

【公報種別】特許法第17条の2の規定による補正の掲載

【部門区分】第6部門第3区分

【発行日】平成25年4月11日(2013.4.11)

【公表番号】特表2008-516335(P2008-516335A)

【公表日】平成20年5月15日(2008.5.15)

【年通号数】公開・登録公報2008-019

【出願番号】特願2007-535710(P2007-535710)

【国際特許分類】

G 06 F 3/041 (2006.01)

【F I】

G 06 F 3/041 330 C

【誤訳訂正書】

【提出日】平成25年2月25日(2013.2.25)

【誤訳訂正1】

【訂正対象書類名】明細書

【訂正対象項目名】全文

【訂正方法】変更

【訂正の内容】

【発明の詳細な説明】

【発明の名称】タッチスクリーンイベントをアプリケーションフォーマットデータに変換するための方法およびシステム

【技術分野】

【0001】

本発明は、概してタッチスクリーンイベントをアプリケーションフォーマットにフォーマットされたデータ(アプリケーションフォーマットデータ)に変換するための方法およびシステムに関する。

【背景技術】

【0002】

今日、多種多様な従来のタッチスクリーンシステムはさまざまなアプリケーションにおいて使用されている。アプリケーションの例としては、小売、レストラン、POS端末装置、キオスク、現金自動預け払い機、医療システム、Eメールパッケージ等がある。タッチスクリーンシステムは、通常タッチまたは近接センサ手段(mechanism)と連結するディスプレイを含む。センサ手段は、ディスプレイの近傍に置かれたとき、ユーザの指や手または道具を検出する。ディスプレイは制御され、特にグラフィック、文字列、映像および音声を含む、アプリケーション固有の情報をユーザに示す。アプリケーション固有の情報の例としては、仮想テレホンパッド(virtual telephone pad)、計算機、キャッシュレジスター、キーボード、電子的なドキュメント及びレシート、並びにウィンドウがある。アプリケーション固有のグラフィックは、ツールバー、ポップアップメニュー、スクロールバー、文字列入力ウィンドウ、アイコン、電子的な書き込みまたは署名ボックス等を示してもよい。

【0003】

センサ手段は、指または道具の存在を検出し、それらに応答してタッチスクリーンイベントを生成する。タッチスクリーンイベントは、タッチイベント、リリースイベント、ストリーミングまたはドラッグイベント等を示してもよい。タッチスクリーンイベントは、イベントタイプを示しかつイベントが発生した位置を識別するデータまたは信号を含む。

【0004】

ディスプレイはシステムコンピュータ上で実行中のアプリケーションにより制御される。アプリケーションはディスプレイを制御し、アプリケーション固有の情報をユーザに示

す。ディスプレイおよびタッチスクリーンはユーザインタフェースとして機能し、そのユーザインタフェースを介して、ユーザはデータをアプリケーションに出力する。ユーザ入力データは、金額、製品情報、患者／顧客情報、医療情報、患者の生体情報 (patient vitals)、テスト結果、インターネットアドレス、ウェブサイトコンテンツ、Eメールと関連するコンテンツ等を示してもよい。ユーザは、キー、メニューアイテムまたはボタンを選択し、ボックス内に記入し、仮想の英数字キーを押すこと等によりデータを入力してもよい。

#### 【発明の開示】

##### 【発明が解決しようとする課題】

##### 【0005】

しかしながら、従来のタッチスクリーンシステムにおいて、ディスプレイを駆動するアプリケーションは、直接タッチスクリーンのセンサ手段と通信する。アプリケーションを書く／修正するとき、プログラマは表示させるための情報を定義する。さらに、アプリケーションとタッチスクリーンの間の直接対話のために、プログラマはアプリケーションの中にアプリケーションとタッチスクリーンの間のインタフェースを定義する命令を組み込まなければならない。インタフェース命令は、タッチスクリーンにユーザにより入力され得るタッチスクリーンイベントの特性を規定する。

##### 【0006】

タッチスクリーンは、通常「加工していない」タッチスクリーンデータ、すなわち検出されたイベント及びそのイベントの位置を生成する。プログラマは、アプリケーションの中に、a) タッチスクリーンイベントを検証し、区別し、b) 各イベントを表示される情報と関連付け、c) それに応じて関連したソフトウェアアプリケーションを制御するために作動する機能を組み込まなければならない。したがって、プログラマは、タッチスクリーンセンサ手段の低いレベルのフォーマットおよび操作、およびタッチスクリーンイベントの特性および内容を詳細に理解する必要がある。さらに、タッチスクリーンには多数のタイプが存在し、各タイプはタッチスクリーンイベント毎に異なるフォーマットを利用してもよい。それゆえ、プログラマは各アプリケーションをタッチスクリーンの対応するタイプ毎に特徴付けなければならない。

