

カスタマイズ可能なQuickLook機能の提案

学生証番号:244208 ネットワークメディア学科 浦中 信吾（荻原研究室）

1 はじめに

Macでは、FinderでQuickLook[1][2]の機能を利用してファイルの内容を簡単に確認できる。QuickLookは高速なプレビューが可能だが、書き込み等の機能は用意されておらず、また、利用者が機能をカスタマイズすることも容易ではない。

本研究では、QuickLookからシェルスクリプトを起動させることで機能の拡張を行う方法を検討した。

2 QuickLook

QuickLookでは、選択したファイルの拡張子に対応したプラグインが自動的に起動する。プラグインは自作して追加可能である。

本研究では、機能の拡張を行うために新しいファイル拡張子を用意し、それに対応するプラグインを作成した。

3 プラグインの概要

本研究で作成したプラグインは、選択したファイルのプレビュー要求がきた時に、プログラム内からシェルスクリプトを動作させる。シェルスクリプトはファイルの絶対パスの情報を利用して動作出来るが、ファイルの内容を表示する以外にも、次節に示すような多様な機能を記述しておける。

実装に使用した言語はObjective-Cである。

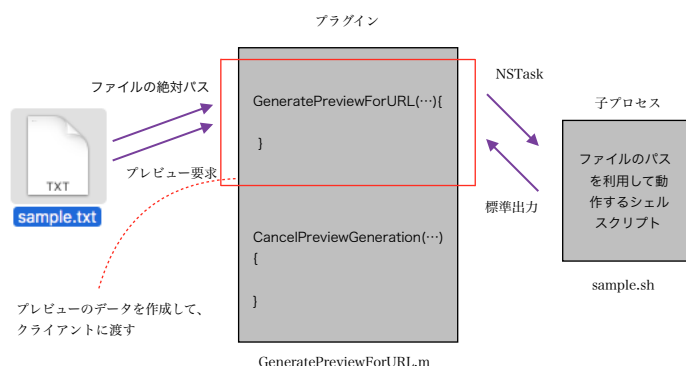


図1.実装プログラムの流れ

4 実装内容

プラグインを動作させるためのファイルには、ファイル名として、システムが認識する本来の拡張子の前に、プラグインの機能を指定する文字列を置く。

プラグインは次の4つの種類を作成した。

1. ディレクトリ内のすべてのファイルに対し、検索用ペストボードの内容を検索して表示(図2)
2. ディレクトリ内のすべてのメモ書きファイルを連結して表示
3. ディレクトリ内のファイルの中で、更新時間が最新のファイルを表示
4. ディレクトリ内のファイルをランダムに選んで表示

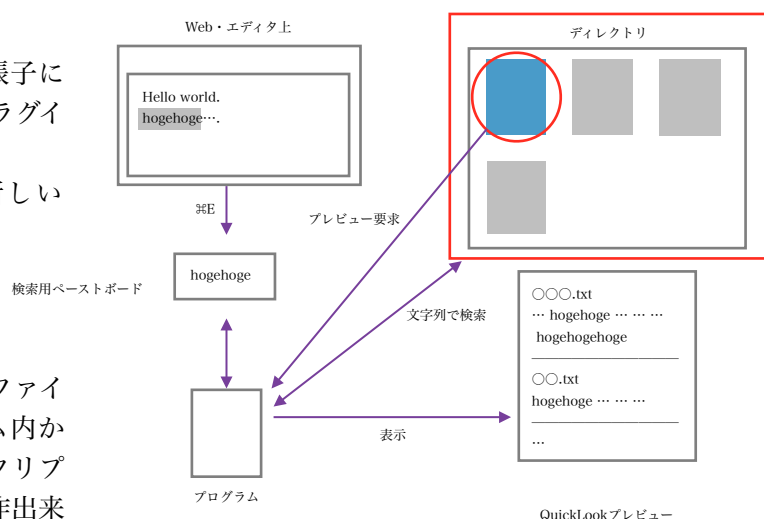


図2.検索用ペストボードを利用したプラグイン

5 まとめ

プラグインからシェルスクリプトを動作させる方法でQuickLookのカスタマイズが確認できた。本研究で実装した機能以外にも、シェルスクリプトの記述次第で様々な機能が実現できる。また、他のスクリプト記述言語を利用することもできるであろう。

参考文献

- [1] Apple, “ファイルやフォルダをクイックルックする”, <https://support.apple.com/kb/PH18813>
- [2] Apple, “Quicklook Programming Guide”, https://developer.apple.com/library/mac/documentation/UserExperience/Conceptual/Quicklook_Programming_Guide/

スマートフォンの操作に特化した 計算機アプリの作成

学生証番号 244523 ネットワークメディア学科 佐藤 睦（荻原研究室）

1 はじめに

本研究では昨年度の卒業研究[1]を取り上げて、スマートフォンの操作に特化した計算機アプリの作成をした。スマートフォン独特の機能にスワイプ操作やフリック操作がある。しかし、一般的な計算機アプリはスワイプ操作などを使用しないものが多い。

