

(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2009-31864

(P2009-31864A)

(43) 公開日 平成21年2月12日(2009.2.12)

(51) Int.Cl.	F I	テーマコード (参考)
G06F 3/048 (2006.01)	G06F 3/048 655A	5C082
G09G 5/00 (2006.01)	G09G 5/00 510H	5E501
G09G 5/14 (2006.01)	G09G 5/00 530T	
G09G 5/36 (2006.01)	G09G 5/00 550X	
	G09G 5/14 A	
審査請求 有 請求項の数 7 O L (全 17 頁) 最終頁に続く		

(21) 出願番号	特願2007-192408 (P2007-192408)	(71) 出願人	392026693
(22) 出願日	平成19年7月24日 (2007.7.24)		株式会社エヌ・ティ・ティ・ドコモ
			東京都千代田区永田町二丁目11番1号
		(74) 代理人	100098084
			弁理士 川▲崎▼ 研二
		(74) 代理人	100128598
			弁理士 高田 聖一
		(74) 代理人	100136146
			弁理士 小谷 明生
		(72) 発明者	加納 出亜
			東京都千代田区永田町二丁目11番1号
			株式会社エヌ・ティ・ティ・ドコモ内
		(72) 発明者	山田 英樹
			東京都千代田区永田町二丁目11番1号
			株式会社エヌ・ティ・ティ・ドコモ内
			最終頁に続く

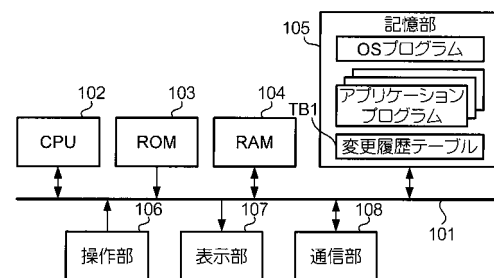
(54) 【発明の名称】 情報処理装置およびプログラム

(57) 【要約】

【課題】ウィンドウ数が多くなった時にユーザにとって優先度の低いウィンドウを最小化する。

【解決手段】携帯電話機1は複数のウィンドウが表示部107に表示されている状態において、いずれかのウィンドウのサイズを変更する操作が行われると、サイズ変更後に各ウィンドウを重複させることなく表示部107に表示させることが可能か判断する。携帯電話機1は、サイズ変更後に各ウィンドウを重複させることなく表示部107に表示できない場合には、サイズ変更されたウィンドウ以外のウィンドウにおいて、ウィンドウのサイズ変更回数が最も多いウィンドウを最小化する。

【選択図】 図2



【特許請求の範囲】**【請求項 1】**

アプリケーションプログラムを記憶した記憶手段と、
ユーザにより操作される操作手段と、
表示手段に表示されたウィンドウのサイズをユーザが前記操作手段にした操作に応じて
変更するサイズ変更手段と、
前記サイズ変更手段によりサイズを変更されたウィンドウのサイズ変更回数を記憶する
変更回数記憶手段と、
前記記憶手段に記憶されたアプリケーションプログラムをユーザが前記操作手段にした
操作に応じて実行する実行手段と、
前記実行手段により実行されたアプリケーションプログラムに対応したウィンドウを表
示手段に表示させる表示制御手段であって、前記実行手段がアプリケーションプログラ
ムの実行を始める際に前記表示手段に複数のウィンドウが表示されている場合、前記表示
手段に既に表示されている複数のウィンドウと、前記実行手段により実行されるアプリ
ケーションプログラムに対応したウィンドウとを重複させることなく前記表示手段の表示領
域内に表示できない場合には、前記表示手段に既に表示されている各ウィンドウのサイズ
変更回数を前記変更回数記憶手段から読み出し、読み出したサイズ変更回数が最大である
ウィンドウのサイズを最小化して前記表示手段に表示させる表示制御手段と
を有する情報処理装置。

