

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2012-164336
(P2012-164336A)

(43) 公開日 平成24年8月30日(2012.8.30)

(51) Int.Cl.	F I	テーマコード (参考)
G06F 3/041 (2006.01)	G06F 3/041 330C	5B068
G06F 3/048 (2006.01)	G06F 3/041 380C	5B087
	G06F 3/041 380D	5E555
	G06F 3/041 330P	
	G06F 3/048 620	

審査請求 有 請求項の数 27 O L 外国語出願 (全 28 頁)

(21) 出願番号 特願2012-88747 (P2012-88747)
 (22) 出願日 平成24年4月9日(2012.4.9)
 (62) 分割の表示 特願2010-502358 (P2010-502358) の分割
 原出願日 平成21年3月3日(2009.3.3)
 (31) 優先権主張番号 12/042,299
 (32) 優先日 平成20年3月4日(2008.3.4)
 (33) 優先権主張国 米国 (US)

(特許庁注：以下のものは登録商標)

1. J A V A

(71) 出願人 503260918
 アップル インコーポレイテッド
 アメリカ合衆国 95014 カリフォルニア州 クパチーノ インフィニット ループ 1
 (74) 代理人 100076428
 弁理士 大塚 康德
 (74) 代理人 100112508
 弁理士 高柳 司郎
 (74) 代理人 100115071
 弁理士 大塚 康弘
 (74) 代理人 100116894
 弁理士 木村 秀二
 (74) 代理人 100130409
 弁理士 下山 治

最終頁に続く

(54) 【発明の名称】 タッチイベントモデルプログラミングインターフェイス

(57) 【要約】

【課題】タッチイベントモデルプログラミングインターフェイスを提供する。

【解決手段】タッチ感知装置から1つ以上のタッチ入力信号を得ることができる。タッチイベントモデルを使用し、それらタッチ入力信号に基づいてタッチ及び/又はジェスチャーイベントを決定することができる。このタッチ及びジェスチャーイベントは、タッチ感知装置上に表示されるウェブページの異なる領域から発生されるタッチ入力信号に関連付けることができる。プログラミングインターフェイスを通して少なくとも1つのタッチ又はジェスチャーイベントへのアクセスを与えることができる。

【選択図】 図3

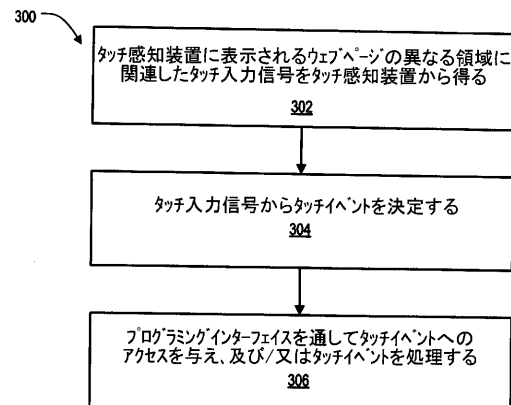


FIG. 3

【特許請求の範囲】**【請求項 1】**

ウェブブラウザにおいて、
ジェスチャーイベントに関連した回転値を受け取るステップと、
前記回転値に基づいてウェブブラウザにおけるジェスチャーイベントに関連したウェブページの要素を動的に回転するステップであって、前記回転値が度での相対的デルタであるステップと、
を備えた方法。

【請求項 2】

前記ジェスチャーイベントは、2つ以上のタッチイベントを含む、請求項 1 に記載の方法。

10

【請求項 3】

ウェブブラウザにおいて、
ジェスチャーイベントに関連したスケール値を受け取るステップと、
前記スケール値に基づいてウェブブラウザにおけるジェスチャーイベントに関連したウェブページの要素を動的にサイズ変更するステップであって、前記スケール値がドキュメントピクセルでの相対的デルタであるようなステップと、
を備えた方法。

【請求項 4】

前記ジェスチャーイベントは、2つ以上のタッチイベントに関連付けられる、請求項 3 に記載の方法。

20

【請求項 5】

ウェブブラウザにおいて、タッチリストを受け取るステップを備え、このタッチリストは、ウェブページ上の1つ以上のタッチを識別するためのタッチイベントデータを含み、このタッチイベントデータは、タッチ識別子と、少なくとも1セットのタッチ位置座標とを含み、前記タッチリストは、各タッチに関連したタッチイベントターゲットに対するタッチイベントデータを更に含み、前記少なくとも1セットのタッチ位置座標は、クライアント座標、ページ座標、及びスクリーン座標の少なくとも1つを含む、方法。

【請求項 6】

前記タッチイベントデータは、1つ以上の変更されたタッチを識別する、請求項 5 に記載の方法。

30

【請求項 7】

プロセッサにより実行されたときに、そのプロセッサが、
ウェブページに関連したジェスチャーイベントに関連した回転値を受け取り、
前記回転値に基づいてウェブブラウザにおけるジェスチャーイベントに関連したウェブページの要素を動的に回転し、前記回転値が度での相対的デルタである、
ことを含むオペレーションを遂行するようにさせるインストラクションが記憶されたコンピュータ読み取り可能な媒体。

【請求項 8】

前記ジェスチャーイベントは、2つ以上のタッチイベントを含む、請求項 1 に記載のコンピュータ読み取り可能な媒体。

40

【請求項 9】

プロセッサにより実行されたときに、そのプロセッサが、
ウェブページに関連したジェスチャーイベントに関連したスケール値を受け取り、
前記スケール値に基づいてウェブブラウザにおけるジェスチャーイベントに関連したウェブページの要素を動的にサイズ変更し、前記スケール値がドキュメントピクセルでの相対的デルタである、
ことを含むオペレーションを遂行するようにさせるインストラクションが記憶されたコンピュータ読み取り可能な媒体。

【請求項 10】

50

前記ジェスチャーイベントは、2つ以上のタッチイベントに関連付けられる、請求項3に記載のコンピュータ読み取り可能な媒体。

【請求項11】

プロセッサにより実行されたときに、そのプロセッサが、
タッチリストを受け取る、

ことを含むオペレーションを遂行するようにさせるインストラクションが記憶され、前記タッチリストは、ウェブページ上の1つ以上のタッチを識別するためのタッチイベントデータを含み、このタッチイベントデータは、タッチ識別子と、少なくとも1セットのタッチ位置座標とを含み、前記タッチリストは、各タッチに関連したタッチイベントターゲットに対するタッチイベントデータを更に含み、前記少なくとも1セットのタッチ位置座標は、クライアント座標、ページ座標、及びスクリーン座標の少なくとも1つを含む、コンピュータ読み取り可能な媒体。

10

【請求項12】

前記タッチイベントデータは、1つ以上の変更されたタッチを識別する、請求項11に記載のコンピュータ読み取り可能な媒体。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、一般的に、ウェブブラウジングサービスに係る。

【背景技術】

20

【0002】

ドキュメント内のテキストベース情報の構造を記述すると共に、そのテキストを双方向フォーム、埋め込まれた映像及び他のオブジェクトで補足するための手段をなすマークアップ言語を使用して、ウェブページが生成される。1つの普及したマークアップ言語は、カギ括弧で囲まれたタグの形態で書かれたハイパーテキストマークアップ言語（HTML）である。HTMLは、ウェブページの見掛け及び意味を記述することができると共に、ウェブブラウザ及び他のHTMLプロセッサの振舞いに影響する埋め込み型スクリプト言語コード（例えば、JavaScript（登録商標））を含むことができる。JavaScript（登録商標）は、マウスイベントハンドラー又はイベントリスナーをウェブページに追加する能力を開発者に与える。これらのマウスイベントハンドラーは、ウェブページの特定の領域に指定され、そしてマウスアップ又はマウスダウンイベントのようなマウスイベントをこれらの領域において受け取るように構成することができる。

30

【発明の概要】

【0003】

対照的に、タッチ感知装置でナビゲートされるウェブページは、ユーザが1つ以上の指でウェブページにタッチしそしてジェスチャーをなすことで発生されたタッチイベントに応答することがしばしば必要である。従来のマウスイベントハンドラーは、これらタッチイベントを正しく解釈することができない。従って、タッチイベントは、タッチイベントを正しく解釈すると共に、タッチ感知ディスプレイ又は装置の能力を開発者が完全に利用できるようにするための異なるタッチイベントモデルを必要としている。

40

【0004】

タッチ感知装置から1つ以上のタッチ入力信号を得ることができる。タッチイベントモデルを使用し、それらタッチ入力信号に基づいてタッチ及び/又はジェスチャーイベントを決定することができる。このタッチ及びジェスチャーイベントは、タッチ感知装置上に表示されるウェブページの異なる領域から発生されるタッチ入力信号に関連付けることができる。プログラミングインターフェイスを通して少なくとも1つのタッチ又はジェスチャーイベントへのアクセスを与えることができる。

【0005】

ある実施形態では、ウェブブラウザにおいて、方法は、ジェスチャーイベントに関連した回転値を受け取るステップと、その回転値に基づいてウェブブラウザにおけるジェスチ

50

ャーイベントに関連したウェブページのエレメントを動的に回転するステップであって、前記回転値が、度での相対的デルタであるようなステップとを備えている。ジェスチャーイベントは、2つ以上のタッチイベントを含むことができる。