本研究ではスワイプ操作を多く取り入れ、数値入力方法をフリック操作とする計算機アプリの作成を試みた。また、アプリ開発には `enchant.js`[2]、`PhoneGap`[3]を用いた。

2 概要

この計算機アプリはデータをボールで表し、それらをスワイプによって衝突させることで計算を行う。本研究ではさらにデータをグループとしてまとめて表示し、それらの合計・平均を計算を行う機能を追加した(図1、図2)。

また、数値入力方法はフリック操作での入力方式としている。0のボタンで1～4、5のボタンで6～9を入力する。

3 グループ機能

通常の電卓で表示されるのは計算結果だけだが、本アプリのグループ機能では、計算の対象である複数のデータをメモリに表示しておける。ボールの移動によってデータの追加や削除ができ、それに応じて合計、平均は自動的に再計算される。グループは複数個を同時に構成できる。

例えば、買い物の合計額を予算に収めたい場合、購入する品の取捨選択が容易である。品物の種類別にグループを作ることできる。関数電卓のような複雑な計算はできないが、日常での様々な用途に役立つと考えられる。

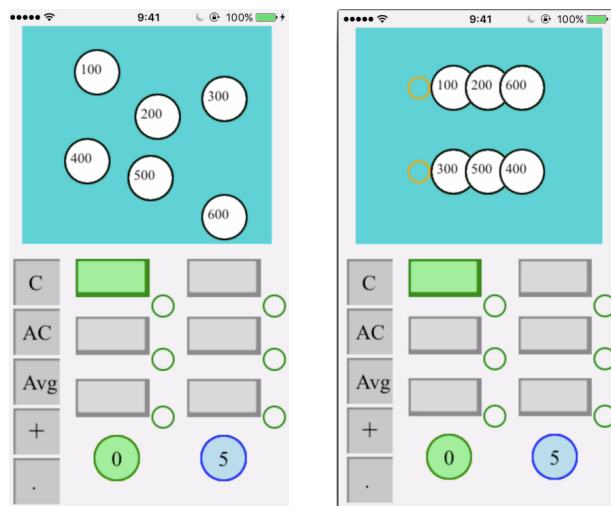


図1 メモリが散らばっている状態

図2 メモリがグループ化された状態

4 まとめ

数値入力方法をフリック操作としたが、数値が10個しかないため入力速度はテンキーと変わらず、テンキーに割り当てていたスペースを他の機能の実装に利用出来る。

スワイプ操作やフリック操作は一般的な電卓とは使用方法が異なるため、使うには慣れが必要である。一方、データを目で見て確認したり、データをグループとしてを扱えることにメリットがあると感じた。

参考文献

- [1]安田悠輝, スマートフォン特有の操作性を生かした計算機アプリの開発, 京都産業大学コンピュータ理工学部, 卒業論文 2015
- [2]UEI, `enchant.js`, <http://enchantjs.com/ja/>
- [3]Asial Corporation : `PhoneGap Fan`, <http://phonegap-fan.com/about/>

ブラウザの閲覧履歴を操作するアドオンの開発

学生証番号 244677

ネットワークメディア学科

武村修平 (荻原研究室)

1 はじめに

本研究ではブラウザの閲覧履歴を操作するアドオンの開発を行った。

作成したアドオンは、閲覧履歴を一括削除してファイルに書き出す機能、ファイルから履歴を復元（リカバリー）する機能、および特定のURLを履歴から削除する機能を実装している。調査した範囲ではこのような機能は既存のアドオンには存在していなかったが、ブラウザを用いたプレゼンテーションを行うような場合に有用な機能である。

2 ブラウザの閲覧履歴

Firefoxにはplace.sqliteと呼ばれるデータベースファイルが存在しており、ブックマーク、お気に入りのアイコン、閲覧履歴、入力履歴等がそれぞれのテーブルに分けられて保存されている[1][2]。

その中でも本研究で使用するテーブルは、moz_places、moz_historyvisitsの2つのテーブルであり、moz_placesテーブルにはページのタイトルやURL、閲覧回数等を示すフィールド、moz_historyvisitsテーブルには閲覧した日付やmoz_placesテーブルのidを参照する等のフィールドを備えている。

履歴の一時削除では、テーブルのレコードをファイルに書き出し、テーブルの情報は削除する（図1①）。リカバリーは逆に、ファイルから読み込んだ情報をテーブルに追加する（図1②）。

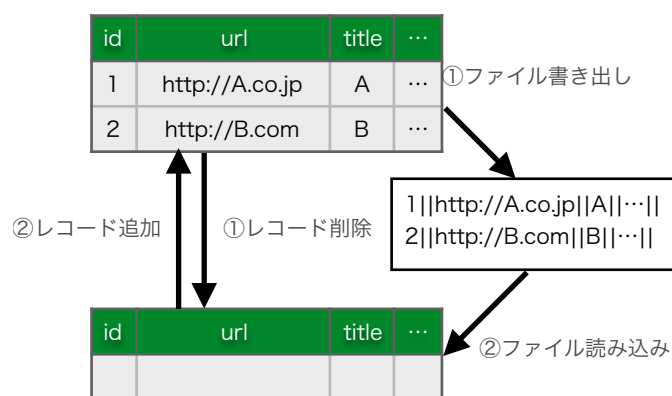


図1 履歴の削除とリカバリー
(moz_placesテーブル)

