

# レイアウトの自由度とキー操作性を両立した ノートテイキングアプリケーションの開発 — キーボード特化型ビジュアルノートアプリ: 鍵記 —

## 1. 背景

キーボードはテキスト入力に特化した入力インターフェースである。1700年代に出現したタイプライターから1900年代に隆盛したワードプロセッサを経て、現在我々が使うコンピュータキーボードへと発展してきた。キーボードによるPC操作(キー操作)を熟達したいという欲求は普遍的なもので、本来マウス操作が多分に必要なGUIアプリケーションにおいても、キーボードショートカットによってキー操作性を高める試みがなされてきた。

本プロジェクトは代表的なGUIアプリケーションの一つであるノートアプリを題材とするが、この中でも特に、キャンバスにオブジェクトを自由に配置していく「ビジュアルノートアプリ」に着目した。昨年度に Obsidian, Logseq といった世界的に人気なノートアプリがビジュアルノート機能に対応したり、米スタートアップファンドの Y Combinator が多機能ビジュアルノートアプリ Heptabase に出資したりと、近年大きな盛り上がりを見せているジャンルである。

## 2. 目的

ビジュアルノートアプリは一般に、キー操作と相性が悪い。キーボードが得意とするテキスト入力以外に、グラフィックスを大量かつ頻繁に扱うためである。マウス操作を一切行わず、キー操作のみでビジュアルノートをとることができるアプリはこれまでに存在しなかった。本プロジェクトの目的は、流行の只中にあるビジュアルノートアプリにおいて、マウスを使わずに操作できる超キー特化なものを作ることである。オブジェクトの配置・移動・編集など、ビジュアルノートの醍醐味である豊富なグラフィックスの操作を全てキーボードで完結することを目指した、ロマンに溢れるプロジェクトである。

## 3. 開発の内容

本プロジェクトでは、ブラウザ上で動作する新しい操作感のノートアプリ「鍵記」を開発した。鍵記では、キャンバスにオブジェクトを配置していくことでビジュアルノートを作っていく。

キャンバスは方眼紙のようにマス目上に区切られており、カーソルやテキストブロックはこのマス目を基準として位置と大きさが定義される(図 1)。ピクセル単位で位置や大きさが定義される従来のアプリと異なり、粗いマスを単位としてオブジェクトを移動することで、素早くオブジェクトを移動できる。また、Vimium や qutebrowser にインスパイアされたヒント選択機能や Markdown のライブプレビュー、グループ化やリレーションの追加など、様々なキーアクションをサポートしている。

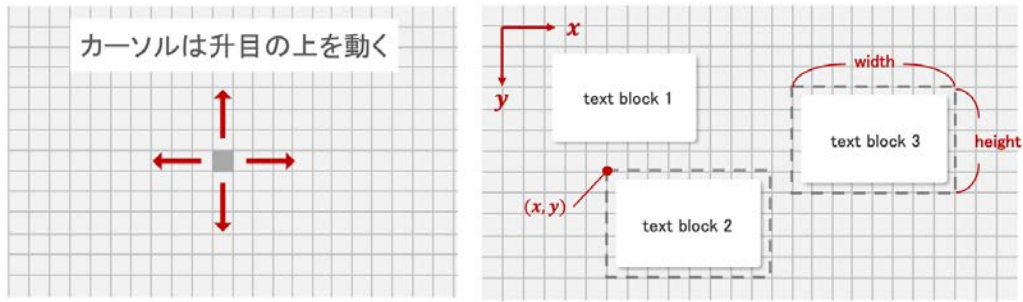


図 1: 鍵記のキャンパスのイメージ(左:カーソルの振る舞い・右:テキストブロック)

#### 4. 従来の技術(または機能)との相違

鍵記の特徴は何と言っても、これまでマウス操作が主流であったビジュアルノートの操作を、全くマウスを使わずにキーボードのみで可能にしたという点である。これにより、図 2 のように、空間的に関係を持たせたノートをキー操作のみで作成できるようになった。