【0006】

ある実施形態では、ウェブブラウザにおいて、方法は、ジェスチャーイベントに関連したスケーリング値を受け取るステップと、そのスケーリング値に基づいてウェブブラウザにおけるジェスチャーイベントに関連したウェブページのエレメントを動的にサイズ変更するステップであって、前記スケーリング値が、ドキュメントピクセルでの相対的デルタであるようなステップとを備えている。ジェスチャーイベントは、2つ以上のタッチイベントに関連付けることができる。

10

【0007】

ある実施形態では、ウェブブラウザにおいて、方法は、タッチリストを受け取るステップを備え、このタッチリストは、ウェブページにおける1つ以上のタッチを識別するためのタッチデータを含み、このタッチデータは、タッチ識別子と、少なくとも1セットのタッチ位置座標とを含み、更に、タッチリストは、各タッチに関連したタッチイベントターゲットに対するデータを含み、前記少なくとも1セットのタッチ位置座標は、クライアント座標のセットと、ページ座標のセットと、スクリーン座標のセットとを含む。タッチデータは、1つ以上の変更されたタッチを識別することができる。

【0008】

システム、方法及びコンピュータ読み取り可能な媒体に向けられた他の実施形態も開示される。

20

【図面の簡単な説明】

【0009】

【図1A】、

【図1B】例示的なウェブページドキュメントを示す。

【図2】例示的なマルチタッチケーパブル装置の処理スタックを示す。

【図3】タッチイベントを処理するための例示的プロセスのフローチャートである。

【図4】例示的なマルチタッチケーパブル装置を示す。

【図5】図4のマルチタッチケーパブル装置に対する例示的なネットワークオペレーティング環境のブロック図である。

30

【図6】図4のマルチタッチケーパブル装置の実施例のブロック図である。

【発明を実施するための形態】

【0010】

例示的ウェブページ構造及びDOM

図1Aは、ブラウザに表示できる例示的ウェブページ100を示す。ブラウザは、図4のマルチタッチケーパブル(multi-touch capable)装置400のようなポータブル装置にホストされてもよい。ウェブページ100には、1つ以上のエレメント、即ちエレメント102(“エレメント1”)、エレメント104(“エレメント2”)及びエレメント106(“エレメント3”)を表示することができる。これらエレメント102、104、106は、ユーザが選択できるウェブページ100の領域に対応することができ、選択の結果として付加的な機能を与えることができる。例えば、これらエレメントは、ウェブページ100のボタンに対応する。又、1つのエレメントが別のエレメントを含むようにエレメントをネスト構成にすることもできる。例えば、エレメント104は、エレメント108を含む。図示された例では、エレメント108は、例えば、メディアプレーヤーユーザインターフェイスであるエレメント104内にネストされたスクラバーコントロールである。

40

【0011】

ある実施形態において、ユーザは、マウスではなく指を使用してウェブページ100のエレメントで種々のファンクションを遂行することができる。ユーザは、例えば、図4に示すタッチ感知ディスプレイ402を使用してウェブページ100のエレメントにタッチ

50

することができる。一実施例では、ユーザは、1つ以上の指でエレメントにタッチし及び/又はスワイプ、ピンチ又は回転運動のようなジェスチャーを行うことで、エレメントを選択することができる。タッチ入力信号を確認するために、ウェブページ100のあるエリアをタッチイベントハンドラーに関連付けることができる。これは、図1Bを参照して述べるように、DOM及び埋め込まれたスクリプト言語で行うことができる。

【0012】

図1Bは、ウェブページ100に関連付けられた例示的DOM150である。このDOM150は、ウェブページ100の構造的表示を与えるもので、スクリプト言語(例えば、JavaScript(登録商標))で解釈できるオブジェクトのセットとしてウェブページコンテンツを記述する。ある実施形態において、DOM150は、ウェブページ100のエレメント102、104、106、108をツリーの個々のノードにマップすることによりウェブページの構造へのアクセスを与える。例えば、エレメント102は、ノード154に対応する。エレメント104は、ノード156に対応する。エレメント106は、ノード158に対応する。エレメント108は、ノード160に対応する。根ノード152は、全ウェブページ100に対応する。

10

【0013】

ある実施形態において、ウェブページ100の1つ以上のエレメント102、104、106、108は、DOM150の対応ノードをタッチイベントハンドラーに関連付けることにより、1つ以上の対応するタッチイベントハンドラーに関連付けることができる。タッチイベントハンドラーは、ウェブページ100のHTMLタグに挿入できると共に、例えば、ユーザがウェブページ100の1つのエレメントにタッチするか又はその中でジェスチャーするとき、スクリプト言語を実行してアクションを遂行することができる。例えば、JavaScript(登録商標)は、DOM150と共に機能して、異なるタッチイベントにアクションをアタッチすることができる。

20

【0014】

ある実施形態では、1つ以上のエレメント102、104、106、108が、イベントハンドラー又はリスナーにより検出されたタッチ入力を受け取ることができる。タッチ入力は、図2を参照して説明するように、ソフトウェアスタックの1つ以上のレイヤにおいて実施できるタッチイベントモデルによって検出されてタッチイベントへと処理することができる。タッチイベントは、ウェブページ100により更に処理することができる。タッチイベントは、タッチ感知装置により発生される生のタッチ入力信号よりもアプリケーションにおいて使用し易いフォーマット(例えば、属性)である。例えば、各タッチイベントは、タッチが現在発生しているところの座標のセットを含むことができる。

30

【0015】

ウェブページ100における各エレメント、及びその関連イベントハンドラーは、タッチイベントを受け取り、処理し、取り扱うことができる。例えば、ドライバ202(図2)が、エレメント102に関連したタッチポイント110を感知するか、又はエレメント104に関連したタッチポイント112を感知する場合に、エレメント102又は104に関連したイベントハンドラーは、エレメントにタッチしたことを指示する個別のタッチイベントを各々受け取ることができ、そして任意であるが、そのタッチイベントを更なる処理のためにウェブページ100へ送ることができる。ある実施形態において、イベントハンドラーに対応しないウェブページ100の領域にタッチした場合には、アプリケーションレイヤ214のブラウザが、ウェブページ100ではなく、入力を処理することができる。

40

【0016】

ある実施形態では、DOM150においてノードごとに指ごとにタッチイベントを検出することができる。例えば、ユーザは、実質的に同時にタッチポイント110及びタッチポイント112においてタッチ感知ディスプレイ402にタッチすることができ、そしてタッチイベントモデルにより2つの個別のタッチイベントを検出することができる。DOM150における各ノード102及びノード104は個別のタッチイベントハンドラーに

50

関連しているので、タッチポイント110及びタッチポイント112に対して個別のタッチイベントを検出することができる。

【0017】

ある実施形態において、タッチイベントは、EventTargetとしてウェブページ100へ配送することができる。タッチイベントの幾つかの例は、touchstart（タッチ開始）、touchmove（タッチ移動）、touchend（タッチ終了）、及びtouchcancel（タッチキャンセル）を含むことができる。他のタッチイベントも考えられる。touchstartは、イベントハンドラーに関連したウェブページ100の領域においてタッチ感知ディスプレイ402にユーザが最初に指を置いたときに検出されるタッチイベントである。ユーザがウェブページ100上で自分の指を動き回らせると、1つ以上のtouchmoveイベントを検出することができる。ユーザがウェブページ100から自分の指を持ち上げると、touchendイベントが検出される。システムが普通のイベントハンドリングを中断するときには、touchcancelを検出することができる。例えば、偶発的なタッチを防止するためにタッチ感知ディスプレイ402がロックされたときにtouchcancelイベントが生じる。

10

【0018】

又、ある実施形態において、2つ以上のタッチイベントを合成することによりジェスチャーイベントを検出することもできる。タッチイベントと同様に、ジェスチャーイベント（GestureEvent）を、EventTargetとしてウェブページ100へ配送することもできる。ジェスチャーイベントの幾つかの例は、gesturestart（ジェスチャー開始）、gesturechange（ジェスチャー変更）、及びgestureend（ジェスチャー終了）である。ジェスチャーイベントは、スケール及び/又は回転情報を含むことができる。回転情報は、度での相対的デルタである回転値を含むことができる。ウェブページ100の要素は、この回転値に基づいて動的に回転することができる。スケール情報は、ドキュメントピクセルでの相対的デルタであるスケール値を含むことができる。ジェスチャーイベントに関連したウェブページ100の要素は、このスケール値に基づいて動的にサイズ変更することができる。他のジェスチャーイベントも考えられる。

20

【0019】

ある実施形態では、ウェブページ100上の1つ以上のタッチを識別するためのタッチイベントデータを含むタッチリストを受け取ることができる。タッチイベントデータは、タッチ識別子と、少なくとも1セットのタッチ位置座標とを含むことができる。又、タッチリストは、各タッチに関連したタッチイベントターゲットに対するタッチイベントデータも含むことができる。ある実施形態では、1セットのタッチ位置座標は、クライアント座標、ページ座標、及びスクリーン座標を含むことができる。ある実施形態では、タッチイベントデータは、1つ以上の変更されたタッチを識別することができる。