##### 【0007】

アプリケーションソフトウェアとタッチスクリーンセンサ手段の間の汎用的なインタフェースを提供することが必要である。

##### 【課題を解決するための手段】

##### 【0008】

タッチスクリーンイベントをアプリケーションに固有にフォーマットされたデータ（アプリケーション固有のフォーマットデータ）に変換するための方法が提供される。その方法は、タッチスクリーンイベントを検出し、タッチスクリーンと関連するアクティブイベントゾーンを識別する。アクティブイベントゾーンはタッチスクリーンイベントを含む。その方法は、さらにアクティブイベントゾーンに基づいてアプリケーション固有のフォーマットデータを出力する。

##### 【0009】

また、その方法は、タッチスクリーンイベントをイベントゾーンのテーブルと比較し、潜在イベントゾーンのリストを生成する。その潜在イベントゾーンのリストから、後に、アクティブイベントゾーンが識別される。潜在イベントゾーンのリストが生成されるとすぐに、アクティブイベントゾーンは優先順位に基づいて識別されてもよい。タッチスクリーンイベントが重複するイベントゾーン内で発生した場合、1つのイベントゾーンがアクティブイベントゾーンとしてイベントゾーンの優先順位に基づいて識別される。タッチスクリーンイベントはタッチイベント、リリースイベント、ドラッグイベントの少なくとも1つを含んでもよいし、タッチスクリーン座標系に関連するイベント位置座標を備えてもよい。各イベントゾーンには少なくとも1つのモード、例えばスクロールモード、電子的書き込みモード、マウス機能モード、ボタンモード等が割り当てられてもよい。

**【発明を実施するための最良の形態】****【0010】**

図1は、タッチスクリーンベースのアプリケーションに関連して示されるタッチスクリーン10である。タッチスクリーン10は、利用可能なタッチエリアを異なるタッチまたはイベントゾーンに分ける。アプリケーションソフトウェアは、異なる機能に関してタッチエリアの異なる部分を使用してもよい。各イベントゾーンは、異なるイベント応答特性またはモードと関連してもよい。

**【0011】**

「タッチスクリーン」という言葉は、最も広い意味で使用されている。例えば、タッチスクリーンは、グラフィックまたは画像情報を示す装置または機器、例えば一体型のまたは分離可能なタッチスクリーンを有する液晶ディスプレイ（LCD）を意味してもよい。LCDはタッチセンサ式であってもよい。また、タッチスクリーンは、ガラスのような、タッチを感知できる物理的な機器を意味してもよい。物理的な機器は、グラフィックまたは画像情報を必ずしも直接表す必要はない。代わりに、タッチセンサ式の物理的な機器は、分離可能なディスプレイスクリーンの前に設置されてもよい。「タッチスクリーン」という言葉は、単独でタッチセンサ式の物理的な機器を表してもよいし、さらにより広く、タッチセンサ式の物理的な機器と組み合わせたディスプレイスクリーンを意味してもよい。

**【0012】**

タッチスクリーン10によりまたはそれに関連して示される情報は、複数のボタンゾーン14（例えば、ボタン#1、ボタン#2、ボタン#3等）を含むツールバー12を含む。バックグラウンドゾーン16は、タッチスクリーン10の中央部分に表示され、その上に重ねられたポップアップメニュー18を含む。ポップアップメニュー18は、一連のメニューアイテムゾーン20～25を含む。各メニューアイテムゾーン20～25は、アイテム機能（例えば、アイテム#1、アイテム#2等）と関連する。一例として、ボタンゾーン14においてボタン#1が選択されたときに、メニュー18を生成するようにしてもよい。縦方向のスクロールバーは縦方向のスクロールゾーン26においてユーザに示される一方、横方向のスクロールバーは横方向のスクロールゾーン28においてユーザに示される。署名ボックスは書き込みゾーン30において示される。ゾーン14～30は、以下でさらに詳細に説明されるように、異なるイベントモードまたは特性と関連する。

**【0013】**

図2は、ディスプレイ44と接続されるタッチスクリーン42を含むタッチスクリーンシステム40のブロック図である。ディスプレイ44は、ディスプレイ制御モジュール46により制御され、図1に示すようなタッチスクリーン10と接続して、グラフィックまたは画像情報を表示する。ディスプレイ制御モジュール46は、特に演算順序、配置、機能及びその他ユーザに提供されるもの等を決定し制御するアプリケーション48と通信する。アプリケーション48はタッチスクリーン制御モジュール50と通信し、タッチスクリーン制御モジュール50はタッチスクリーン42を駆動し、タッチスクリーン42からタッチスクリーンイベントを受信する。コンピュータマウス52は、任意にタッチスクリーン制御モジュール50および/またはアプリケーション48に接続されてもよい。アプリケーション48は汎用コンピュータ等で実行されてもよい。

