- ・近道
 - ・大まかなルールをすべて書き出す
 - ・ルールを目に見える形で書き出す

フローチャート

動詞と名詞

- ・新たな視点
 - ・マイクロインタラクション全体を文ととらえる

動詞は操作

名詞は操作対象

画面と状態

- ・画面をすこしずつ変化
 - ・これまでのウィザード形式よりよい
- ・操作対象の状態
 - ・待ち(デフォルト)
 - ・アクティブ
 - ・更新された

制約

- ・主なもの
 - ・使える入出力
 - ・入力の型と範囲
 - ・リソース負荷
 - ・使えるデータ
 - ・集められるデータ

ゼロから始めない

- ・まず自問する
 - ・ユーザーとバックグラウンドについて何を知っているのか
- ・持っている情報を使えば改善できる
 - ・例

プラットフォームやデバイス

時刻

室内の物音

最終利用からの経過時間

会議中か

一人か

バッテリー寿命

現在地、移動方向

過去の行動

他人のデータも利用できる

複雑さを軽減

- ・テスラーの複雑さ保存の法則
 - ・すべての作業には固有の複雑さがある
 - ・これ以上単純化できない限界がある
- ・固有の複雑さをどう扱うか
 - ・システムが処理

ユーザーから権限を奪う

マイクロインタラクションに関してはこちらがよい

- ・ユーザーが処理

ユーザーにより多くの決定

この場合でもシステムが手助けできないか知恵を絞る

- ・検討
 - ・中心の複雑さを見つける
 - ・全体の処理過程を考慮
 - ・ユーザーがいつ、どの部分に関与したいか検討
 - ・ユーザーによる制御がどうしても必要なら与える

選択肢の制限と賢いデフォルト

- ・ルールが少ないほど優れたマイクロインタラクション
 - ・選択肢が増えれば、ルールも増える

選択肢は容赦なく切り捨てる

- ・選択肢は絞り、代わりに賢いデフォルトを与えるのがよい
- ・ユーザーが次にとりそうな行動を目立つように表示するのがよい
 - ・他の選択肢を省く
 - ・ボタンを大きく
- ・選択肢を提示しなければならない
 - ・提示方法が結果を左右

リストの最上部、最下部は印象に残りやすい
目立つように表示されれば選ばれやすい

- ・エッジケース
 - ・まれにしか生じない解決しがたい問題
 - ・選択肢を絞りこむと取り除ける
 - ・エッジケースが存在する

例外ケースをデザインすることになる
マイクロインタラクションが台無し

コントロールとユーザー入力

- ・選択肢
 - ・操作が単純

各コントロールがそれぞれ別の機能

- . めったに行わないマイクロインタラクション
- . すべての選択肢を明示
- .. 前提知識を少なく

- ・結果の認識が容易

1つのコントロールが複数の機能

- . 繰り返し行われるマイクロインタラクションの場合
- . 素早い操作が必要で操作ミスが起きにくい場合は例外

- ・テキスト入力欄
 - ・入力内容の許容範囲をできるだけ広く

画面の背後で直す

必要多様性

最少有効多様性

エラーの防止

- ・ポカよけ
 - ・従う

それ自体がエラーを受け付けない

- ・メッセージ
 - ・正しく操作したときには何も表示しない
 - ・システムが適切な反応を返せないときのみ表示

ポップアップメッセージは怠け者のための手段

マイクロコピー

- ・ラベル、表示などごく短いテキスト
 - ・固定的なフィードバック
 - ・フィードバックフォワード

アルゴリズム

- ・処理順序
- ・分岐
- ・繰返し
- ・変数

フィードバック

■一貫したフィードバック

求められる

望ましい行動を肯定的に強化促進

■ルールを理解を助ける

フィードバックの真の目的

起動されたら知らせる必要

- ・何が行われた
- ・結果として何が起きた

実際にどう機能するかを教える必要はない

- ・どのような感じで処理されるのか
 - ・メンタルモデルを構築できるように
- ・何が出来て、何が出来ないか

■提供しなければならない場面

手動トリガーの直後、ルールを手動で調整しているとき

システムトリガーにより著しい変化がもたらされた

ユーザーがルールを破りそう、破った場合常に

システムがコマンドを実行出来ない

重要な処理なら進捗状況を表示

原則 1: フィードバックでユーザーに負担をかけない

■対象は人間

マイクロインタラクションのフィードバックで伝えること

- ・何かが起きた
- ・ユーザーが何かを実行した
- ・処理が始まった
- ・処理が終了した
- ・処理が続行中
- ・ユーザーに「それはできない」と知らせる

人間は自分の期待が裏切られたときに学習する

原則 2: フィードバックは的はずれなものであってはならない

■省略の美

原則 3: 最少のフィードバックで最大限のメッセージを伝える

原則 4: 見落とされがちなものを使ってメッセージを伝える

■個性を表現する手段

人間的な反応を示す

- ・対する人間の反応もよくなる
- ・製品を擬人化

ユーザーがイライラする瞬間こそが、緊張を和らげる「個性」を発揮する場

- ・表面的な個性を目指す

■方法

視覚

- ・何を見せたらよいかをまず考える
- ・できるだけ目立たない方法で表示
- ・ユーザーが知りたい
 - ・時間
 - ・残り作業
 - ・未読メッセージ
- ・目的は明確化
 - ・混乱、重複は避ける