30

【0020】

ある実施形態では、GestureEventを、TouchEventの前にウェブページ100へ送り出すことができる。例えば、ユーザがタッチポイント110及びタッチポイント112に指を置いて、タッチ感知ディスプレイ上で指を時計方向又は反時計方向に動かして回転ジェスチャーを行う場合には、タッチイベントモデルがこれらの複数タッチイベントを検出し、そのタッチイベントをジェスチャーイベントへと合成する。次いで、このジェスチャーイベントをウェブページ100へ送り、その後、そのジェスチャーイベントを形成するために合成されたタッチイベントを送ることができる。このように、開発者は、ジェスチャーイベント、及びジェスチャーイベントの個々のタッチイベントにアクセスすることができ、これは、ウェブアプリケーションを開発するときに、より大きな融通性を開発者に与えるものである。

40

【0021】

ある実施形態では、タッチイベントが次の順序で受け取られる。即ち、touchstartイベント、1つ以上のtouchmoveイベント、そしてtouchend又は

50

touchcancel イベント。図 1 A の実施例を使用すると、ユーザがタッチポイント 110 にタッチしたときに、エレメント 102 に関連した第 1 のタッチイベントハンドラーにより第 1 の touchstart イベントが検出される。ユーザがタッチポイント 112 にタッチすると、エレメント 104 に関連した第 2 のタッチイベントハンドラーにより第 2 の touchstart イベントが検出される。ユーザが自分の指を持ち上げずに自分の指を回転するにつれて、第 1 及び第 2 のタッチイベントハンドラーが touchmove イベントを検出し、これは、タッチイベントモデルにより回転ジェスチャーイベントとして解釈することができる。ユーザが回転を終了して自分の指をウェブページ 100 から持ち上げると、第 1 及び第 2 のタッチイベントハンドラーが touchend イベントを検出する。これらのタッチイベントの全部又は幾つかを、タッチイベント「アプリケーションプログラミングインターフェイス (API)」により、開発者に利用できるようにすることができる。タッチ API は、「ソフトウェア開発キット (SDK)」として、又はアプリケーションの一部として (例えば、ブラウザツールキットの一部として) 開発者に利用できるようにすることができる。タッチイベント API は、その種々のファンクションを遂行するために、他のサービス、フレームワーク及びオペレーティングシステムに依存することができる。これらのサービス、フレームワーク及びオペレーティングシステムは、図 2 を参照して述べるようにソフトウェア又は処理スタックの一部でよく、ここでは、アプリケーションにおけるイベントアクションを定義するためにドキュメントに挿入できる属性にタッチイベントが関連付けされる。

10

20

【0022】

例示的 IDL

例示的なタッチイベントモデルを、「インターフェイス記述言語 (IDL)」で以下に説明する。IDL のファンクション及びデータ構造は、ウェブ設計者又はアプリケーション開発者により API を通してアクセスすることができる。タッチイベント及び / 又はジェスチャーイベントへのアクセスは、アプリケーションにおけるイベントアクションを定義するためにマークアップ言語ドキュメント (例えば、HTML、XML) に挿入できる属性に関連付けることができる。例えば、これら属性は、タッチ感知ディスプレイ 402 に表示されるウェブページを発生するために HTML ドキュメントの 1 つ以上の HTML タグに挿入することができる。イベントアクションは、植え込み型スクリプト (例えば、JavaScript (登録商標)) を実行することを含むことができる。

30

```

interface {
    Conditional=TOUCH_EVENTS,
    GenerateConstructor
} TouchEvent : UIEvent {

    void initTouchEvent(in AtomicString type,
        in boolean canBubble,
        in boolean cancelable,
        in DOMWindow view,
        in long detail,
        in long screenX,
        in long screenY,
        in long clientX,
        in long clientY,
        in boolean ctrlKey,
        in boolean altKey,
        in boolean shiftKey,
        in boolean metaKey,
        in TouchList touches,
        in TouchList targetTouches,
        in TouchList changedTouches,
        in long scale,
        in long rotation);

    readonly attribute TouchList touches; // all touches
    readonly attribute TouchList targetTouches; // all touches in this TouchEvent Target
    readonly attribute TouchList changedTouches; // all touches changed in the current
event

    readonly attribute long scale;
    readonly attribute long rotation;
    readonly attribute boolean ctrlKey;
    readonly attribute boolean shiftKey;
    readonly attribute boolean altKey;
    readonly attribute boolean metaKey;
};

interface [
    Conditional=TOUCH_EVENTS,
] Touch {
    readonly attribute EventTarget target;

    readonly attribute long identifier;

    readonly attribute long clientX;
    readonly attribute long clientY;
    readonly attribute long pageX;
    readonly attribute long pageY;
    readonly attribute long screenX;
    readonly attribute long screenY;
};

interface [
    Conditional=TOUCH_EVENTS,
    HasIndexGetter,
] TouchList {
    readonly attribute unsigned long length;

```

```

    Touch    item(in unsigned long index);
};

interface [
    Conditional=TOUCH_EVENTS,
    GenerateConstructor
] GestureEvent : UIEvent {
    void initGestureEvent( in AtomicString type,
        in boolean canBubble,
        in boolean cancelable,
        in DOMWindow view,
        in long detail,
        in long screenX,
        in long screenY,
        in long clientX,
        in long clientY,
        in boolean ctrlKey,
        in boolean altKey,
        in boolean shiftKey,
        in boolean metaKey,
        in EventTarget target,
        in long scale,
        in long rotation);

    readonly attribute EventTarget target;

    readonly attribute long scale;
    readonly attribute long rotation;

    readonly attribute boolean ctrlKey;
    readonly attribute boolean shiftKey;
    readonly attribute boolean altKey;
    readonly attribute boolean metaKey;
};

In Document.idl:

Touch    createTouch(in EventTarget target,
        in long identifier,
        in long clientX,
        in long clientY,
        in long pageX,
        in long pageY,
        in long screenX,
        in long screenY)
    raises (DOMException);
[Custom] TouchList    createTouchList()
    raises (DOMException);

```

【 0 0 2 3 】

次は、上述した例示的IDLを使用してタッチイベントを処理するためのHTMLコード断片の例である。以下のHTMLは、例えば、タッチイベントリスナーTouchStart及びGestureStartが、HTMLコードをもつエレメントに追加されたことを示す。

```

this.element.addEventListener('touchstart',    function(e)    {    return
self.onTouchStart(e) }, false);
this.element.addEventListener('gesturestart', function(e)    {    return
self.onGestureStart(e) }, false);

```

【 0 0 2 4 】

前記 I D L に対応する H T M L コードは、次のものでよい。

```

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

```

10

```

<html lang="en">

```

```

  <head>

```

```

    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">

```

```

    <meta name="viewport" content="initial-scale=1.0" />

```

```

    <title>Transform Gestures</title>

```

```

    <style type="text/css" media="screen">

```

```

      .box {

```

```

        position: absolute;

```

```

        height: 150px;

```

```

        width: 150px;

```

```

        background-color: blue;

```

```

      }

```

20

```

      .box:active {

```

```

        background-color: red;

```

```

      }

```

```

      body {

```

```

        margin: 0px;

```

```

      }

```

```

      #container {

```

```

        position: absolute;

```

```

        width: 100%;

```

```

        height: 100%;

```

```

      }

```

30

```

      #main-box2 {

```

```

        top: 10px;

```

```

        left: 155px;

```

```

        background: red;

```

```

        z-index: 1;

```

```

      }

```

40

```

</style>
<script type="text/javascript" charset="utf-8">

    var trackedObjectCount = 0;

    function Box(inElement)
    {
        var self = this;

        this.element = inElement;

        this.scale = 1.0;
        this.rotation = 0;
        this.position = '0,0';

        this.element.addEventListener('touchstart', function(e) { return
self.onTouchStart(e) }, false);
        this.element.addEventListener('gesturestart', function(e) { return
self.onGestureStart(e) }, false);
    }

    Box.prototype = {
        // position strings are "x,y" with no units
        get position()
        {
            return this._position;
        },

        set position(pos)
        {
            this._position = pos;

            var components = pos.split(',')
            var x = components[0];
            var y = components[1];

            const kUseTransform = true;
            if (kUseTransform) {
                this.element.style.webkitTransform = 'rotate(' + this.rotation + 'deg) scale(' +
this.scale + ') translate(' + x + 'px, ' + y + 'px);
            }
            else {
                this.element.style.left = x + 'px';
                this.element.style.top = y + 'px';
            }
        },

        get x()
        {
            return parseInt(this._position.split(',')[0]);
        },
    }

```

```

set x(inX)
{
  var comps = this._position.split(',');
  comps[0] = inX;
  this.position = comps.join(',');
},

get y()
{
  return parseInt(this._position.split(',')[1]);
},
10

set y(inY)
{
  var comps = this._position.split(',');
  comps[1] = inY;
  this.position = comps.join(',');
},

filterEvent: function(e)
{
  // Prevent the browser from doing its default thing (scroll, zoom)
  e.preventDefault();
20

  // Event listeners are added at the document level, so we receive gesturechange
events for other elements.
  return (e.target == this.element);
},

onTouchStart: function(e)
{
  if (!this.filterEvent(e))
    return false;

  // Start tracking when the first finger comes down in this element
  if (e.targetTouches.length != 1)
    return false;
30

  this.startX = e.targetTouches[0].clientX;
  this.startY = e.targetTouches[0].clientY;

  var self = this;
  if (!("touchMoveHandler" in this)) {
    this.touchMoveHandler = function(e) { return self.onTouchMove(e) }
    this.touchEndHandler = function(e) { return self.onTouchEnd(e) }
  }
40

  document.addEventListener('touchmove', this.touchMoveHandler, false);
  document.addEventListener('touchend', this.touchEndHandler, false);

  trackedObjectCount++;

  return false;