**【0014】**

タッチスクリーン制御モジュール50は、タッチスクリーン42内のセンサに駆動信号を伝送するタッチスクリーンインタフェースまたはドライバ54を含む。また、タッチスクリーン制御モジュール50は、タッチスクリーン42から受信したタッチスクリーンイベントを処理する、イベントタイプ識別モジュール56とイベント位置識別モジュール58を含む。イベントタイプ識別モジュール56は、イベントタイプを識別する一方、イベント位置識別モジュール58は、イベント位置を識別する。イベントタイプの例としては、タッチイベント、リリースイベント、およびドラッグまたはストリーミングイベントがある。イベント位置は、画素位置、行および列の識別子、またはX-Y座標の組み合わせ

等のようなタッチスクリーン 4 2 の座標系に基づいて定義されてもよい。

【 0 0 1 5 】

タッチスクリーン制御モジュール 5 0 は、さらにゾーン位置テーブル 6 0 と、ゾーンモードテーブル 6 2 と、アプリケーションデータセットテーブル 6 4 とアプリケーションインターフェース 6 6 とを含む。

【 0 0 1 6 】

ゾーン位置テーブル 6 0 は、イベントゾーンレコードのリストを含む。各イベントゾーンレコードは一意的にイベントゾーンと関連する。ゾーン位置テーブル 6 0 のイベントゾーンレコードのリストは、ディスプレイ 4 4 上に表示されるタッチスクリーン 1 0 に関連して利用されるすべてのイベントゾーンを含んでもよい。または、ゾーン位置テーブル 6 0 は、アプリケーション 4 8 の動作中ディスプレイ 4 4 上に表示されるように、複数のタッチスクリーン 1 0 と関連するイベントゾーンレコードのリスト一式を格納してもよい。後者の場合、各イベントゾーンレコードは、現在のタッチスクリーン 1 0 に関連して現在利用されているイベントゾーンを示す「動作可能な」フィールドも含む。

【 0 0 1 7 】

各イベントゾーンレコードは、イベントゾーン ID、関連するイベントゾーンの境界を定義する座標、例えばイベントゾーンの四辺形の対角部（例えば、 $X_n$ 、 $Y_n$  及び  $x_n$ 、 $y_n$ ）、イベントゾーンのサイズ、イベントゾーンの形状、重複フラグ  $F_{overlap}$ 、選択順位 (preference ranking)  $P_{rank}$  等を含んでもよい。イベントゾーンは長方形、正方形、円形、橢円形、三角形および任意の他の境界のある形状であってよい。重複フラグ  $F_{overlap}$  は、あるイベントゾーンが他のイベントゾーン（例えば、ポップアップウィンドウ）と重なっているか否かを示すために利用される。選択または優先順位 (priority ranking)  $P_{rank}$  は、重複している 2 つ以上のイベントゾーン内でタッチスクリーンイベントが発生したとき、どのイベントゾーンをアクティブにするかを決定するために使用されてもよい。例えば、ポップアップメニューが他のグラフィック、例えばアイコン、ツールバーボタン等と重複している場合である。ポップアップウィンドウのメニューアイテムゾーンは、その下に配置されたグラフィックと関連するイベントゾーンよりも高い優先または選択順位が与えられてもよい。

【 0 0 1 8 】

ゾーンモードテーブル 6 2 は、イベントゾーン ID 及び 1 つ以上のイベントモードフラグ  $F_{mode\#N}$  を含むゾーンモードレコードを格納する。ゾーンモードテーブル 6 2 内のイベントゾーン ID は、ゾーン位置テーブル 6 0 内のイベントゾーン ID に対応して、それらの間でのクロスリファレンスを可能にする。イベントモードフラグ  $F_{mode\#N}$  は、要求されるイベントタイプおよび／またはイベント順序を、アプリケーションフォーマットにフォーマットされたデータセット（アプリケーションフォーマットデータセット）の形式で、アプリケーション 4 8 に出力されるアプリケーションに固有の応答と関連付けるために使用される。一例として、イベントモードは、 $F_{mode\#1} =$  “イベントゾーンにおけるタッチ応答”、 $F_{mode\#2} =$  “イベントゾーンにおけるタッチ応答なし”、 $F_{mode\#3} =$  “タッチしてクリック”、 $F_{mode\#4} =$  “リリースしてクリック”、 $F_{mode\#5} =$  “タッチしてドラッグ”、 $F_{mode\#6} =$  “ダブルクリック左ボタン”、 $F_{mode\#6} =$  “右クリックボタン” 等がある。

