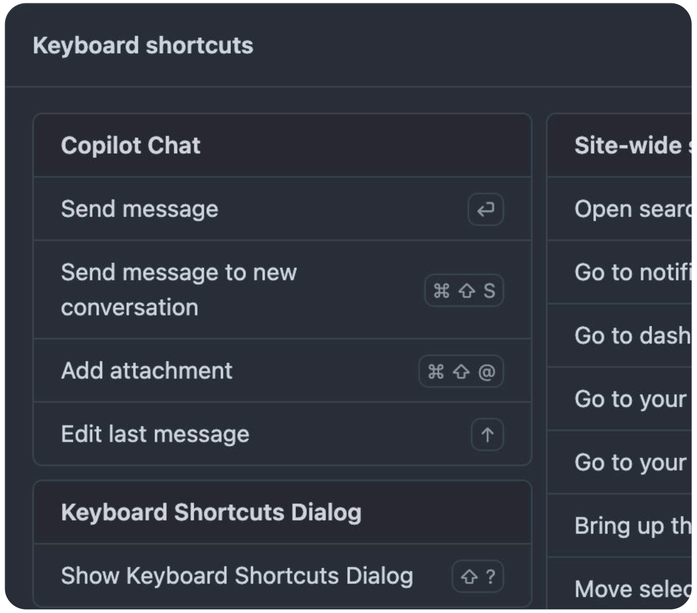
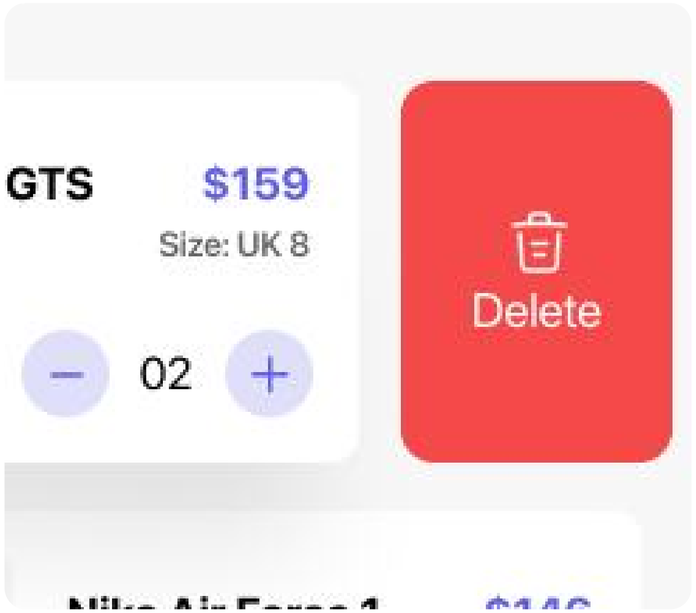
Acceleratorというui
Webでの実践