```

```

},

onTouchMove: function(e)
{
  if (!this.filterEvent(e))
    return false;

  // Don't track motion when multiple touches are down in this element (that's a
gesture)
  if (e.targetTouches.length != 1)
    return false;

  var leftDelta = e.targetTouches[0].clientX - this.startX;
  var topDelta = e.targetTouches[0].clientY - this.startY;

  var newLeft = (this.x) + leftDelta;
  var newTop = (this.y) + topDelta;

  this.position = newLeft + ',' + newTop;

  this.startX = e.targetTouches[0].clientX;
  this.startY = e.targetTouches[0].clientY;

  return false;
},

onTouchEnd: function(e)
{
  if (!this.filterEvent(e))
    return false;

  // Stop tracking when the last finger is removed from this element
  if (e.targetTouches.length > 0)
    return false;

  document.removeEventListener('touchmove', this.touchMoveHandler, false);
  document.removeEventListener('touchend', this.touchEndHandler, false);

  trackedObjectCount--;

  return false;
},

onGestureStart: function(e)
{
  if (!this.filterEvent(e))
    return false;

  var self = this;
  if (!("gestureChangeHandler" in this)) {
    this.gestureChangeHandler = function(e) { return self.onGestureChange(e) }
    this.gestureEndHandler = function(e) { return self.onGestureEnd(e) }
  }
}

```

10

20

30

40

```

document.addEventListener('gesturechange', this.gestureChangeHandler, true);
document.addEventListener('gestureend', this.gestureEndHandler, true);

return false;
},

onGestureChange: function(e)
{
  if (!this.filterEvent(e))
    return false;

  // Only interpret gestures when tracking one object. Otherwise, interpret raw
touch events
  // to move the tracked objects.
  if (trackedObjectCount == 1) {
    this.scale += e.scaling * 0.01;
    this.rotation += e.rotation / 2;
    this.position = this.position;
  }

  return false;
},

onGestureEnd: function(e)
{
  if (!this.filterEvent(e))
    return false;

  document.removeEventListener('gesturechange', this.gestureChangeHandler,
true);
  document.removeEventListener('gestureend', this.gestureEndHandler, true);

  return false;
},
}

function loaded()
{
  new Box(document.getElementById('main-box'));
  new Box(document.getElementById('main-box2'));
}
window.addEventListener('load', loaded, true);
</script>
</head>
<body>
  <div id="container">
    <div id="main-box" class="box"></div>
    <div id="main-box2" class="box"></div>
  </div>
</body>
</html>

```

【 0 0 2 5 】

マルチタッチ装置のための例示的処理スタック

図 2 は、例示的なマルチタッチケーパブル装置の処理スタックを示す図である。上述したタッチイベントモデルは、処理スタックの 1 つ以上の領域及びスタックにおけるユーザの種々のリソースにおいて実施することができる。ハードウェア 2 0 0 のレイヤは、種々

のハードウェアインターフェイスコンポーネント、例えば、タッチ感知又はイネーブル型装置或いはタッチ感知ディスプレイを含むことができる。タッチ感知装置は、ディスプレイと、複数のタッチを同時に感知するパネルとを含むことができる。又、ハードウェアレイヤ200は、タッチ感知ディスプレイ又は装置の向き（例えば、肖像、風景）を検出するための加速度計も含むことができる。従って、向きを表す信号は、最適な表示のためにウェブページをスケーリングするようにタッチイベントモデルにより使用することができる。

【0026】

ドライバレイヤ202における1つ以上のドライバは、ハードウェア200と通信することができる。例えば、ドライバは、ハードウェアレイヤ200のタッチ感知ディスプレイ又は装置によって発生されるタッチ入力信号を受け取って処理することができる。コアオペレーティングシステム(OS)204は、ドライバ(1つ又は複数)と通信することができる。コアOS204は、ドライバ(1つ又は複数)から受け取られた生の入力データを処理することができる。ある実施形態では、ドライバは、コアOS204の一部であると考えることもできる。

10

【0027】

OSアプリケーションプログラミングインターフェイス(API)206のセットがコアOS204と通信することができる。これらのAPIは、オペレーティングシステムと共に通常含まれるAPIのセットでよい(例えば、Linux(登録商標)又はUNIX(登録商標)API)。

20

コアファウンデーションAPI208のセットは、OS API206を使用することができ、そしてファウンデーションAPI210のセットは、コアファウンデーションAPI208を使用することができる。

【0028】

ウェブページソフトウェア開発キット(SDK)210は、装置で実行されるアプリケーションにより使用するように設計されたAPIのセットを含むことができる。タッチイベントAPIは、例えば、ウェブページSDK210に含ませることができる。ウェブページSDK210のAPIは、ファウンデーションAPI208を使用することができる。ウェブページSDK210は、例えば、アップル社により提供されるウェブキットを含むことができる。ウェブページSDK210は、APIとしてオファーすることもできるし、或いはアプリケーション、例えば、アップル社により提供されるSAFARI(登録商標)のようなブラウザを通してアクセスすることもできる。

30

【0029】

装置で実行されるアプリケーション214は、ウェブページSDK210のAPIを使用してウェブページを生成することができる。次いで、ウェブページSDK210のAPIは、低レベルエレメントと通信し、最終的には、タッチ感知ディスプレイ又は装置及び種々の他のユーザインターフェイスハードウェアと通信することができる。各レイヤは、その下のレイヤを使用できるが、これは、常に必要とされない。例えば、ある実施形態では、装置214は、OS API206と時々通信することができる。

【0030】

40

例示的タッチイベントプロセス

図3は、APIを経てタッチ及び/又はジェスチャーイベントへのアクセスを与えるためのプロセス300のフローチャートである。このプロセス300は、1つ以上のタッチ入力信号を得ることで開始できる(302)。タッチ入力信号は、タッチ感知ディスプレイ又は装置から得ることができる。タッチイベント及び/又はジェスチャーの決定は、タッチイベントモデルを使用してタッチ入力信号に基づいて行うことができる(304)。タッチイベントは、タッチ感知ディスプレイ又は装置に表示されるウェブページの領域に関連付けることができる。例えば、タッチ感知ディスプレイは、携帯電話のディスプレイでよく、タッチ感知装置は、ノートブックコンピュータのタッチ感知パッドでよい。タッチイベント及び/又はジェスチャーイベントへのアクセスは、プログラミングインターフ

50

ェイスを経て与えることができる(306)。例えば、図2を参照して上述したHTML断片については、この断片をウェブ開発者によりHTMLドキュメントに挿入し、タッチ及び/又はジェスチャーイベントへのアクセスを開発者に与えることができる。タッチイベント及び/又はジェスチャーイベントは、HTMLドキュメントのコードによって更に処理されて、イベントアクションを開始することができる(306)。

【0031】

移動装置の概略

図4は、例示的なマルチタッチケーブル装置400のブロック図である。ある実施形態では、マルチタッチケーブル装置400は、タッチ感知ディスプレイ402を備えている。このタッチ感知ディスプレイ402は、液晶ディスプレイ(LCD)技術、光放射ポリマーディスプレイ(LPD)技術、又は他のディスプレイ技術を実施することができる。タッチ感知ディスプレイ402は、ユーザとの触覚及び/又は触感接触を感知することができる。

10

【0032】

ある実施形態では、タッチ感知ディスプレイ402は、マルチタッチ感知ディスプレイ402である。タッチ感知ディスプレイ402は、例えば、複数の同時タッチポイントを処理することができる。これは、各タッチポイントの圧力、程度及び/又は位置に関連したデータを処理することを含む。このような処理は、複数の指でのジェスチャー及び相互作用、コーディング(chording)、及び他の相互作用を容易にする。他のタッチ感知ディスプレイ技術、例えば、スタイラス又は他のポインティング装置を使用して接触がなされるディスプレイも、使用することができる。マルチタッチ感知ディスプレイ技術の幾つかの例が、米国特許第6,323,846号、第6,570,557号、第6,677,932号、及び米国特許公告第2002/0015024A1号に説明されており、その各々は、参考としてここにそのまま援用する。ある実施形態では、マルチタッチケーブル装置400は、種々のシステムオブジェクトへのユーザアクセスを与えると共にユーザへ情報を伝達するためにタッチ感知ディスプレイ402に1つ以上のグラフィックユーザインターフェイスを表示することができる。