【 0 0 1 9 】

上記例において、イベントモード  $F_{mode\#1}$  は、タッチイベントが検出されたときに、タッチスクリーン制御モジュール 5 0 が、すぐにタッチ応答をアプリケーションインターフェース 6 6 からアプリケーション 4 8 に出力すべきことを示す。イベントモード  $F_{mode\#2}$  は、タッチイベントが検出されたときに、タッチスクリーン制御モジュール 5 0 がいかなる出力もなさず、タッチイベントを無視すべきことを示す。イベントモード  $F_{mode\#3}$  は、タッチイベントが検出されたときに、タッチスクリーン制御モジュール 5 0 が、すぐにコンピュータマウスの左ボタンのクリックに対応するコマンドを出力すべきことを示す。イベントモード  $F_{mode\#4}$  は、有効なタッチイベントと有効なリリースイベン

トの両方を検出した後のみ、タッチスクリーン制御モジュール 50 がコンピュータマウスの左ボタンのクリックに対応するコマンドを出力すべきことを示す。イベントモード  $F_{m\_o\_d\_e\_5}$  及び  $F_{m\_o\_d\_e\_5}$  は、関連するイベントゾーン内の対応する有効な一連のタッチ及びリリースイベントを検出した後、タッチスクリーン制御モジュール 50 がコンピュータマウスの左ボタンのダブルクリック及び右ボタンのシングルクリックそれぞれに対応するコマンドを出力すべきことを示す。

#### 【0020】

アプリケーションデータセットテーブル 64 はデータセットを格納する。各データセットは固有のアプリケーションのためにフォーマットされる。各アプリケーションフォーマットデータセットは、アプリケーション 48 により定義され、アプリケーション 48 が受け入れ可能な入力値を示す。一例として、アプリケーションフォーマットデータセットは、シングル左ボタンマウスクリック、ダブル左ボタンマウスクリック、右ボタンマウスクリック、ASCII 文字、ASCII 文字列、例えばエンター、コントロール、またはアルト機能等のキーボード機能、計算機と関連する機能、例えば署名を識別する等の一連の座標、または入力機器からデータシーケンスにより起動されてもよい任意のシステム機能コマンドと関連するコマンドを示してもよい。また、アプリケーションフォーマットデータセットは、ボタンや実質的なキーボードのキーを再定義またはその出力先を変更してもよい。例えば、キーボードのキー配列を整理し直してもよい。

#### 【0021】

初期化中、アプリケーション 48 はゾーン位置テーブル 60、ゾーンモードテーブル 62 及びアプリケーションデータセットテーブル 64 をアプリケーションインターフェース 66 を介して読み込んでもよい。また、アプリケーションは、同時にリアルタイムで、ゾーン位置テーブル 60、ゾーンモードテーブル 62 及びアプリケーションデータセットテーブル 64 を変更してもよい。

#### 【0022】

アプリケーション 48 及びタッチスクリーン制御モジュール 50 は、シングルプロセッサ、並列プロセッサ、個別の専用プロセッサ等を利用して実行されてもよい。タッチスクリーン制御モジュール 50 は、アプリケーション 48 を実行するホストコンピュータシステムから個別のエンティティを示してもよい。また、タッチスクリーン制御モジュール 50 は、ホストコンピュータシステムの一部として実行されてもよい。また、タッチスクリーン制御モジュール 50 およびアプリケーション 48 の機能は、任意にホストおよび個別のコンピュータシステムにより組み合わせて、または別個の (distinct) 1 組の個別の独立の機能エンティティとして実行されてもよい。

#### 【0023】

以下、図 3A 及び図 3B を参照して、タッチスクリーン制御モジュール 50 の動作をより詳細に説明する。

#### 【0024】

図 3A 及び図 3B は、タッチスクリーン制御モジュール 50 により実行される、タッチスクリーンイベントをアプリケーションフォーマットデータセットに変換するための処理の論理フローチャートである。ステップ 100 において、タッチスクリーン 42 はタッチスクリーンイベントを検出し、イベントタイプおよびイベント位置をタッチスクリーン制御モジュール 50 に提供する。イベントタイプ識別モジュール 56 は、ステップ 102 においてイベントタイプを識別する。イベント位置識別モジュール 58 は、ステップ 104 においてイベント位置をゾーン位置テーブル 60 内のイベントゾーンレコードと比較する。ステップ 104 における比較は、タッチスクリーンイベントの位置を現在選択されたイベントゾーンの境界座標と比較することにより実行される。