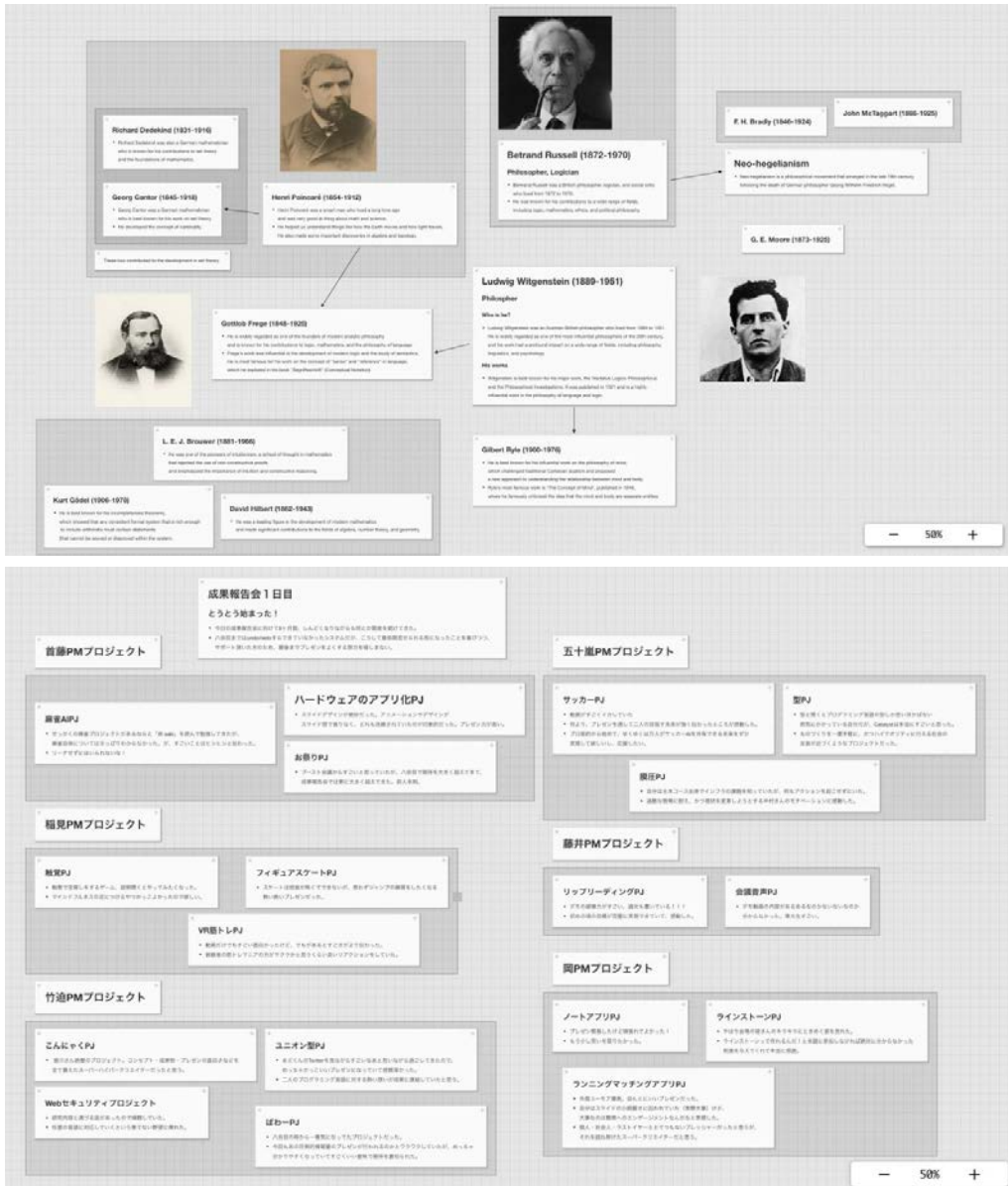


図 2: 鍵記によって作成したノート例

また、キー操作性が高いからといって、使いづらければ元も子もない。キー操作性の高いアプリケーションにおいて懸念されるのが「キーショートカットが複雑怪奇で覚えづらい」という事態であるが、鍵記は段階的なショートカットを用いることでその解消を試みた(図 3)。

