

UI/UX 創造を支援するログ解析ツール

—ユーザの操作を検索し、発見する—

1. 背景

スマートフォンと開発環境の普及に伴い、多くの人々が UI を開発することが可能となった。しかし、開発した UI についてユーザから適切なフィードバックを受ける事は難しく、利用する様子を観察することは困難であった。特に、ユーザの指の動きについては着目されていなかった。

2. 目的

開発者が発明・導入した新規の UI に対して、その UI を実際のユーザがどのように扱っているかを明らかにし、ユーザの利用状況を分析することによって、UI の評価や改善点の抽出を可能とする。特に、ユーザから集めた膨大なデータの中から、開発者の議論の元となるような指の動きの発見を目指す。

3. 開発の内容

本プロジェクトではユーザから操作ログを収集し検索を行うためのシステムとして、HeartMap を開発した。HeartMap は 4 つのモジュールから構成される。

3.1. HeartMap ライブラリ

開発者がアプリケーションに組み込み、ユーザから操作ログを収集するためのライブラリである。開発者は図 1 のようにトークンをライブラリに渡すプログラムを 1 行記述するのみで、操作ログの収集が可能となる。

```
[HMLogManager managerWithToken:@"xxxx-xxxx-xxxx-xxxx"];
```

図 1 ライブラリのインスタンス生成時の命令

3.2. 集計サーバ

HeartMap ライブラリを用いて収集した操作ログを集計し、2 分間の操作を集めたセッションを生成する。タッチ操作毎に画面遷移と UI の利用状況を紐付ける。

3.3. 検索サーバ

集計サーバで生成したセッションを保存し、検索を行い統計情報や再生可能なセッションデータを出力する。検索には、時系列順に並べた UI の利用状況や、ユーザの属性をクエリとして定義して用いる。UI の利用状況を検索するために、UI の操作によって呼び出されたメソッドの情報を用いる。メソッドの情報は事前に登録され、IDとして検索を行う。

1. TOUCH(...)

タッチを検索するクエリ。内部にはメソッド ID を記述する。OR で複数のメソッド ID を記述することで、OR 検索が可能である。タッチの入力時間を不等号で指定することが可能である(例: “TOUCH(1 OR 2 > 1.0)” →メソッド ID1 または 2 で 1 秒以上の入力時間)。このクエリを複数並べた場合は、時系列順に全てのタッチ操作が行われたセッションを検索する。また、タッチ操作を行わなかったものとして NOT TOUCH(...) の検索が可能である。

2. USER(...)

ユーザを検索するクエリ。MODEL や OS、TYPE (HeavyUser 等) について検索が可能である。また、USER(IN(...)) と記述することで、内包するクエリのセッションを持つユーザを指定することが可能である。また NOT USER(...) でクエリに該当しないユーザを指定することが可能である。このクエリを複数並べた場合は OR 検索として扱われる。

3. VERSION(...)

バージョンを検索するクエリ。このクエリを複数並べた場合は OR 検索として扱われる。

3.4. HeartMap クライアント

検索サーバにアクセスし、統計情報の表示やセッションを再生し指の動きを確認することが可能である。HeartMap クライアントは MacOS 向けのネイティブアプリケーションである。全体の GUI を図 2 に示す。