#### 【0025】

タッチスクリーンイベント位置がイベントゾーンの境界の内部にある場合、ステップ 106 においてイベントゾーンは潜在イベントゾーンのリストに追加される。ステップ 108 において、a) ステップ 106 において解析されたイベントゾーンがゾーン位置テーブ

ル 6 0 内の最後のイベントゾーンであるか否か、 b ) イベントゾーンがバックグラウンドゾーンであるか否か、および c ) 重複フラグが現在のイベントゾーンに関連して設定されたか否かが判定される。重複フラグは、現在のイベントゾーンがディスプレイ 4 4 上の他のイベントゾーンと重複する場合に、設定される。ステップ 1 0 8 の判定が Y e s の場合、フローはステップ 1 1 0 に進み、処理はゾーン位置テーブル 6 0 ( 図 2 ) 内の次のイベントゾーンレコードに移動する。ステップ 1 0 6 、 1 0 8 及び 1 1 0 は、各イベントゾーンレコードが検討され、またはイベント位置がバックグラウンドゾーン内に存在すると判定されるまで、繰り返される。

#### 【 0 0 2 6 】

ステップ 1 1 2 において、重複フラグが潜在イベントゾーンリスト上のイベントゾーンに対してクリアか否かが判定される。 Y e s の場合、フローは図 3 B のステップ 1 1 8 に進む。 N o の場合、フローはステップ 1 1 4 に進み、現在イベントゾーンが、ゾーン位置テーブル 6 0 内の最後のイベントゾーンを示すか否かが判定される。またステップ 1 1 4 において、イベント位置がディスプレイ 4 4 上で現在利用されているすべてのイベントゾーンの外側になるか否かが判定される。ステップ 1 1 4 における判定が Y e s の場合、フローはステップ 1 1 6 に進み、イベント位置がバックグラウンドゾーン内にあると判定され、処理は終了する。ステップ 1 1 4 において、イベント位置が少なくとも 1 つの他のイベントゾーン内にあると判定される場合、フローは図 3 B のステップ 1 1 8 に進む。

#### 【 0 0 2 7 】

図 3 B に戻り、ステップ 1 1 8 において、潜在イベントゾーンリストが空か否かが判定される。 Y e s の場合、バックグラウンドゾーンはステップ 1 2 0 においてアクティブである指定され、フローは終了する。また、ステップ 1 1 8 において、潜在イベントゾーンリストが空でない場合、フローはステップ 1 2 2 に進み、潜在イベントゾーンリストが、優先度が最も高いイベントゾーンを求めるために検索される。ゾーン位置テーブル 6 0 内の各イベントゾーンレコードは、優先度が最も高いイベントゾーンを識別するためのステップ 1 2 2 で使用される選択または優先順位を備える。ステップ 1 2 4 において、優先度が最も高いイベントゾーンは、アクティブイベントゾーンに指定される。ステップ 1 2 6 において、アクティブイベントゾーンのゾーンモードテーブル 6 2 内のゾーンモードレコードがアクセスされ、アクティブイベントゾーンと関連するイベントモードを取得する。ステップ 1 2 8 において、イベントモードがアプリケーション応答を含むか否かが判定される。アプリケーション応答が存在する場合、これはタッチスクリーン制御モジュール 5 0 がいくつかのタイプのデータセットをアプリケーション 4 8 ( 図 2 ) に提供するべきであることを示す。イベントモードがアプリケーション応答を含まない場合、フローはステップ 1 3 0 に進み、タッチスクリーンイベントは廃棄され、この処理を終了する。ステップ 1 2 8 においてイベントモードがアプリケーション応答を含む場合、フローはステップ 1 3 2 に進み、ゾーンモードテーブル 6 2 はアクセスされ、イベントモードおよびイベントタイプに基づいてモードフラグを取得する。ステップ 1 3 4 において、ゾーンモードテーブル 6 2 ( 図 2 ) からのインデックスまたはモードフラグを使用し、テーブル 6 4 に設定されたアプリケーションデータ内において、そのときアプリケーション 4 8 に出力されるアプリケーションフォーマットデータセットを識別する。その後処理は終了し、フローはステップ 1 0 0 に戻り、次のタッチスクリーンイベントの検出を待つ。