3 アドオン概要

開発したアドオンはFirefoxブラウザの右クリックで表示されるコンテキストメニュー(図2)から各項目を選択

することで実行する機能と、特定のURLを持つwebページ上で自動的に実行される機能の二つに分けられる。

実装した機能の内訳は以下のとおりとなっている。

【コンテキストメニューから実行する機能】

- ・履歴の一時削除
- ・リカバリーの機能

【特定のwebページで実行される機能】

- ・特定のURLを持つ履歴レコードの削除



図2 コンテキストメニュー

4 考察

本アドオンは、他人に自身の履歴情報を見られることを防ぐのに有効である。

しかし、今後の課題としてユーザー自身がURLを設定するための入力ウィンドウの実装や、個々の履歴レコードに対する一時削除・リカバリーの機能といった改善点が挙げられる。

5 まとめ

今回開発したアドオン自体の機能は決して多いわけではない。しかしアドオンの性質上、本研究で開発した新たな機能を持つアドオンと既存のアドオンとを組み合わせることで、さらに利便性の高い履歴操作が実現できるだろう。

参考文献

[1]MDN, Places, <https://developer.mozilla.org/ja/docs/Places> (2013)

[2]MDN, The Places database, https://developer.mozilla.org/ja/docs/The_Places_database (2008)

盗み見に強い小型タッチパネル端末向け スクリーンロック手法の提案

244776 インテリジェントシステム学科 天良 勇人 (荻原研究室)

1 はじめに

スクリーンロックとはスマートフォンなどにおいて、所有者以外の端末利用を制限する認証機能である。

タッチパネルを用いた既存手法は、周囲に人の居る環境下で利用した際、盗み見により解除手順が推定される恐れがある[1]と考え、盗み見についての調査と、対策の提案を行った。

2 対盗み見強度推定調査

様々な角度からロック解除場面を撮った動画を見ることで、解除手順の推定が可能かを、荻原研究室、林原研究室、その他同期の皆様の御協力を得て、調査した。

結果、既存手法のPIN、パターンロックは画面が見えない角度から場面を撮った動画でも、手順の推定が可能であることが分かった。

3 手法の提案

既存手法は解除に必要なタッチ入力的位置が毎回同じなため、指の動きの観察で解除手順が推定されたのではないかと考えられる。

そこで、ボタンの位置にランダム要素を取り入れ、タッチ位置を一定で無くせば、盗み見による推定を防ぐことができるのではないかと考え、4つの手法を提案した(図1～4)。提案手法はフラットデザイン[2]を取り入れ、iPodTouch上で動作するアプリケーションとして開発を行った。

4 提案手法の説明

まず既存のPINのボタン位置をランダムにしたランダムPIN(図1)、自転車の鍵などで馴染みのあるダイヤルロックを提案した。その後、操作性や視認性について考え、数字をなぞって解除を行うスターロック(図3)を提案した。

数字ボタンを環状に配置したフラワーロック(図4)は、調査での被験者の回答傾向から、斜め方向の入力は指の観察によって推定されにくいのではないかと考え、調査後に提案・開発した。

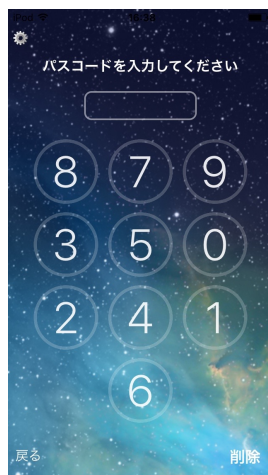


図1. ランダムPIN



図2. ダイヤルロック



図3. スターロック

5 アンケート調査

対盗み見強度推定調査と平行して、実際に提案手法のサンプルを見せ、感想や改善点などを回答してもらった。結果、「盗み見に対しての効果が期待できる」「ランダムだが押しやすい」という評価の反面、「操作が面倒」「解除に時間がかかる」等の回答も見られた。



図4. フラワーロック

6 まとめ

ランダム要素の取り入れにより、盗み見に対し強度が得られた反面、視認性が悪くなり、解除時間が増えてしまった。解除手順を知る所有者には使いやすい、知らない人間には困難なよう、表示方法などに工夫を加え、解除時間や手間を減らす事が今後の課題である。

参考文献

- [1] 岡嶋裕史, “ハッカーの手口 ソーシャルからサイバー攻撃まで”, PHP新書(2012).
- [2] 宇野雄, “フラットデザインで考える新しいUIデザインのセオリー”, 技術評論社(2014).