10

【請求項 2】

前記表示制御手段は、サイズ変更回数が最大であるウィンドウのサイズを最小化しても
複数のウィンドウを重複させることなく前記表示手段の表示領域内に表示できない場合
には、読み出したサイズ変更回数が大であるウィンドウから順次ウィンドウのサイズを最
小化し、一のウィンドウのサイズを最小化する毎に、最小化したウィンドウと、最小化さ
れていないウィンドウと、前記実行手段により実行されるアプリケーションプログラムの
ウィンドウとを重複させることなく前記表示手段の表示領域内に表示可能であるか判断し、
表示可能であると判断した場合には、最小化されたウィンドウと、最小化されていない
ウィンドウと、前記実行手段により実行されるアプリケーションプログラムのウィンドウ
とを前記表示手段に表示させること
を特徴とする請求項 1 に記載の情報処理装置。

20

30

【請求項 3】

アプリケーションプログラムを記憶した記憶手段と、
ユーザにより操作される操作手段と、
表示手段に表示されたウィンドウのサイズをユーザが前記操作手段にした操作に応じて
変更するサイズ変更手段と、
前記サイズ変更手段によりサイズを変更されたウィンドウのサイズ変更回数を記憶する
変更回数記憶手段と、
前記記憶手段に記憶されたアプリケーションプログラムをユーザが前記操作手段にした
操作に応じて実行する実行手段と、
前記実行手段により実行されたアプリケーションプログラムに対応したウィンドウを表
示手段に表示させる表示制御手段であって、前記サイズ変更手段がウィンドウのサイズ
を変更する際に前記表示手段に複数のウィンドウが表示されている場合、表示されてい
るウィンドウにおいて前記サイズ変更手段によりサイズ変更されないウィンドウと、前記サ
イズ変更手段によりサイズ変更されるウィンドウとを重複させることなく前記表示手段
の表示領域内に表示できない場合には、前記表示手段に表示されているウィンドウにお
いて前記サイズ変更手段によりサイズ変更されない各ウィンドウのサイズ変更回数を
前記変更回数記憶手段から読み出し、読み出したサイズ変更回数が最大であるウィンド
ウのサイズを最小化して前記表示手段に表示させる表示制御手段と
を有する情報処理装置。

40

【請求項 4】

50

前記表示制御手段は、サイズ変更回数が最大であるウィンドウのサイズを最小化しても複数のウィンドウを重複させることなく前記表示手段の表示領域内に表示できない場合には、読み出したサイズ変更回数が大であるウィンドウから順次ウィンドウのサイズを最小化し、一のウィンドウのサイズを最小化する毎に、最小化したウィンドウと、最小化されていないウィンドウと、前記サイズ変更手段によりサイズ変更されたウィンドウとを重複させることなく前記表示手段の表示領域内に表示可能であるか判断し、表示可能であると判断した場合には、最小化したウィンドウと、最小化されていないウィンドウと、前記サイズ変更手段によりサイズ変更されたウィンドウとを前記表示手段に表示させることを特徴とする請求項 3 に記載の情報処理装置。

【請求項 5】

10

前記表示制御手段は、前記表示手段に表示させる複数のウィンドウが重複しないように各ウィンドウを前記表示手段に表示させることを特徴とする請求項 1 または請求項 3 に記載の情報処理装置。

【請求項 6】

コンピュータを、

ユーザにより操作される操作手段と、

表示手段に表示されたウィンドウのサイズをユーザが前記操作手段にした操作に応じて変更するサイズ変更手段と、

前記サイズ変更手段によりサイズを変更されたウィンドウのサイズ変更回数を記憶する変更回数記憶手段と、

20

記憶手段に記憶されたアプリケーションプログラムをユーザが前記操作手段にした操作に応じて実行する実行手段と、

前記実行手段により実行されたアプリケーションプログラムに対応したウィンドウを表示手段に表示させる表示制御手段であって、前記実行手段がアプリケーションプログラムの実行を始める際に前記表示手段に複数のウィンドウが表示されている場合、前記表示手段に既に表示されている複数のウィンドウと、前記実行手段により実行されるアプリケーションプログラムに対応したウィンドウとを重複させることなく前記表示手段の表示領域内に表示できない場合には、前記表示手段に既に表示されている各ウィンドウのサイズ変更回数を前記変更回数記憶手段から読み出し、読み出したサイズ変更回数最大であるウィンドウのサイズを最小化して前記表示手段に表示させる表示制御手段