#### 【 0 0 2 8 】

アプリケーションベースの座標系は、タッチスクリーン 4 2 の座標系とは異なってよい。例えば、アプリケーションベースの座標系はより低い解像度 ( 例えば、 1 0 2 4 × 1 0 2 4 ) を有する一方で、タッチスクリーン 4 2 は第 1 の解像度 ( 例えば、 4 0 0 0 × 4 0 0 0 ) を有する座標系を含んでもよい。また、タッチスクリーン 4 2 は、極座標系に基づいて動作する一方で、アプリケーションベースの座標系はデカルト座標系であってよい ( 逆もまた同様 ) 。タッチスクリーン制御モジュール 5 0 は座標系間の変換を実施する。

#### 【 0 0 2 9 】

また、タッチスクリーン制御モジュール50は、ユーザがタッチスクリーンで指または道具をドラッグするとき、ユーザの指（例えば、マウスまたはペン（line））に従い、その下に配置されたグラフィックの表示がユーザの指に遅れを取るような「遅延ドラッグ」機能を備えてもよい。また、タッチスクリーン制御モジュール50は、ユーザの指がタッチスクリーンの境界に近づいたとき、アプリケーション48に出力されるイベント位置情報がユーザの指の実際の位置より境界に近いと示されるような、タッチスクリーンの境界の近くで「拡張（extended）タッチ」機能を備えてもよい。イベントゾーンが小さいときや、ディスプレイ44の角や側辺の近くに位置しているときに、拡張タッチ機能は有用である。

#### 【0030】

さまざまな具体的実施の形態において本発明を記述したが、当事者は本発明を請求項の意図および範囲内の変更を加えても実施できると認めるであろう。

#### 【図面の簡単な説明】

#### 【0031】

【図1】本発明の実施の形態による、タッチスクリーンアプリケーションに関連して示されるタッチスクリーンを示す図

【図2】本発明の実施の形態により形成されるタッチスクリーンシステムのブロック図

【図3A】本発明の実施の形態による、タッチスクリーンイベントをアプリケーション固有のフォーマットデータに変換する論理フローチャート

【図3B】本発明の実施の形態による、タッチスクリーンイベントをアプリケーション固有のフォーマットデータに変換する論理フローチャート

#### 【符号の説明】

#### 【0032】

- 10 タッチスクリーン
- 12 ツールバー
- 14 ボタンゾーン
- 16 バックグラウンドゾーン
- 18 ポップアップメニュー
- 20、21、22、23、24、25 メニューアイテムゾーン
- 26 縦方向のスクロールゾーン
- 28 横方向のスクロールゾーン
- 30 書き込みゾーン
- 40 タッチスクリーンシステム
- 42 タッチスクリーン
- 44 ディスプレイ
- 46 ディスプレイ制御モジュール
- 48 アプリケーション
- 50 タッチスクリーン制御モジュール
- 52 マウス
- 54 タッチスクリーンドライバ
- 56 イベントタイプ識別モジュール
- 58 イベント位置識別モジュール
- 60 ゾーン位置テーブル
- 62 ゾーンモードテーブル
- 64 アプリケーションデータセットテーブル
- 66 アプリケーションインターフェース

#### 【誤訳訂正2】

【訂正対象書類名】特許請求の範囲

【訂正対象項目名】全文

【訂正方法】変更

**【訂正の内容】****【特許請求の範囲】****【請求項 1】**

タッチスクリーンイベントをアプリケーションフォーマットデータに変換するための方法において、

タッチスクリーンイベントを検出するステップと、

アクティブイベントゾーンが優先順位の基礎とする、潜在イベントゾーンのリストを生成するステップと、

タッチスクリーンと関連し、タッチスクリーンイベントを包含するアクティブイベントゾーンを識別するステップと、

アクティブイベントゾーンおよびタッチスクリーンイベントのタイプに基づいてイベントモードを識別するステップと、

アクティブイベントゾーンを用いてアプリケーションデータセットテーブルにアクセスすることにより特定のアプリケーションのためのアプリケーションフォーマットデータを生成するステップであって、前記アプリケーションデータセットテーブルは前記特定のアプリケーションのための、アクティブイベントゾーンに関連するインプット値を規定する少なくとも一つのアプリケーションフォーマットデータセットを含む、ステップと、

アクティブイベントゾーンに基づいて前記アプリケーションフォーマットデータを前記特定のアプリケーションに出力するステップと

を含むことを特徴とする方法。

**【請求項 2】**

少なくとも 1 つの潜在イベントゾーンを判定するために、タッチスクリーンイベントをイベントゾーンのテーブルと比較するステップをさら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】**

タッチスクリーンイベントが、重複している第 1 および第 2 のイベントゾーンまたは複数の重複しているゾーンの中にあるか否かを判定するステップをさらに含むことを特徴とする請求項 1 に記載の方法。