20

【0033】

例示的なマルチタッチケーブル装置のファンクション

ある実施形態において、マルチタッチケーブル装置400は、電話装置、e-メール装置、ネットワークデータ通信装置、Wi-Fiベースステーション装置、及び媒体処理装置のような複数の装置ファンクションを実施することができる。ある実施形態では、マルチタッチケーブル装置400は、ウェブページ(例えば、ウェブページ100)を表示するためのウェブブラウザ404を含むことができる。タッチ感知ディスプレイ402は、ウェブページ100において作られるタッチ入力信号を受け取ることができ、そして上述したタッチモデルを使用して、タッチ入力信号に基づきタッチ及び/又はジェスチャーイベントを決定することができる。ある実施形態では、マルチタッチケーブル装置400は、ネットワーク配布ファンクションを実施することができる。ある実施形態では、タッチ感知ディスプレイ402は、マルチタッチケーブル装置400がユーザの耳の付近にあるときに、ロックダウンすることができる。このロックダウンは、図1Bを参照して述べたように、touchcancelイベントを生じさせる。

30

40

【0034】

ある実施形態では、加速度計472は、方向矢印474により指示されたように、マルチタッチケーブル装置400の移動を検出するのに使用することができる。従って、表示オブジェクト及び/又は媒体は、例えば、肖像又は風景のような検出された向きに基づいてプレゼンテーションすることができる。ある実施形態では、マルチタッチケーブル装置400は、グローバルポジショニングシステム(GPS)、又は他のポジショニングシステム(例えば、Wi-Fiアクセスポイント、テレビジョン信号、セルラーグリッド、ユニフォームリソースロケータ(URL)を使用するシステム)により与えられるような位置決定能力をサポートするための回路及びセンサを含むことができる。ある実施形態で

50

は、ポジショニングシステム（例えば、GPS受信器）は、マルチタッチケーパブル装置400に一体化することもできるし、又は位置に基づくサービスへのアクセスを与えるようにインターフェイスを経てマルチタッチケーパブル装置400に結合できる個別の装置として設けることもできる。又、マルチタッチケーパブル装置400は、1つ以上のワイヤレス通信サブシステムを含むこともできる。

【0035】

ある実施形態では、例えば、ユニバーサルシリアルバス（USB）ポート又はドッキングポート或いは他のワイヤードポート接続のようなポート装置を含ませることができる。このポート装置は、例えば、他のコンピューティング装置、例えば、他のマルチタッチケーパブル装置400、ネットワークアクセス装置、パーソナルコンピュータ、プリンタ、又はデータを受信及び/又は送信できる他の処理装置へのワイヤード接続を確立するのに使用することができる。ある実施形態では、ポート装置は、マルチタッチケーパブル装置400が、1つ以上のプロトコル、例えば、TCP/IP、HTTP、UDP及び他の既知のプロトコルを使用して、ホスト装置と同期をとれるようにする。

10

【0036】

ネットワークオペレーティング環境

図5は、図4のマルチタッチケーパブル装置400に対する例示的なネットワークオペレーティング環境600のブロック図である。図4のマルチタッチケーパブル装置400は、例えば、データ通信において1つ以上のワイヤード及び/又はワイヤレスネットワーク510を経て通信することができる。例えば、ワイヤレスネットワーク512、例えば、セルラーネットワークは、ゲートウェイ516の使用により、インターネットのようなワイドエリアネットワーク（WAN）514と通信することができる。同様に、802.11gワイヤレスアクセスポイントのようなアクセスポイント518は、ワイドエリアネットワーク514へ通信アクセスを与えることができる。ある実施形態では、ワイヤレスネットワーク512及びアクセスポイント518を経て音声及びデータの両通信を確立することができる。例えば、マルチタッチケーパブル装置400aは、（例えば、VoIPプロトコルを使用して）電話コールを発信及び受信することができ、（例えば、POP3プロトコルを使用して）e-メールメッセージを送信及び受信することができ、且つ（例えば、TCP/IP又はUDPプロトコルを使用して）ワイヤレスネットワーク512、ゲートウェイ516及びワイドエリアネットワーク514を経てウェブページ、写真及びビデオのような電子ドキュメント及び/又はストリームを検索することができる。同様に、マルチタッチケーパブル装置400bは、電話コールを発信及び受信し、e-メールメッセージを送信及び受信し、且つアクセスポイント518及びワイドエリアネットワーク514を経て電子ドキュメントを検索することができる。ある実施形態では、マルチタッチケーパブル装置400は、1つ以上のケーブルを使用してアクセスポイント518へ物理的に接続することができ、又、アクセスポイント518は、パーソナルコンピュータでよい。この構成では、マルチタッチケーパブル装置400は、「テザード(tethered)」装置と称することができる。

20

30

【0037】

又、マルチタッチケーパブル装置400a及び400bは、他の手段により通信を確立することもできる。例えば、マルチタッチケーパブル装置400aは、ワイヤレスネットワーク512を経て、他のワイヤレス装置、例えば、他のマルチタッチケーパブル装置400、セルラー電話、等と通信することができる。同様に、マルチタッチケーパブル装置400a及び400bは、図4に示すBluetooth（登録商標）通信装置488のような1つ以上の通信サブシステムの使用により、例えば、パーソナルエリアネットワークのようなピア・ツー・ピア通信520を確立することができる。又、他の通信プロトコル及びトポロジーを実施することもできる。

40

【0038】

マルチタッチケーパブル装置400は、例えば、1つ以上のワイヤード及び/又はワイヤレスネットワーク510を経てネットワークリソース530と通信することができる。

50

例えば、ネットワークリソースは、図 1 及び 2 を参照して述べたように、タッチモデルを経てタッチできるウェブページを配送するためのウェブサーバーでよい。

【 0 0 3 9 】

又、マルチタッチケーパブル装置 4 0 0 上のソフトウェアに対してソフトウェア更新が存在するかどうか自動的に決定し、次いで、そのソフトウェア更新をマルチタッチケーパブル装置 4 0 0 にダウンロードし、そこで、手動で又は自動的にアンパックし及び / 又はインストールすることのできるソフトウェア更新サービスを含む他のサービスを提供することもできる。

【 0 0 4 0 】

例示的移動装置アーキテクチャー

図 6 は、図 4 のマルチタッチケーパブル装置 4 0 0 の実施形態のブロック図 6 0 0 である。マルチタッチケーパブル装置 4 0 0 は、メモリインターフェイス 6 0 2 と、1 つ以上のデータプロセッサと、映像プロセッサ及び / 又は中央処理ユニット 6 0 4 と、周辺インターフェイス 6 0 6 とを含むことができる。メモリインターフェイス 6 0 2 、1 つ以上のプロセッサ 6 0 4 及び / 又は周辺インターフェイス 6 0 6 は、個別のコンポーネントでもよいし、或いは 1 つ以上の集積回路に一体化されてもよい。マルチタッチケーパブル装置 4 0 0 における種々のコンポーネントは、1 つ以上の通信バス又は信号ラインにより結合することができる。

10

【 0 0 4 1 】

センサ、装置及びサブシステムは、複数のファンクションを容易にするために、周辺インターフェイス 6 0 6 に結合することができる。例えば、運動センサ 6 1 0 、光センサ 6 1 2 、及び接近センサ 6 1 4 は、図 4 を参照して述べた方向、照明及び接近ファンクションを容易にするために、周辺インターフェイス 6 0 6 に結合することができる。又、関連ファンクションを容易にするために、ポジショニングシステム（例えば、GPS 受信器）、温度センサ、生体測定センサ、又は他のセンサ装置のような他のセンサ 6 1 6 を周辺インターフェイス 6 0 6 に接続することもできる。

20

【 0 0 4 2 】

又、記録写真及びビデオクリップのようなカメラファンクションを容易にするために、カメラサブシステム 6 2 0 及び光学的センサ 6 2 2 、例えば、電荷結合装置（CCD）又は相補的金属酸化物半導体（CMOS）光学的センサを使用することができる。

30

【 0 0 4 3 】

又、高周波受信器及び送信器、及び / 又は光学的（例えば、赤外線）受信器及び送信器を含む 1 つ以上のワイヤレス通信サブシステム 6 2 4 により通信ファンクションを容易にすることができる。通信サブシステム 6 2 4 の特定の設計及び実施は、マルチタッチケーパブル装置 4 0 0 が動作することが意図された通信ネットワーク（1 つ又は複数）に依存する。例えば、マルチタッチケーパブル装置 4 0 0 は、GSM（登録商標）ネットワーク、GPRS ネットワーク、EDGE ネットワーク、Wi-Fi 又は WiMax ネットワーク、及び Bluetooth（登録商標）ネットワークを経て動作するように設計された通信サブシステム 6 2 4 を含むことができる。特に、ワイヤレス通信サブシステム 6 2 4 は、装置 5 0 0 が他のワイヤレス装置のためのベースステーションとして構成されるようにホスティングプロトコルを含むことができる。

40

【 0 0 4 4 】

オーディオサブシステム 6 2 6 は、ボイスイネーブル型ファンクション、例えば、音声認識、音声複製、デジタル記録及び電話ファンクションを容易にするために、スピーカ 6 2 8 及びマイクロホン 6 3 0 に結合することができる。

【 0 0 4 5 】

I/O サブシステム 6 4 0 は、タッチスクリーンコントローラ 6 4 2 及び / 又は他の入力コントローラ 6 4 4 （1 つ又は複数）を含むことができる。タッチスクリーンコントローラ 6 4 2 は、タッチスクリーン 6 4 6 に結合することができる。タッチスクリーン 6 4

