**第３章　一時的UX取得システムの入出力インタフェ―スの検証**

本研究では一時的UXをリアルタイムで取得する「UX PLOT SYSTEM」というシステムを用いて研究を行う．このシステムは一時的UXである製品やサービスの利用中の体験をユーザ自身が逐次入力できる．入力されたデータはCSVや音声ファイルとして出力されるので，解析者はデータをもとに製品の評価を行うことが可能である．以下にUX PLOT SYSTEMの構成やインタフェースについて述べていく．

**3.1 一時的UX取得システム**

**本研究ではApple社のiPhone5s**[27]**を使用してUX入力アプリケーションの開発を行った．開発は**アップルの総合開発環境であるXcode[28]を使用した**．図3.1にアプリケーションのインタフェースを示す．アプアプリケーションは図3.1(a)の画面でUX値**，**図3.1(b)の画面で感情の変化の要因を入力することでUXの入力が完了する．ユーザは**製品やサービスの利用中にこの二つの画面を通して**逐次入力を行っていく．**

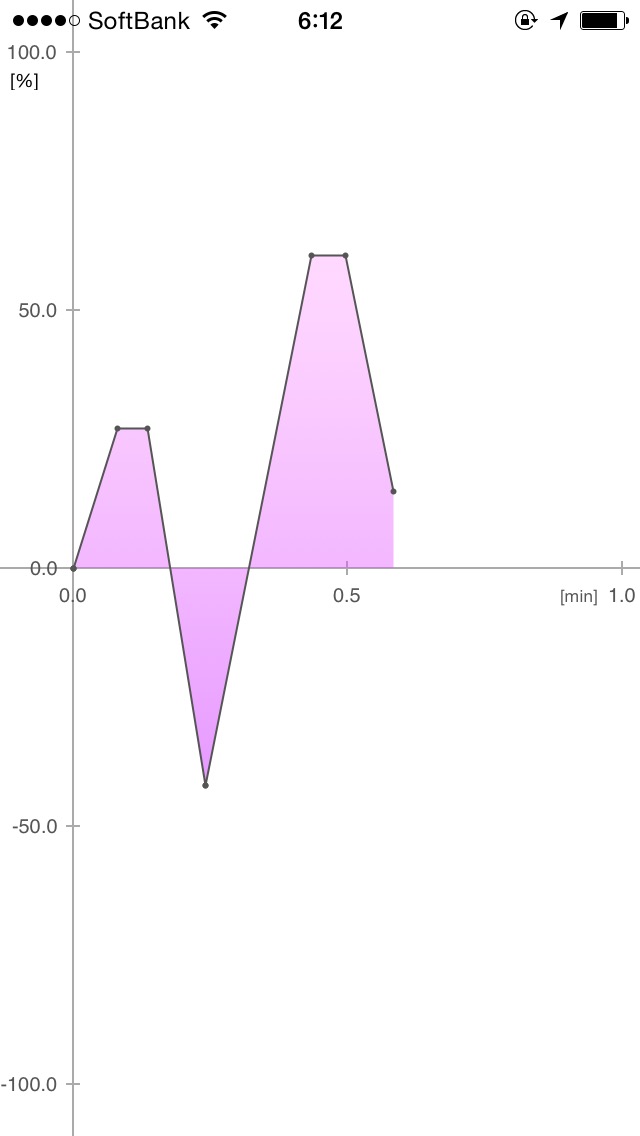


図3.1 (a). 一時的UX入力画面　　(b) 変化理由入力ポップアップ

**表3.1. デバイスとUXPLOTのスペック一覧**

|  |  |
| --- | --- |
| **入力デバイス** | Apple iPhone5s（iOS 7.0.4） |
| **開発環境** | Apple Xcode 5.0.2 |
| **開発言語** | Objective-C |
| **オープンソース** | Core Plot core-plot |

**・UX値入力画面**

**図3.1(a)の画面では**，**縦方向の**スワイプ操作によってUX値を100[%]〜-100[%]の値でUXの値を入力できる**．**スワイプ操作が終わると図3.1(b)の画面の変化理由画面のようにポップアップが出現する**．**

**・変化理由入力画面**

**UX値の入力を終えると図3.1(b)の画面のようなポップアップが現れる．ここではUXが変化した理由を要因と感情に分け，何のせいで（モノや現象などの要因）どのように感じたのか（感情や反応）の記入を行う．**

**3.2 入出力インタフェース比較実験**

**本実験では**，「UX PLOT SYSTEM」のインタフェースを評価し，インタフェースの違いによって一時的UXの取得にどんな影響をもたらすか検証する．スマートデバイスでは，GUI(Graphical User Interface)でアプリケーションを操作する．GUIは主に人間の意志を伝達する「入力インタフェース」と，処理結果をユーザに伝達する「出力インタフェース」に分けることができる．本実験では入力インタフェース，出力インタフェースの2つの観点からインタフェースの比較を行う．

**3.2.1 出力インタフェース比較実験のインタフェース**

UX入力後にユーザへのフィードバックとなるグラフが表示される場合とされない場合のインタフェースを開発した．これは**出力インタフェースの処理結果をユーザに示す**という観点に着目している．