30

として機能させるプログラム。

【請求項 7】

コンピュータを、

ユーザにより操作される操作手段と、

表示手段に表示されたウィンドウのサイズをユーザが前記操作手段にした操作に応じて変更するサイズ変更手段と、

前記サイズ変更手段によりサイズを変更されたウィンドウのサイズ変更回数を記憶する変更回数記憶手段と、

記憶手段に記憶されたアプリケーションプログラムをユーザが前記操作手段にした操作に応じて実行する実行手段と、

40

前記実行手段により実行されたアプリケーションプログラムに対応したウィンドウを表示手段に表示させる表示制御手段であって、前記サイズ変更手段がウィンドウのサイズを変更する際に前記表示手段に複数のウィンドウが表示されている場合、表示されているウィンドウにおいて前記サイズ変更手段によりサイズ変更されないウィンドウと、前記サイズ変更手段によりサイズ変更されるウィンドウとを重複させることなく前記表示手段の表示領域内に表示できない場合には、前記表示手段に表示されているウィンドウにおいて前記サイズ変更手段によりサイズ変更されない各ウィンドウのサイズ変更回数を前記変更回数記憶手段から読み出し、読み出したサイズ変更回数最大であるウィンドウのサイズを最小化して前記表示手段に表示させる表示制御手段

として機能させるプログラム。

50

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、コンピュータ装置のグラフィカルユーザインターフェースに関する。

【背景技術】

【0002】

オペレーティングシステム上でアプリケーションプログラムを実行するコンピュータ装置の中には、複数のアプリケーションプログラムを並行して実行し、ユーザインターフェースとなるウィンドウをアプリケーション毎に表示するものがあるが、多数のプログラムを実行し、ウィンドウの数が増えると表示画面が複雑になってしまう。この場合、ユーザは使用の優先順位の低いアプリケーションのウィンドウをアイコン化して表示画面を整理する。

10

【0003】

しかしながら、ユーザがアイコン化するウィンドウを選び、アイコン化する操作を行うことは、ユーザにとって面倒な作業であることから、このような面倒さを解消する技術として、例えば、特許文献1に開示されているシステムが考案されている。特許文献1に開示されているシステムは、新たにウィンドウを表示する際には、既に表示されているウィンドウの数を検知する。そして、検知したウィンドウの数が所定値以上の場合、ウィンドウがフォーカスされていた時間を加味して各ウィンドウに優先順位を付し、優先順位の低いウィンドウをアイコン化する。このシステムによれば、ウィンドウ数が増えると自動的にいずれかのウィンドウがアイコン化されるので、画面がウィンドウで複雑になることがない。

20

【0004】

【特許文献1】特開平5 - 250126号公報

【発明の開示】

【発明が解決しようとする課題】

【0005】

さて、特許文献1に開示されているシステムにおいては、フォーカスされた時間が長いウィンドウの優先順位が高くなるが、ユーザにとってのアプリケーションの優先順位は、ウィンドウがフォーカスされている時間によらない場合もある。一例をあげると、例えば、パーソナルコンピュータで株取引を行うユーザにとっては、最新の株価情報を得るために、株価情報を取得して更新表示するアプリケーションの優先順位は高いものとなる。しかし、このアプリケーションが、自動で株価情報を取得して更新表示するものであると、フォーカスされて操作されなくとも株価情報を更新表示するためフォーカスされる時間は短くなる。この場合、アプリケーションの優先順位はユーザにとっては優先順位が高いものの、特許文献1に開示されている技術によれば、フォーカスされる時間が少なくなるため、システムの判断においては優先順位が低くされてしまう。すると、ウィンドウの数が所定値以上となった際、ユーザにとって優先度の高いウィンドウであるにもかかわらずアイコン化されてしまい、ユーザにとって不便となる。