- ・場所
 - ・ユーザー入力箇所周辺が理想的

聴覚

触覚

アニメーション

- ・脳は動きに強く反応する
 - ・アニメーションは控えめに

使わないに越したことはない

軽快

認識負荷が小さい

ちょっとしたアニメーションをうまく使うこと

- ・重要
 - ・動きを見れば機能を的確に予測できる
 - ・最良のアニメーションとはユーザーに何かを伝えるもの
- ・欠かせない特性
 - ・高速
 - ・滑らか
 - ・自然
 - ・シンプル
 - ・目的がある

コンテキストを保つ

たった今起きたことを説明する

オブジェクト間の関係を示す

焦点を鮮明にする

パフォーマンスが上がったように感じさせる

バーチャルスペースという錯覚を生み出す

・トランジション

.. どのようにスライドして何に切り替わるか

.. 仮想的な移動を表現

「もっと使って」と働きかける

- ・長さ
 - ・これくらいはあるべきと考える半分にする

可能ならさらに半分

メッセージ

- ・最低でも正確である必要がある
- ・短く明確に
- ・エラーの場合修正方法も伝える
 - ・修正手順が入ると理想的

サウンド

- ・聴覚刺激は素早く脳に到達
- ・強い反応を引き起こす

使用は控えめにする

- ・聴覚に訴える
- ・強調

ユーザー動作に念押し

・クリック音

- ・警告

システム動作を通知

- ・2種類
- ・イヤコン

アイコンのもじり

特徴ある短い音

- ・言葉

ハプティクス

- ・触覚を通じて情報を伝達する技術
- ・視覚、聴覚に比べ感度は劣る

■ルール

フィードバックを変える

- ・コンテキストによる変化
 - ・夜には音量を上げる / 下げる
- ・継続時間
- ・程度
- ・繰返し

ループとモード

■モード

ルールが二股以上

マイクロインタラクションでは極力避ける

- ・間違いのもととなる

設ける理由

- ・頻繁には利用しない操作がある
 - ・モードを分けなければ、主目的の操作が煩雑化

よくあるモード

- ・設定モード

設けなければならない場合

- ・可能な限り専用の画面を用意

■バネ仕掛けのモードと 1 回完結のモード

バネ仕掛けモード

- ・疑似モード
 - ・特定の物理操作時アクティブ

キーボードの shift キー等

■ループ

定められた間だけ繰り返すサイクル

直接または、間接的にルールで示される

形式

- ・回数制御
- ・条件制御
- ・コレクション制御
- ・無限

種類

- ・開ループ
 - ・フィードバックに反応しない
- ・閉ループ
 - ・フィードバックに対応する
 - ・自己調整

ロングループ

- ・長い期間にわたる閉ループ、時と共に改良されていく閉ループ
- ・ロング・ワオ (The Long Wow)
 - ・ある程度の期間使い続けることで体験できる

新しい使いごごち
機能

- ・ユーザーを長くつなぎ止める効果
- ・ループで実現できる

ユーザーが同じマイクロインタラクションに戻ったときに、前回は記憶

- ・段階的な開示や省略
- ・熟練ユーザー

手早く操作できるショートカット
高度な機能
提示内容を段階的に減らす

実践例

■マイクロインタラクション改善方法

記憶に残る瞬間にする必要があるか

ゼロから始めようとしているのか？

一番重要なデータは何か？前面に出せるか？

カスタムコントロールが適しているか？

ヒューマンエラーを予防できるか？

見落とされがちなものを活用しているか？

上級ユーザー向けに不可視トリガーを作れないか？

テキストやアイコンは自然か？

アニメーションを加えれば、もっと表情や動きが出せるだろうか？

他に加えるべきフィードバックの手段がないか？

ユーザーがこのマイクロインタラクションに 2 回目に接したとき何が起こるか？100 回目には？

■小さな視野で考えよう

大規模システムを人間味あふれるものにする

- ・ マイクロインタラクション

細部が体现する

- ・ 作り手のユーザーに対する
 - ・ 気配り
 - ・ こころづかい
 - ・ 思いやり
 - ・ おもてなしの心
- ・ 万人が欲しているもの

マイクロインタラクションのテスト