@nekobato

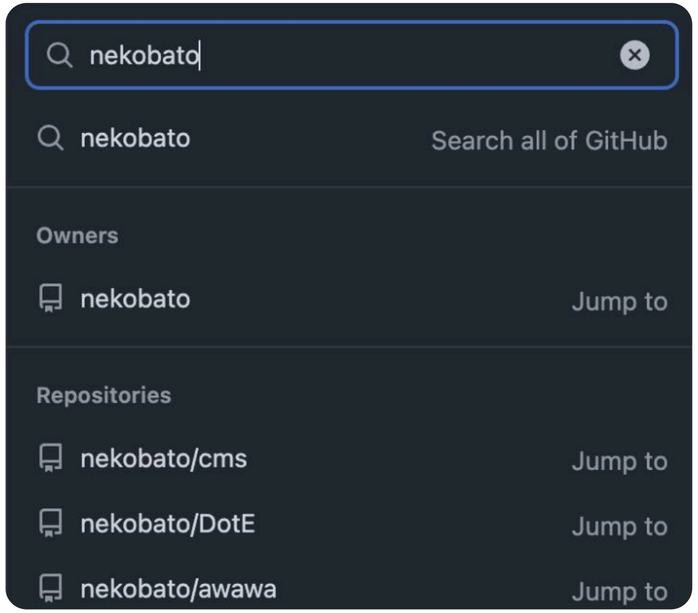
操作を短縮する、熟練者のための第二の導線



Keyboard Shortcut



Swipe / Hover Action



Predictive input / Inline Action

などなど

代表的なアクセラレーター

Ctrl + C, Ctrl + V

MacだとCommand + C, Command + V

a11yとの区別

スクリーンリーダー

name, role, value と読み順を整え、意味を
機械へ適切に伝える

キーボード到達性

誰もが目的に到達し、マウス無しでも完遂できる
基盤

アクセラレーター

熟練者のための第二の導線
既存導線を置き換えず短縮する

なぜWebサービスで
アクセラレーターを
実装するのか

- 操作を置換せず短縮する第二導線として
  熟練者の満足と定着を促す

- 初心者の利便性を壊さないので
  実装を積んだ分だけ利用できる

気を付けること

衝突を避ける

より上位の操作を上書きしない
または適切に無効化し、共存できる方法で実装する

発見可能性

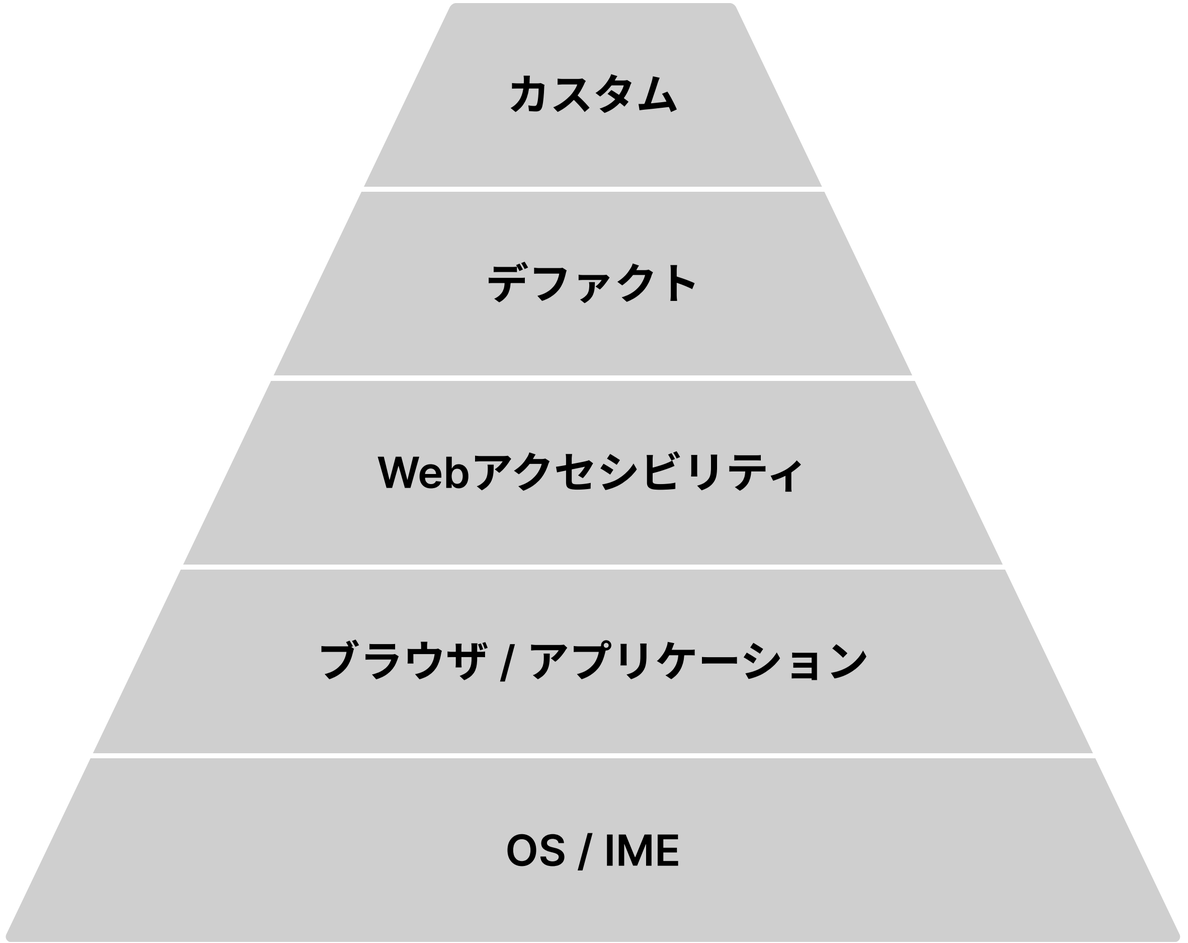
熟練度に従ってヒントを表示し
操作を学ぶ機会を与える
チュートリアル/Tooltip/併記