50

6 及びタッチスクリーンコントローラ 6 4 2 は、例えば、これらに限定されないが、容量性、抵抗性、赤外線、及び表面音波技術、並びにタッチスクリーン 6 4 6 との 1 つ以上の接触ポイントを決定するための他の接近センサレイ又は他の素子を含む複数のタッチ感知技術のいずれかを使用して、接触及び移動又はその切断を検出することができる。

【 0 0 4 6 】

他の入力コントローラ 6 4 4 (1 つ又は複数) は、他の入力 / 制御装置 6 4 8、例えば、1 つ以上のボタン、ロッカースイッチ、サムホイール、赤外線ポート、U S B ポート、及び / 又はスタイラス等のポインタ装置に結合することができる。1 つ以上のボタン (図示せず) は、スピーカ 6 2 8 及び / 又はマイクロホン 6 3 0 の音量制御のためのアップ / ダウンボタンを含むことができる。

10

【 0 0 4 7 】

一実施形態では、第 1 時間中ボタンを押すと、タッチスクリーン 6 4 6 のロックが解離し、その第 1 時間より長い第 2 時間中ボタンを押すと、マルチタッチケーブル装置 4 0 0 への電源をターンオン又はオフにすることができる。ユーザは、1 つ以上のボタンのファンクションをカスタマイズすることができる。タッチスクリーン 6 4 6 は、例えば、バーチャル又はソフトボタン、及び / 又はキーパッド又はキーボードを実施するのにも使用できる。

【 0 0 4 8 】

ある実施形態では、マルチタッチケーブル装置 4 0 0 は、M P 3、A A C、および M P E G ファイルのような記録されたオーディオ及び / 又はビデオファイルをプレゼンテーションすることができる。ある実施形態では、マルチタッチケーブル装置 4 0 0 は、i P o d ™のような M P 3 プレーヤのファンクションを含むことができる。それ故、マルチタッチケーブル装置 4 0 0 は、i P o d に適合できる 3 2 ピンコネクタを含むことができる。又、他の入力 / 出力及び制御装置を使用することもできる。

20

【 0 0 4 9 】

メモリアンターフェイス 6 0 2 は、メモリ 6 5 0 に結合することができる。メモリ 6 5 0 は、高速ランダムアクセスメモリ及び / 又は不揮発性メモリ、例えば、1 つ以上の磁気ディスク記憶装置、1 つ以上の光学的記憶装置、及び / 又はフラッシュメモリ (例えば、N A N D、N O R) を含むことができる。メモリ 6 5 0 は、オペレーティングシステム 6 5 2、例えば、D a r w i n、R T X C、L I N U X、U N I X、O S X、W I N D O W S (登録商標)、又は埋め込み型オペレーティングシステム、例えば、V x W o r k s を記憶することができる。オペレーティングシステム 6 5 2 は、基本的システムサービスをハンドリングし及びハードウェア依存性タスクを遂行するためのインストラクションを含むことができる。

30

【 0 0 5 0 】

又、メモリ 6 5 0 は、1 つ以上の付加的な装置、1 つ以上のコンピュータ、及び / 又は 1 つ以上のサーバーとの通信を容易にするために通信インストラクション 6 5 4 を記憶することもできる。メモリ 6 5 0 は、グラフィックユーザインターフェイス処理を容易にするためのグラフィックユーザインターフェイスインストラクション 6 5 6、センサ関連処理及びファンクションを容易にするためのセンサ処理インストラクション 6 5 8、電話関連プロセス及びファンクションを容易にするための電話インストラクション 6 6 0、電子メッセージング関連プロセス及びファンクションを容易にするための電子メッセージングインストラクション 6 6 2、ウェブブラウジング関連プロセス及びファンクションを容易にするためのウェブブラウジングインストラクション 6 6 4、メディア処理関連プロセス及びファンクションを容易にするためのメディア処理インストラクション 6 6 6、G P S 及びナビゲーション関連プロセス及びファンクションを容易にするための G P S / ナビゲーションインストラクション 6 6 8、カメラ関連プロセス及びファンクションを容易にするためのカメラインストラクション 6 7 0、及び / 又は図 1 - 5 を参照して述べたプロセス及びファンクションを容易にするための他のメッセージングインストラクション 6 7 2

40

50

を含むことができる。

【0051】

上述したインストラクション及びアプリケーションの各々は、上述した1つ以上のファンクションを遂行するためのインストラクションのセットに対応することができる。これらのインストラクションは、個別のソフトウェアプログラム、手順又はモジュールとして実施される必要はない。メモリ650は、付加的なインストラクションを含むこともできるし、又はより少数のインストラクションを含んでもよい。マルチタッチケーブル装置400の種々のファンクションは、1つ以上の信号処理及び/又は特定用途向け集積回路を含めて、ハードウェア及び/又はソフトウェアで実施することができる。

【0052】

ここに述べる特徴は、デジタル電子回路、或いはコンピュータハードウェア、ファームウェア、ソフトウェア、又はそれらの組合せで実施することができる。これらの特徴は、プログラム可能なプロセッサにより実行するために、例えば、マシン読み取り可能な記憶装置又は伝播信号のような情報キャリアに明白に埋め込まれたコンピュータプログラム製品において具現化することができ、そして入力データに対して動作して出力を発生することにより前記実施形態のファンクションを遂行するためにインストラクションのプログラムを実行するプログラム可能なプロセッサにより方法ステップを遂行することができる。

【0053】

上述した特徴は、データ記憶システムからデータ及びインストラクションを受け取ると共にデータ記憶システムへデータ及びインストラクションを送り出すように結合された少なくとも1つのプログラム可能なプロセッサと、少なくとも1つの入力装置と、少なくとも1つの出力装置とを含むプログラム可能なシステムにおいて実行できる1つ以上のコンピュータプログラムにおいて具現化できるのが好都合である。コンピュータプログラムとは、コンピュータに直接的又は間接的に使用して、あるアクティビティを遂行するか又はある結果を得ることができるインストラクションのセットである。コンピュータプログラムは、コンパイルされた言語又は解釈された言語を含む任意の形態のプログラミング言語（例えば、Objective-C、Java）で書くことができ、そしてコンピューティング環境で使用するのに適したコンポーネント、サブルーチン又は他のユニットをスタンドアロンプログラム又はモジュールとして含む任意の形態で展開することができる。

【0054】

インストラクションのプログラムを実行するのに適したプロセッサは、例えば、任意の種類のコピュータの、汎用及び特殊目的の両マイクロプロセッサ、単独プロセッサ又はマルチプロセッサ又はコアの1つを含む。一般的に、プロセッサは、リードオンリメモリ又はランダムアクセスメモリ或いはその両方からインストラクション及びデータを受け取る。コンピュータの本質的要素は、インストラクションを実行するためのプロセッサと、インストラクション及びデータを記憶するための1つ以上のメモリである。又、一般的に、コンピュータは、データファイルを記憶するための1つ以上の大量記憶装置を含むか、又はそれらと通信するように作動的に結合され、このような装置は、磁気ディスク、例えば、内部ハードディスク及び取り外し可能なディスクと、磁気-光学ディスクと、光学的ディスクとを含む。コンピュータプログラムインストラクション及びデータを明白に実施するのに適した記憶装置は、例えば、EPROM、EEPROM及びフラッシュメモリ装置のような半導体メモリ装置と、内部ハードディスク及び取り外し可能なディスクのような磁気ディスクと、磁気-光学ディスクと、CD-ROM及びDVD-ROMディスクとを含む全ての形態の不揮発性メモリを含む。プロセッサ及びメモリは、ASIC（特定用途向け集積回路）により補足することもできるし又はそれに組み込むこともできる。

【0055】

ユーザとの対話を与えるために、ユーザに情報を表示するためのディスプレイ装置、例えば、CRT（陰極線管）又はLCD（液晶ディスプレイ）モニタと、ユーザがコンピュータへ入力を与えることのできるキーボード及びポインティング装置、例えば、マウス又はトラックボールとを有するコンピュータにおいて、これらの特徴を具現化することがで

10

20

30

40

50

きる。

【 0 0 5 6 】

又、データサーバーのようなバックエンドコンポーネントを含むか、アプリケーションサーバー又はインターネットサーバーのようなミドルウェアコンポーネントを含むか、或いはグラフィックユーザインターフェイス又はインターネットブラウザ或いはその組合せを有するクライアントコンピュータのようなフロントエンドコンポーネントを含むコンピュータシステムにおいて、これらの特徴を具現化することもできる。システムのこれらのコンポーネントは、通信ネットワークのような任意のデジタルデータ通信形態又は媒体によって接続することができる。通信ネットワークは、例えば、LAN、WAN、及びインターネットを形成するコンピュータ及びネットワークを含む。

10

【 0 0 5 7 】

コンピュータシステムは、クライアント及びサーバーを含むことができる。クライアント及びサーバーは、一般的に、互いに離れており、典型的に、ネットワークを通して対話する。クライアント及びサーバーの関係は、各コンピュータにおいて実行されて互いにクライアント - サーバー関係を有するコンピュータプログラムによって生じる。

【 0 0 5 8 】

多数の実施形態について説明した。それでも、種々の変更がなされ得ることを理解されたい。例えば、1つ以上の実施形態の要素を結合、削除、変更、又は補足して、更に別の実施形態を形成することができる。更に別の実施例として、図示された論理フローは、希望の結果を達成するのに、図示された特定の順序又は逐次の順序を必要とするものではない。更に、他のステップを設けてもよいし、又は上述したフローからステップを削除してもよく、更に、上述したシステムに他のコンポーネントを追加しても、又、そこから除去してもよい。従って、これら他の実施形態は、特許請求の範囲に包含される。