**【請求項 4】**

複数のゾーンをタッチスクリーンと関連付けるステップをさらに含むことを特徴とする請求項 1 に記載の方法。

**【請求項 5】**

イベントゾーンに、スクロールモード、電子的書き込みモード、マウス機能モードおよびボタンモードの内の少なくとも 1 つを指定するステップをさらに含むことを特徴とする請求項 1 に記載の方法。

**【請求項 6】**

前記タッチスクリーンイベントは、タッチイベント、リリースイベント、ドラッグイベントおよびストリーミングイベントの内の少なくとも 1 つを含むことを特徴とする請求項 1 に記載の方法。

**【請求項 7】**

前記タッチスクリーンイベントは、タッチスクリーン座標系と関連するイベント位置座標を含むことを特徴とする請求項 1 に記載の方法。

**【請求項 8】**

タッチスクリーン座標系、および、タッチスクリーン座標系とは異なるアプリケーションベースの座標系をさらに含むことを特徴とする請求項 1 に記載の方法。

**【請求項 9】**

アプリケーション固有の情報を表示するタッチスクリーンと、

アプリケーション固有の情報を表示する前記タッチスクリーンと対話するアプリケーションであって、前記アプリケーションは、アプリケーション固有の情報に関連して前記アプリケーションにより利用されるアプリケーションフォーマットデータセットを定義する

ものである、アプリケーションと、

前記タッチスクリーンで発生するタッチスクリーンイベントを感知する、タッチスクリーンに隣接するセンサユニットと、

前記タッチスクリーン上に表示されるアプリケーション固有の情報と関連するイベントゾーンを含むイベントテーブルと、

前記アプリケーションのためのアプリケーションフォーマットデータセットを規定する少なくとも一つのデータセットを含むアプリケーションデータセットテーブルと、

アクティブイベントゾーンが優先順位の基礎とする潜在イベントゾーンのリストを生成し、前記イベントテーブル内の前記イベントゾーンから、前記タッチスクリーンイベントを含むアクティブイベントゾーンを識別し、アクティブイベントゾーンおよびタッチスクリーンイベントのタイプに基づいてイベントモードを識別し、アクティブイベントゾーンを用いて前記アプリケーションデータセットテーブルにアクセスすることにより前記アプリケーションフォーマットデータを生成し、更に、前記アクティブイベントゾーンに基づいて前記アプリケーションフォーマットデータセットを前記アプリケーションに出力するタッチスクリーン制御モジュールと  
を備えることを特徴とするタッチスクリーンシステム。

#### 【請求項 10】

タッチスクリーンイベントをアプリケーションフォーマットデータセットに変換するためのタッチスクリーン制御モジュールにおいて、

タッチスクリーンと通信するために構成されるタッチスクリーンインタフェースであつて、前記タッチスクリーンインタフェースは、タッチスクリーンイベントを受信し、各タッチスクリーンイベントは、タッチイベント、リリースイベント、ドラッグイベントおよびストリーミングイベントの内の少なくとも1つと関連し、アプリケーションインタフェースは、ソフトウェアアプリケーションと通信するために構成され、前記アプリケーションはイベントゾーン、前記イベントゾーンと関連するイベントモードおよび前記イベントモードと関連するアプリケーションフォーマットデータセットを定義する、タッチスクリーンインタフェースと、

イベントモードを前記イベントゾーンと関連付けるゾーンモードレコードを含むゾーンモードテーブルと、

前記アプリケーションのための少なくとも一つのアプリケーションフォーマットデータセットを規定する少なくとも一つのデータセットを含むアプリケーションデータセットテーブルと

を含み、

前記タッチスクリーン制御モジュールは、アクティブイベントゾーンが優先順位の基礎とする潜在イベントゾーンのリストを生成し、前記タッチスクリーンイベントに基づいて、前記イベントゾーンの一つをアクティブイベントゾーンとして指定するイベント識別子を生成し、アクティブイベントゾーンを用いて前記アプリケーションデータセットテーブルにアクセスすることにより前記アプリケーションフォーマットデータを生成し、前記アプリケーションインタフェースは、前記アプリケーションに、前記タッチスクリーンイベントと関連する前記アプリケーションフォーマットデータセットを、前記アクティブイベントゾーンおよびそれに関連するイベントモードに基づいて出力することを特徴とするタッチスクリーン制御モジュール。