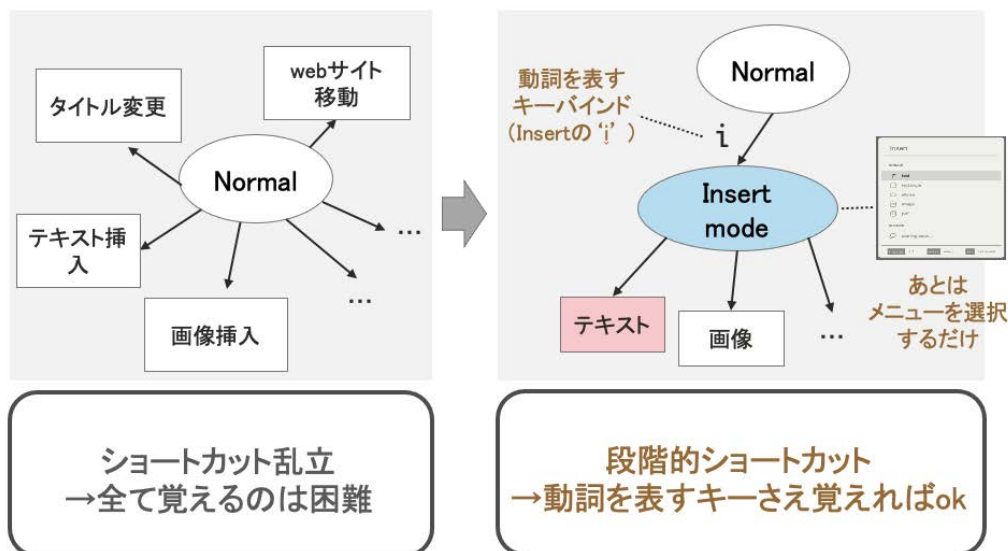


図 3: 段階的なショートカットの説明(テキストの挿入の例)

例えば、「テキストを挿入したい」ならば、動詞に相当する「挿入」を表すキーバインド`i`を入力し、挿入したいオブジェクトを選択するモーダル(図 4)に飛ぶ。そこで目的の項目(ここでは「text」)をキー移動で選択すると、オブジェクトが挿入できる。この工夫により、例えば「〇〇に移動したい」といった場合も同様のフローによってシンプルに操作が行える。

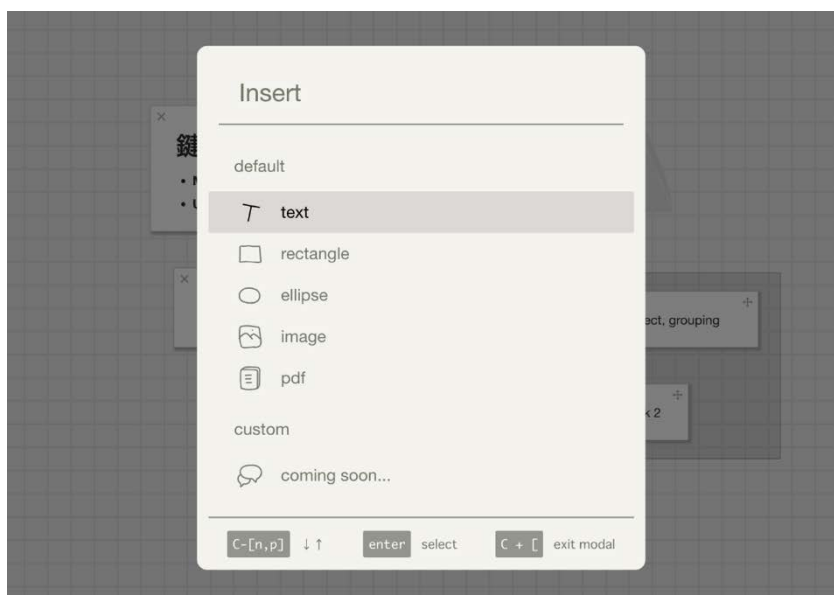


図 4: 挿入したいオブジェクトを挿入するモーダル画面

## 5. 期待される効果

これまでのビジュアルノートアプリにおいては、頻繁なマウス操作を要請する Mouse-Driven な GUI が主流であった。本プロジェクトはキーボード操作のみでビジュアルノートを記述できる「鍵記」を開発し、Keyboard-Driven Visual Notes という新しいジャンルを創出したと言える。「Vim や Emacs は慣れるのが難しくてハードルが高い」という方でも、GUI なら幾分気楽に触ることができるだろう。鍵記は Vim や Emacs の愛好家のみならず、キーボード愛好家やある程度 PC 操作に習熟したユーザを中心として、使用が広がっていくことが期待される。

## 6. 普及(または活用)の見通し

ノートアプリは一般に機能が多く、人にとって求める機能も違うことから、本プロジェクト期間内ではプロトタイプの実装に留まった。鍵記はこれから、キーカスタマイズ機能の追加やプラグインシステムの実装に加え、分かりやすいユーザガイドや動画の作成・公開など、多くのユーザを獲得するための活動を行っていく。また、ノートアプリのみならず、一般にグラフィックスをキー操作する体系を形成していくことで、後続の Keyboard-Driven GUI に影響を与えられる存在感を示してゆく。

## 7. クリエータ名(所属)

稲葉皓信(京都大学 大学院情報学研究科 通信情報システム専攻 1年)

(参考)関連 URL

鍵記 Playground: <https://kenki-editor.vercel.app/>