コンテキストを限定する

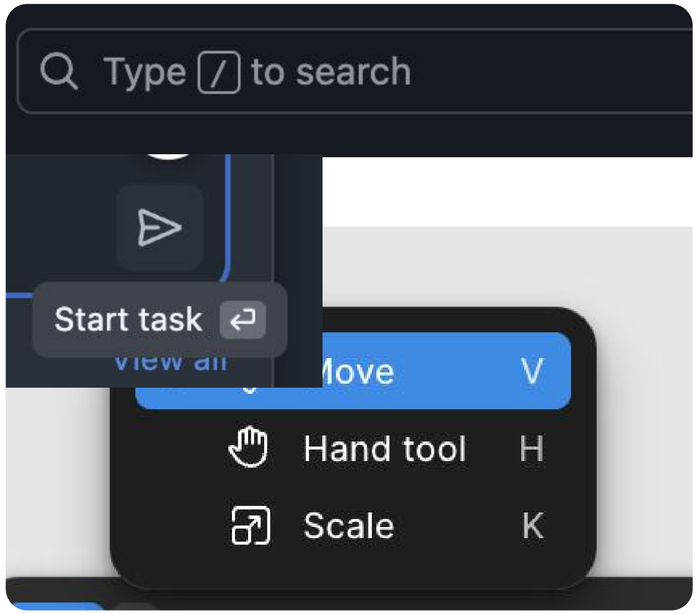
ページとフォーカスのコンテキストを限定し、ユーザーの意図や他の操作を認識して区別する

衝突を避けて積み重ねる

優先するべき基底の操作との衝突を避ける



発見可能性



併記

一覧を用意

チュートリアル

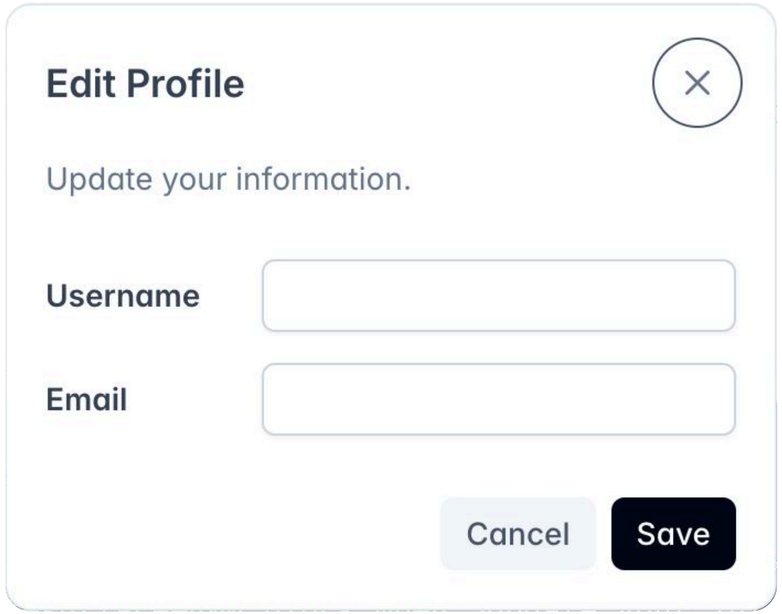
コンテキストを限定する

完全に衝突を避けることは難しいが、
ページやフォーカスなどのコンテキストを限定することで
限定的にアクセラレーターを利用可能にする（赦しを得る）

Dialog

Floating要素が出現している間は
Escで閉じられてほしい

Focus Trap (a11y)



ESC to Close (a11y + Accelerator)

動画プレイヤー

Desktop

再生/一時停止：**Space** **K**
短いシーク（5~10秒）：**←** **→** **J** **L**
ボリューム：**↑** **↓**
ミュート：**M**
フルスクリーン：**F** **Esc** (解除)
字幕 toggle：**C**
%ジャンプ：**0-9**
速度調整：**<** **>**
フレーム単位：**,** **.** **Shift + ←** **Shift + →**
ショートカット一覧の呼び出し：**?**

Mobile

短いシーク：**左右ダブルタップ**
早送り：**長押し**

Web上での実装とライブラリ

ショートカットキー設定

Vanilla: HotKeys.js, ...

Vue: @vueuse/core, defineShortcuts(Nuxt UI), ...

React: tinykeys, react-hotkeys-hook, ...