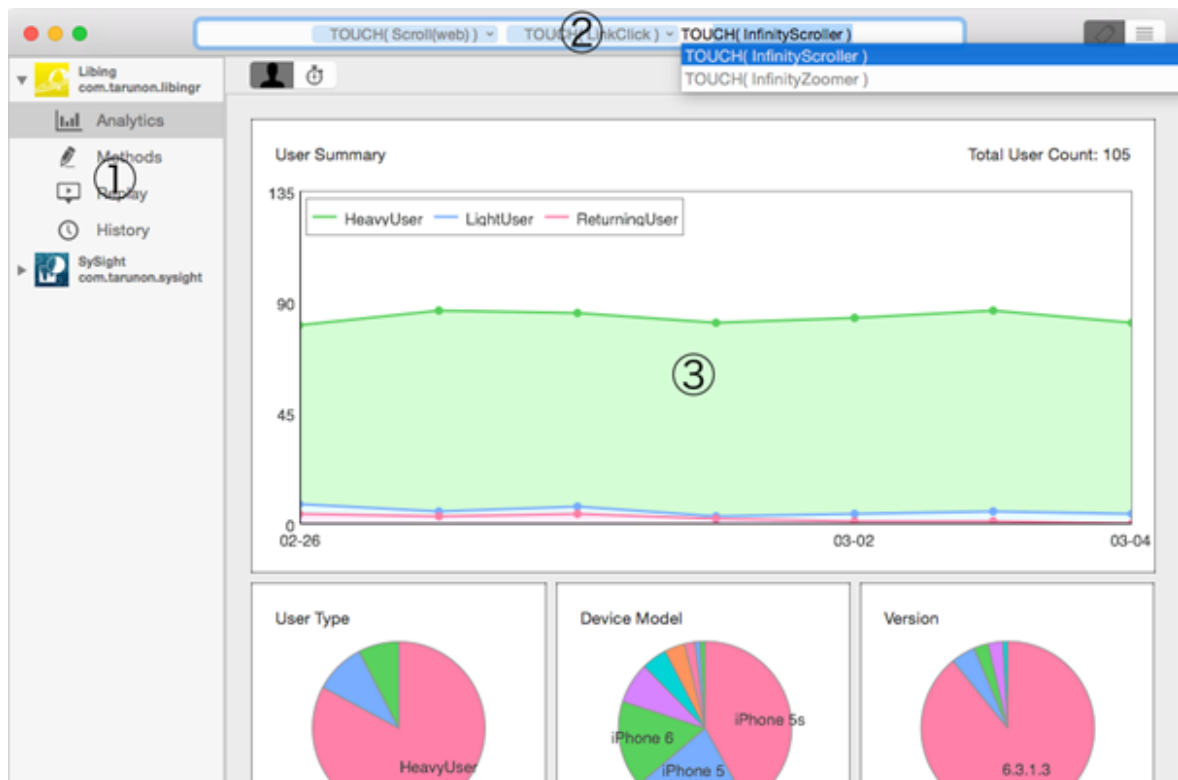


図 2 HeartMap クライアントの GUI

①はアプリケーションメニューである。HeartMap ライブラリを導入したアプリケーションが表示され、ここから各種情報にアクセスする。

②は検索フィールドである。ここに 3.3 で定義したクエリを入力することで、検索結果の情報が表示される。

③は統計情報である。ユーザのタイプ、利用している端末、アプリのバージョンのユーザ数が表示される。それぞれセッション数を表示することも可能である。操作のリプレイ画面を図 3 に示す。



図 3 操作のリプレイ画面

①はリプレイ画面である。ここに指の動きが再現される。また、セッションに含まれる指の動きが全て半透明で表示される。

②はジェスチャセルである。左から順に、タッチ毎の指の動きの概形、タッチの時刻、入力に要した時間の長さ、使用した UI の情報が表示される。クリックすることでその指の動きから再生を開始する。

リプレイ画面は時系列順の指の動きを再現する。

4. 従来の技術(または機能)との相違

表 1 に代表的な既存のツールとの比較を示す。

HeartMap は他のツールと比較してイベントの指定無しに操作の検索、抽出が可能であること、ライブラリの動作に必要なマシンリソースが低い事が利点として挙げられる。最も近いツールは Appsee であるが、Appsee はジェスチャ入力に対して有効なアプローチを有していない。HeartMap は図 3 から見て取れるように、セッションに含まれる全てのタッチ操作の概形を表示しており、ユーザの指の動きにより着目したツールとなっている。また、ジェスチャやその入力時間による検索も可能とした。

表 1 代表的な既存ツールとの比較

	ライブラリの動作に必要なマシンリソース	ライブラリによるイベントの指定	検索、抽出対象	ユーザの操作状況の観察
HeartMap (本プロジェクト)	低	無(必要に応じてメソッドを定義)	UI 操作によって呼び出されたメソッド	可能
Google Analytics	低	有(画面遷移も指定する必要がある)	定義した画面遷移とイベント	不可能
Flurry	低	有	定義したイベント	不可能
Appsee	高 (画面録画時)	無(ジェスチャには対応しない)	画面遷移とボタン、メニュー	可能(画面の録画が可能)
HEAP	低	有	定義したイベント	不可能
MixPanel	低	有	定義したイベント	不可能

5. 期待される効果

HeartMap を用いる事で、開発した UI を実際に利用するユーザを観察し、UI の評価や改善案の提案が可能となる。実際に開発中に利用することで得た効果について次に詳述する。縦方向にのみスクロールする画面で、横方向へのスワイプ操作をジェスチャとして追加した。しかし、追加したジェスチャを起点とする操作の目的の達成率が73%と高い数値ではなかった。これを元に、目的を達成していないユーザを検索したところ、次の条件で操作ミスを行うユーザを発見することができた。

1. 縦方向のスクロール操作が、斜め方向に強い入力であった(図 4)。

縦方向へのスクロールを行う際の入力は端末を保持する手の方へ弧を描くが、その傾向が極めて強いユーザが存在した。このユーザは、平均よりも操作の目的の達成率が低い傾向にあった。

2. エッジスワイプジェスチャとの混同が見られた(図 5)。

画面端からスワイプを行う別のジェスチャとの混同が見られた。連続してエッジスワイプジェスチャを利用した後に、追加したジェスチャを利用し、キャンセルしている。

これらの問題点に対して、斜め方向の入力を無視することで、ジェスチャを実行したユーザの目的の達成率を91%まで引き上げることが出来た。また、エッジスワイプジェスチャの判定の境目に、パディングを設けることでさらに改善する見込みがある。

HeartMap を用いることで、上に挙げた例のような、具体的な議論が可能となる



図 4 斜め方向へスクロール操作を行うユーザ

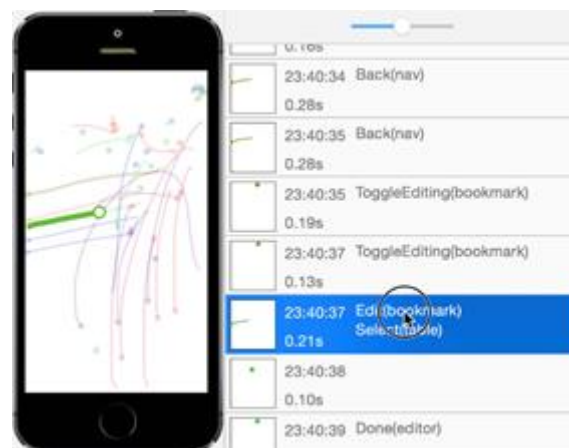


図 5 エッジスワイプジェスチャとの混同があるユーザ

6. 普及(または活用)の見通し

本プロジェクトで開発した HeartMap は未公開であるが、ターゲットは iOS 向けのアプリケーションの開発者であり、世界中で 30 万人弱存在する(2014 年現在)。国内では iOS のシェアは Android に勝っており、国内の開発者も相当数存在するものである。広く普及するため、公開を目指して継続的な活動を行いたい。

7. クリエータ名(所属)

齋藤 暢郎 (九州工業大学大学院)