20

【 符号の説明 】

【 0 0 5 9 】

- 1 0 0 : ウェブページ
- 1 0 2 : エlement 1 (ノード 1)
- 1 0 4 : エlement 2 (ノード 2)
- 1 0 6 : エlement 3 (ノード 3)
- 1 0 8 : エlement 4 (ノード 4)
- 1 1 0 : タッチ
- 1 1 2 : タッチ
- 1 5 2 : ノード 0
- 1 5 4 : ノード 1
- 1 5 6 : ノード 2
- 1 5 8 : ノード 3
- 1 6 0 : ノード 4
- 2 0 0 : ハードウェア
- 2 0 2 : ドライバ
- 2 0 4 : コア OS
- 2 0 6 : コアファウンデーション
- 2 0 6 : OS API
- 2 0 8 : ファウンデーション
- 2 1 0 : ウェブページ SDK
- 2 1 2 : アプリケーション
- 5 1 4 : ワイドエリアネットワーク
- 5 1 6 : ゲートウェイ
- 5 1 8 : アクセスポイント
- 5 3 0 : ネットワークリソース (ウェブサーバー)

30

40

【 図 1 A 】

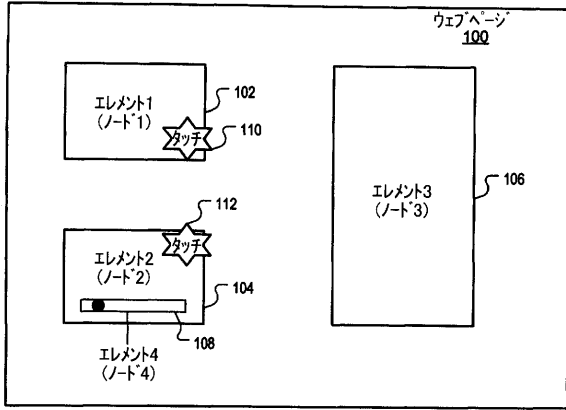


FIG.1A

【 図 2 】

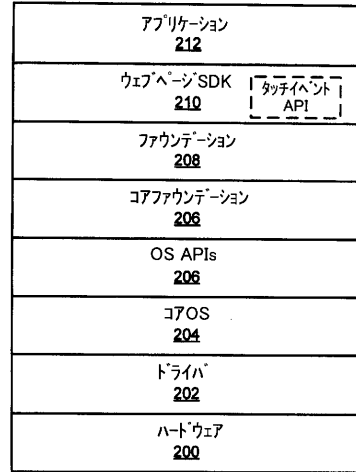


FIG.2

【 図 1 B 】

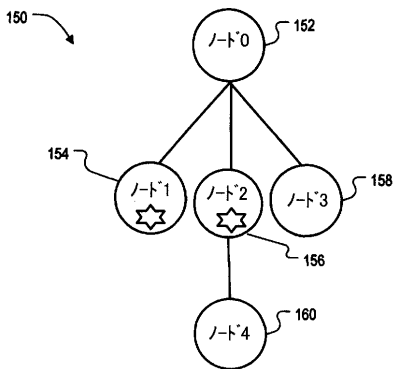


FIG.1B

【 図 3 】

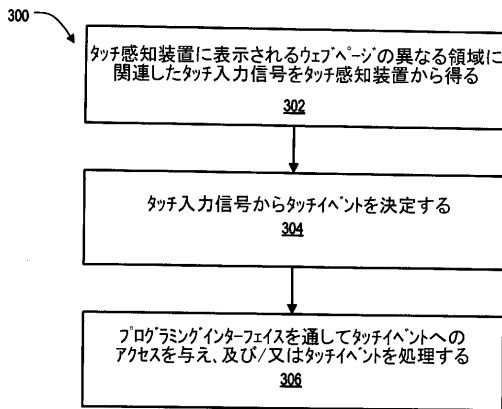


FIG.3

【 図 4 】

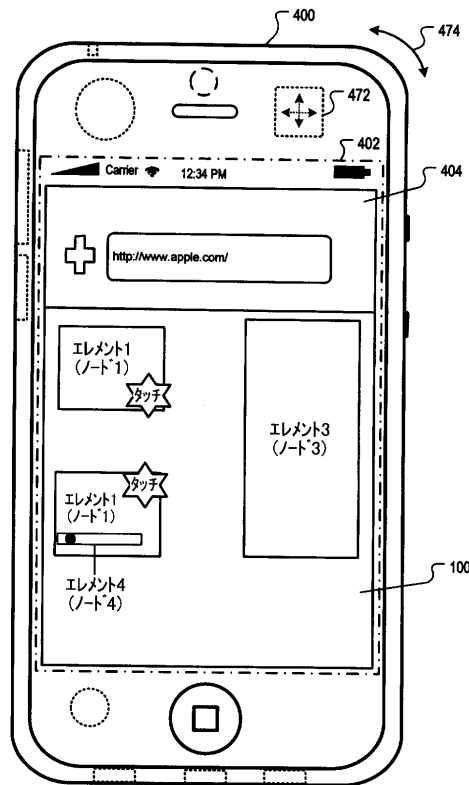


FIG.4

【 図 5 】

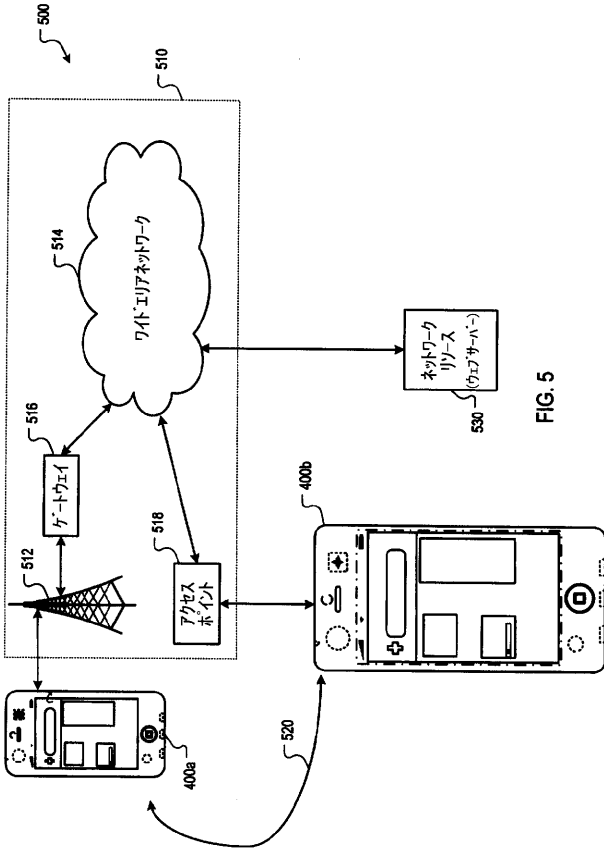


FIG. 5

【 図 6 】

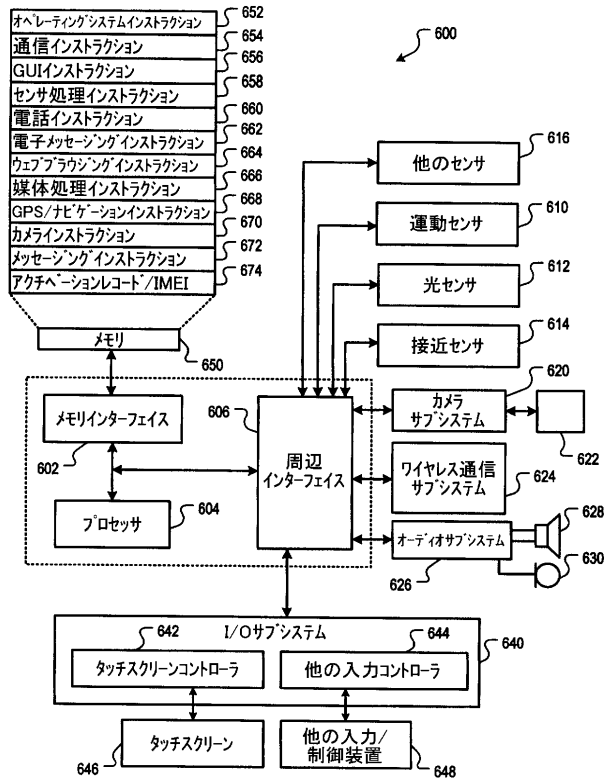


FIG. 6

【 手続 補 正 書 】

【 提 出 日 】 平 成 24 年 6 月 25 日 (2012.6.25)

【 手 続 補 正 1 】

【 補 正 対 象 書 類 名 】 特 許 請 求 の 範 囲

【 補 正 対 象 項 目 名 】 全 文

【 補 正 方 法 】 変 更

【 補 正 の 内 容 】

【 特 許 請 求 の 範 囲 】

【 請 求 項 1 】

1 つ以上のプロセッサと、前記 1 つ以上のプロセッサにより実行される 1 つ以上のプログラムを格納するメモリとを備える電子デバイスにより実行される方法であって、
複数のタッチリストを含むタッチイベントを受け取るステップと、
前記タッチイベントを処理するステップと
 を備え、