フレームワーク使ってるなら
自分で書くのが柔軟という感想がある

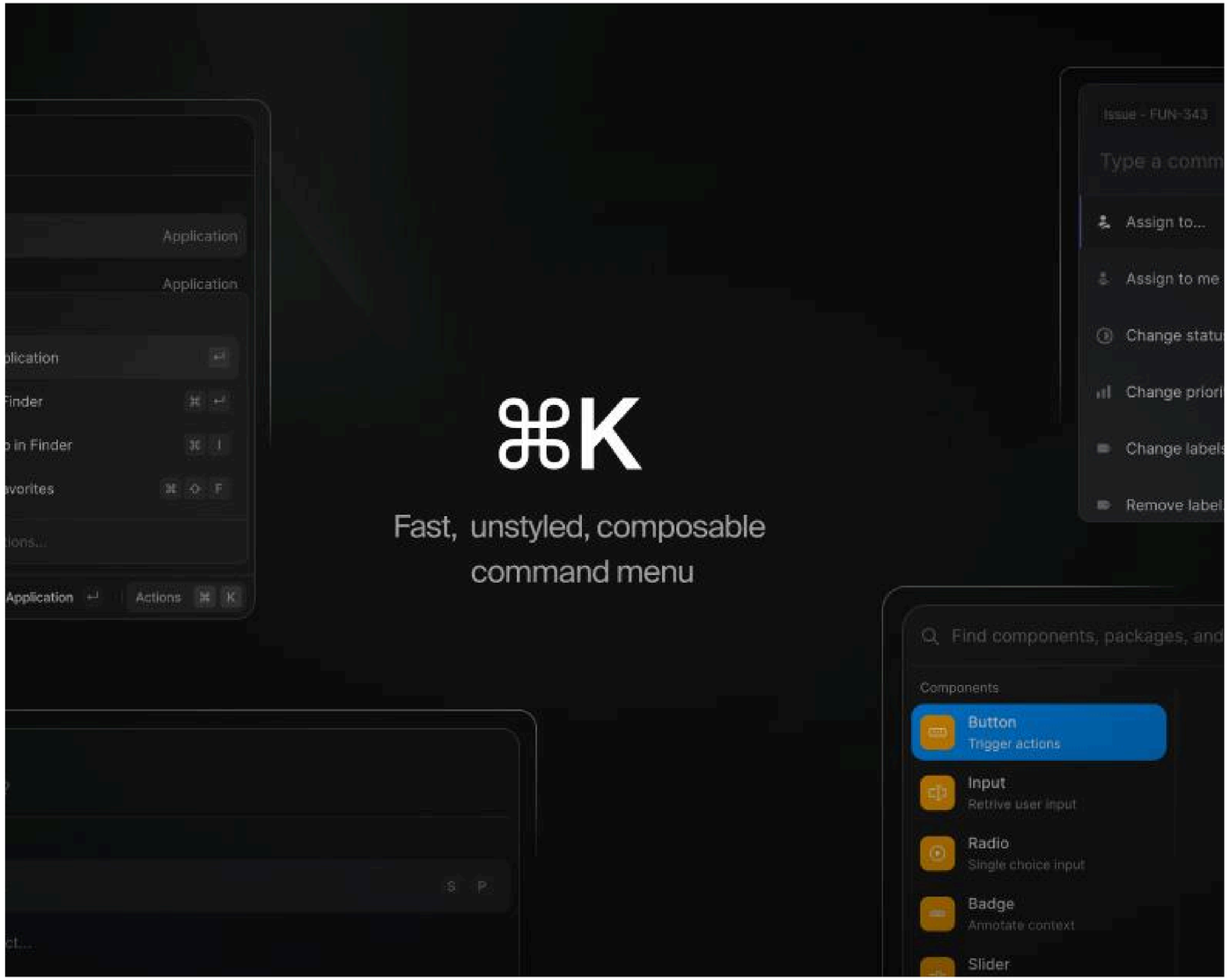
録画型入力 (recorder)

React HotkeysのuseRecordHotkeysとか

```
const isEditable = (el) =>
  el && (el.isContentEditable ||
/^(INPUT|TEXTAREA|SELECT)$/.test(el.tagName));

addEventListener(
  "keydown",
  (e) => {
    // IME入力中は無効
    if (e.isComposing) return;

    const mod = e.metaKey || e.ctrlKey;

    // 入力中はショートカットを無効化
    if (isEditable(e.target) && e.key !== "Escape") return;

    // "/" で検索
    if (e.key === "/" && !e.shiftKey && !e.altKey && !mod) {
      e.preventDefault();
      document.getElementById("search")?.focus();
      return;
    }

    // "?" で一覧
    if (e.key === "?" || (e.key === "/" && e.shiftKey)) {
      e.preventDefault();
      dispatchEvent(new CustomEvent("open-shortcuts-overlay"));
      return;
    }
  },
  { capture: true }
);
```

コマンドパレット

最近はCommand(meta) + Kで開く
汎用コマンドパレットも見かける



特殊キーはアイコン使うのが便利

Lucide

Phospher

Material Design Icons

よくある問題(ショートカットキー)

? と書くのか、Shift + / と書くのか

キーボードレイアウトによって組み合わせが違うので
code（物理キー）で取得してShift + / をと表示する、が無難

OSの判別  

```
const isMac = () => navigator.platform.toLowerCase().includes("mac");
```

実装の用法用量の守り方

実際に実装してみて空気を読んでissueに耳を傾けるしかない
デファクトが何なのかは大手を見て空気を読む

でした。

References

Accelerators Maximize Efficiency in User Interfaces
https://www.nngroup.com/articles/ui-accelerators/

Flexibility and Efficiency of Use (Usability Heuristic #7)
https://www.nngroup.com/articles/flexibility-efficiency-heuristic/

Keyboard-Only Navigation for Improved Accessibility
https://www.nngroup.com/articles/keyboard-accessibility/?utm_source=chatgpt.com