　入力は縦方向のスワイプ操作によって行い，UX値の入力が完了したら図3.3のようなポップアップが出現し，UXの入力の確認がなされる．過去に入力したUX値は，グラフ有りの場合は図3.2(a)のようにユーザは確認できるのに対し，グラフ無しの場合では過去に入力されたUX値は確認できない．なお，入力の手順はグラフ有り・無しの両者で同様である．一般にユーザインタフェースでは,タスクの種類に応じた適切な時間内に,レスポンスが返されることが期待され,単純な入力においては1秒程度以下のシステム応答時間が好ましいとされている[29]．よってUX値の入力が完了し,ポップアップが表示するまでの時間も2秒から1秒に変更した．

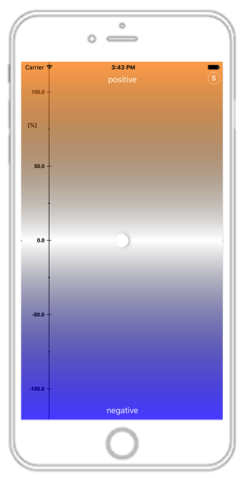
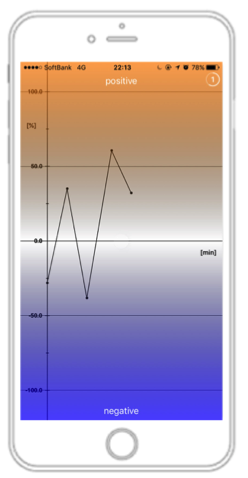


図3.2 (a). UX入力画面（グラフあり）　　(b) UX入力画面（グラフなし）

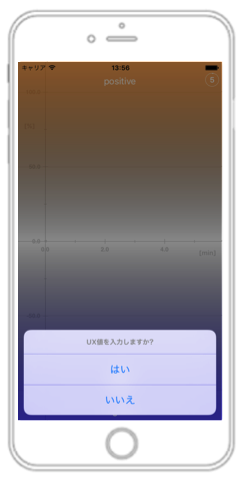


図3.3 (a). UXの入力確認ポップアップ

**表3.2. デバイスと出力インタフェースのスペック一覧**

|  |  |
| --- | --- |
| **入力デバイス** | Apple iPhone5s（iOS 8.4） |
| **開発環境** | Apple Xcode 6.4 |
| **開発言語** | Objective-C  Swift |
| **オープンソース** | core-plot  UAProgressView |

**3.2.2 入力インタフェース比較実験のインタフェース**

**ユーザのUXの情報である感情と感情の変化の要因入力に着目し**，**感情と感情の変化の要因の入力を**テキストで入力するインタフェースと音声で入力するインタフェースを開発した．前節で述べたiOSヒューマンインタフェースガイドラインではユーザの入力の手間は最小限にすることが推奨されているため，インタラクションを少なくするという観点で，比較するインタフェースは音声入力を採用した．

　UX値を入力後の感情と感情の要因入力画面において，テキスト入力の場合は図3.4(a)のようにキーボードが出現しテキストで入力を行う．一方音声入力の場合は，図3.4(b)の画面で音声入力を行い，中央の音声マークをタップすることで入力が完了する．

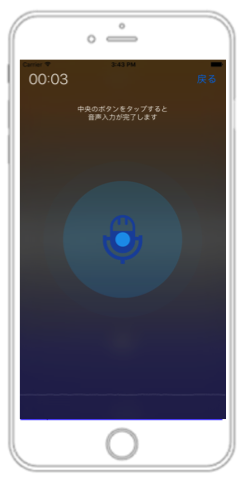
****

図3.4 (a).要因理由入力画面（テキスト） b) 要因理由入力画面（音声）

**表3.3. デバイスと入力インタフェースのスペック一覧**

|  |  |
| --- | --- |
| **入力デバイス** | Apple iPhone5s（iOS 8.4） |
| **開発環境** | Apple Xcode 7.3.1 |
| **開発言語** | Objective-C  Swift |
| **オープンソース** | core-plot  UAProgressView  Pulsator  EZAudio |

**3.2.3 音声解析**

**音声ファイルをテキスト出力し解析を行う為**，音声解析ツールを開発した．音声解析ツールはAppleが提供しているSFSpeechRecognizer[30]のAPIを使用している．

**表3.3. 音声解析ツールスペック一覧**

|  |  |
| --- | --- |
| **入力デバイス** | Apple iPhone6s（iOS 10.1） |
| **開発環境** | Apple Xcode 8.1 |
| **開発言語** | Swift |

**[27]** Apple Inc , iPhone5s – 技術仕様, **https://support.apple.com/kb/SP685?locale=ja\_JP&viewlocale=ja\_JP**, 2016/12/7アクセス.

**[28]** Apple Inc , Xcode , **https://developer.apple.com/support/xcode/jp/**, 2016/12/7アクセス.

**[29]** Butler, T.W, Computer response time and user performance, Proc. Human Factors in Computer Systems (CHI ’83 ), pp.56–62 (1983).

**[30]** Apple Inc , API Reference SFSpeechRecognizer , **https://developer.apple.com/reference/speech/sfspeechrecognizer**, 2016/12/7アクセス.