前記複数のタッチリストは、前記タッチイベントのターゲットと関連する 1 つ以上のタッチのための第 1 のタッチリストを少なくとも含むことを特徴とする方法。

【 請 求 項 2 】

前記複数のタッチリストは、

1 つ以上の変化したタッチのための第 2 のタッチリストと、

タッチ感知表面において検出された全てのタッチのための第 3 のタッチリストと

を更に含むことを特徴とする請求項 1 に記載の方法。

【 請 求 項 3 】

前記第 1 のタッチリストにおける前記 1 つ以上のタッチと関連したタッチイベントを処理するステップを含むことを特徴とする請求項 1 又は 2 に記載の方法。

【 請 求 項 4 】

各タッチリストは、ウェブページ上の1つ以上のタッチを識別するためにタッチイベントデータを含み、該タッチイベントデータは、タッチ識別子と少なくとも1セットのタッチ位置座標とを含み、

前記各タッチリストは更に各タッチと関連するタッチイベントターゲットを含み、

前記少なくとも1セットのタッチ位置座標は、少なくとも1つのクライアント座標、ページ座標、及び、スクリーン座標を含む

ことを特徴とする請求項1乃至3のいずれか1項に記載の方法。

【請求項5】

2つ以上のタッチイベントを合成してジェスチャーイベントを生成するステップを更に備えることを特徴とする請求項4に記載の方法。

【請求項6】

前記ジェスチャーイベントは、スケール情報及び回転情報のうちの1つを含み、前記スケール情報はスケール値を含み、前記回転情報は回転値を含むことを特徴とする請求項5に記載の方法。

【請求項7】

前記複数のタッチリストに含まれる値との関連で、表示されたウェブページを動的に更新するステップを更に備えることを特徴とする請求項1乃至6のいずれか1項に記載の方法。

【請求項8】

前記複数のタッチリストに含まれる値との関連で、ウェブページに表示されている要素を動的に回転またはリサイズするステップを更に備えることを特徴とする請求項1乃至6のいずれか1項に記載の方法。

【請求項9】

前記複数のタッチリストに含まれる値との関連で、ウェブページに表示されている要素を動的に回転及びリサイズするステップを更に備えることを特徴とする請求項1乃至6のいずれか1項に記載の方法。

【請求項10】

1つ以上のプロセッサと、前記1つ以上のプロセッサにより実行される1つ以上のプログラムを格納するメモリとを備える電子デバイスであって、

前記1つ以上のプログラムは、

複数のタッチリストを含むタッチイベントを受け取り、

前記タッチイベントを処理する

ためのインストラクションを含み、

前記複数のタッチリストは、前記タッチイベントのターゲットと関連する1つ以上のタッチのための第1のタッチリストを少なくとも含むことを特徴とする電子デバイス。

【請求項11】

前記複数のタッチリストは、

1つ以上の変化したタッチのための第2のタッチリストと、

タッチ感知表面において検出された全てのタッチのための第3のタッチリストと

を更に含むことを特徴とする請求項10に記載の電子デバイス。

【請求項12】

前記1つ以上のプログラムは、前記第1のタッチリストにおける前記1つ以上のタッチと関連したタッチイベントを処理するためのインストラクションを含むことを特徴とする請求項10または11に記載の電子デバイス。

【請求項13】

各タッチリストは、ウェブページ上の1つ以上のタッチを識別するためにタッチイベントデータを含み、該タッチイベントデータは、タッチ識別子と少なくとも1セットのタッチ位置座標とを含み、

前記各タッチリストは更に各タッチと関連するタッチイベントターゲットを含み、

前記少なくとも1セットのタッチ位置座標は、少なくとも1つのクライアント座標、ペ

ージ座標、及び、スクリーン座標を含む

ことを特徴とする請求項 10 乃至 12 のいずれか 1 項に記載の電子デバイス。

【請求項 14】

前記 1 つ以上のプログラムは、2 つ以上のタッチイベントを合成してジェスチャーイベントを生成するためのインストラクションを含むことを特徴とする請求項 13 に記載の電子デバイス。

【請求項 15】

前記ジェスチャーイベントは、スケール情報及び回転情報のうちの一つを含み、前記スケール情報はスケール値を含み、前記回転情報は回転値を含むことを特徴とする請求項 14 に記載の電子デバイス。

【請求項 16】

前記 1 つ以上のプログラムは、前記複数のタッチリストに含まれる値との関連で、表示されたウェブページを動的に更新するためのインストラクションを含むことを特徴とする請求項 10 乃至 15 のいずれか 1 項に記載の電子デバイス。

【請求項 17】

前記 1 つ以上のプログラムは、前記複数のタッチリストに含まれる値との関連で、ウェブページに表示されている要素を動的に回転またはリサイズするインストラクションを更に備えることを特徴とする請求項 10 乃至 15 のいずれか 1 項に記載の電子デバイス。

【請求項 18】

前記 1 つ以上のプログラムは、前記複数のタッチリストに含まれる値との関連で、ウェブページに表示されている要素を動的に回転及びリサイズするインストラクションを更に備えることを特徴とする請求項 10 乃至 15 のいずれか 1 項に記載の電子デバイス。

【請求項 19】

電子デバイスの 1 つ以上のプロセッサにより実行される 1 つ以上のプログラムを格納するコンピュータで読み取り可能な記憶媒体であって、

前記 1 つ以上のプログラムは、

複数のタッチリストを含むタッチイベントを受け取り、

前記タッチイベントを処理する

ためのインストラクションを含み、

前記複数のタッチリストは、前記タッチイベントのターゲットと関連する 1 つ以上のタッチのための第 1 のタッチリストを少なくとも含むことを特徴とする記憶媒体。

【請求項 20】

前記複数のタッチリストは、

1 つ以上の変化したタッチのための第 2 のタッチリストと、

タッチ感知表面において検出された全てのタッチのための第 3 のタッチリストと

を更に含むことを特徴とする請求項 19 に記載の記憶媒体。

【請求項 21】

前記 1 つ以上のプログラムは、前記第 1 のタッチリストにおける前記 1 つ以上のタッチと関連したタッチイベントを処理するためのインストラクションを含むことを特徴とする請求項 19 または 20 に記載の記憶媒体。

【請求項 22】

各タッチリストは、ウェブページ上の 1 つ以上のタッチを識別するためにタッチイベントデータを含み、該タッチイベントデータは、タッチ識別子と少なくとも 1 セットのタッチ位置座標とを含み、

前記各タッチリストは更に各タッチと関連するタッチイベントターゲットを含み、

前記少なくとも 1 セットのタッチ位置座標は、少なくとも 1 つのクライアント座標、ページ座標、及び、スクリーン座標を含む

ことを特徴とする請求項 19 乃至 21 のいずれか 1 項に記載の記憶媒体。

【請求項 23】

前記 1 つ以上のプログラムは、2 つ以上のタッチイベントを合成してジェスチャーイベ

ントを生成するためのインストラクションを含むことを特徴とする請求項 2 2 に記載の記憶媒体。

【請求項 2 4】

前記ジェスチャーイベントは、スケール情報及び回転情報のうちの一つを含み、前記スケール情報はスケール値を含み、前記回転情報は回転値を含むことを特徴とする請求項 2 3 に記載の記憶媒体。

【請求項 2 5】

前記 1 つ以上のプログラムは、前記複数のタッチリストに含まれる値との関連で、表示されたウェブページを動的に更新するためのインストラクションを含むことを特徴とする請求項 1 9 乃至 2 4 のいずれか 1 項に記載の記憶媒体。

【請求項 2 6】

前記 1 つ以上のプログラムは、前記複数のタッチリストに含まれる値との関連で、ウェブページに表示されている要素を動的に回転またはリサイズするインストラクションを更に備えることを特徴とする請求項 1 9 乃至 2 4 のいずれか 1 項に記載の記憶媒体。

【請求項 2 7】

前記 1 つ以上のプログラムは、前記複数のタッチリストに含まれる値との関連で、ウェブページに表示されている要素を動的に回転及びリサイズするインストラクションを更に備えることを特徴とする請求項 1 9 乃至 2 4 のいずれか 1 項に記載の記憶媒体。

フロントページの続き

(74)代理人 100134175

弁理士 永川 行光

(72)発明者 ウィリアムソン リチャード

アメリカ合衆国 カリフォルニア州 95033-7828 ロス ガトス アウター ゼイヤ
ン ト ロード 340

(72)発明者 ボルシンガ グレゴリー デニス

アメリカ合衆国 カリフォルニア州 94102 サンフランシスコ ローサット ストリート
19

(72)発明者 オマニック ティモシー ピー

アメリカ合衆国 カリフォルニア州 94041 マウンテン ヴィュー ヴィクトリー ウェイ
691 #1

Fターム(参考) 5B068 AA05 BB01 CC07 CC17

5B087 AA09 BC06 CC01 DD03 DD12

5E555 BA04 BB04 BC08 CA28 CB13 CB16 CB32 CB55 CC23 CC27

DB03 DC24 DC25 EA07

【外国語明細書】

2012164336000001.pdf