#### 【請求項 11】

前記タッチスクリーンインタフェースは、一連のタッチイベントおよびリリースイベントを受信し、

それらに基づいて、前記アプリケーションインタフェースは、シングル左ボタンマウスクリック、ダブル左ボタンマウスクリック、右ボタンマウスクリック、A S C I I 文字、A S C I I 文字列、キーボード機能、計算機能、署名、一連の座標、およびシステム機能コマンドの内の少なくとも1つと対応するコマンドを出力することを特徴とする請求項10に記載のタッチスクリーン制御モジュール。

## 【誤訳訂正3】

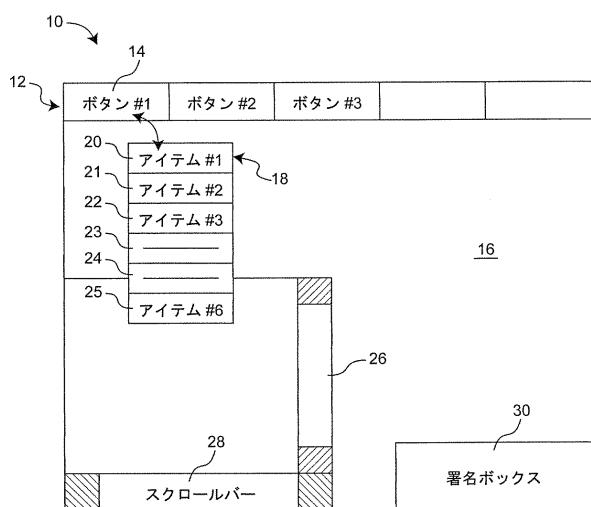
【訂正対象書類名】図面

【訂正対象項目名】全図

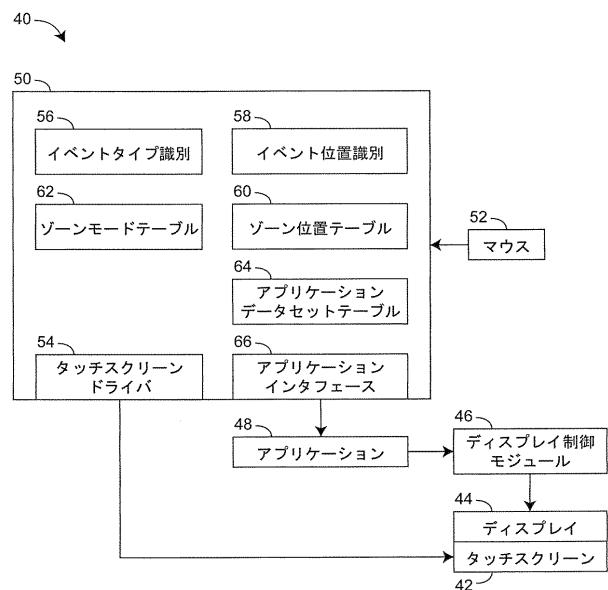
【訂正方法】変更

【訂正の内容】

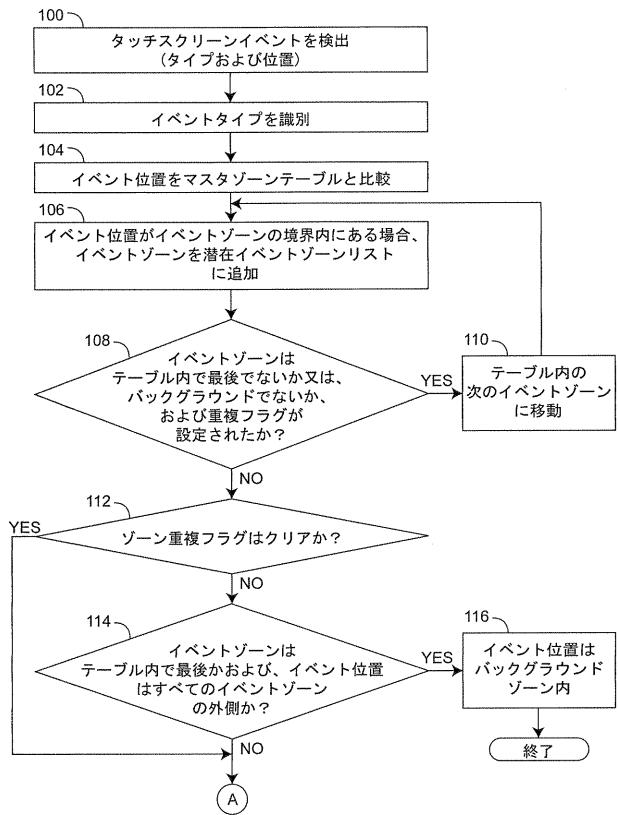
【図1】



【図2】



【図3A】



【図3B】