30

【0006】

本発明は、上述した背景の下になされたものであり、ウィンドウ数が増えた時にユーザにとって優先度の低いウィンドウを最小化する技術を提供することを目的とする。

40

【課題を解決するための手段】

【0007】

上述した課題を解決するために本発明は、アプリケーションプログラムを記憶した記憶手段と、ユーザにより操作される操作手段と、表示手段に表示されたウィンドウのサイズをユーザが前記操作手段にした操作に応じて変更するサイズ変更手段と、前記サイズ変更手段によりサイズを変更されたウィンドウのサイズ変更回数を記憶する変更回数記憶手段と、前記記憶手段に記憶されたアプリケーションプログラムをユーザが前記操作手段にした操作に応じて実行する実行手段と、前記実行手段により実行されたアプリケーションプ

50

プログラムに対応したウィンドウを表示手段に表示させる表示制御手段であって、前記実行手段がアプリケーションプログラムの実行を始める際に前記表示手段に複数のウィンドウが表示されている場合、前記表示手段に既に表示されている複数のウィンドウと、前記実行手段により実行されるアプリケーションプログラムに対応したウィンドウとを重複させることなく前記表示手段の表示領域内に表示できない場合には、前記表示手段に既に表示されている各ウィンドウのサイズ変更回数を前記変更回数記憶手段から読み出し、読み出したサイズ変更回数が最大であるウィンドウのサイズを最小化して前記表示手段に表示させる表示制御手段とを有する情報処理装置を提供する。

【0008】

この構成においては、前記表示制御手段は、サイズ変更回数が最大であるウィンドウのサイズを最小化しても複数のウィンドウを重複させることなく前記表示手段の表示領域内に表示できない場合には、読み出したサイズ変更回数が大であるウィンドウから順次ウィンドウのサイズを最小化し、一のウィンドウのサイズを最小化する毎に、最小化したウィンドウと、最小化されていないウィンドウと、前記実行手段により実行されるアプリケーションプログラムのウィンドウとを重複させることなく前記表示手段の表示領域内に表示可能であるか判断し、表示可能であると判断した場合には、最小化されたウィンドウと、最小化されていないウィンドウと、前記実行手段により実行されるアプリケーションプログラムのウィンドウとを前記表示手段に表示させてもよい。

【0009】

また、本発明は、アプリケーションプログラムを記憶した記憶手段と、ユーザにより操作される操作手段と、表示手段に表示されたウィンドウのサイズをユーザが前記操作手段にした操作に応じて変更するサイズ変更手段と、前記サイズ変更手段によりサイズを変更されたウィンドウのサイズ変更回数を記憶する変更回数記憶手段と、前記記憶手段に記憶されたアプリケーションプログラムをユーザが前記操作手段にした操作に応じて実行する実行手段と、前記実行手段により実行されたアプリケーションプログラムに対応したウィンドウを表示手段に表示させる表示制御手段であって、前記サイズ変更手段がウィンドウのサイズを変更する際に前記表示手段に複数のウィンドウが表示されている場合、表示されているウィンドウにおいて前記サイズ変更手段によりサイズ変更されないウィンドウと、前記サイズ変更手段によりサイズ変更されるウィンドウとを重複させることなく前記表示手段の表示領域内に表示できない場合には、前記表示手段に表示されているウィンドウにおいて前記サイズ変更手段によりサイズ変更されない各ウィンドウのサイズ変更回数を前記変更回数記憶手段から読み出し、読み出したサイズ変更回数が最大であるウィンドウのサイズを最小化して前記表示手段に表示させる表示制御手段とを有する情報処理装置を提供する。

【0010】

この構成においては、前記表示制御手段は、サイズ変更回数が最大であるウィンドウのサイズを最小化しても複数のウィンドウを重複させることなく前記表示手段の表示領域内に表示できない場合には、読み出したサイズ変更回数が大であるウィンドウから順次ウィンドウのサイズを最小化し、一のウィンドウのサイズを最小化する毎に、最小化したウィンドウと、最小化されていないウィンドウと、前記サイズ変更手段によりサイズ変更されたウィンドウとを重複させることなく前記表示手段の表示領域内に表示可能であるか判断し、表示可能であると判断した場合には、最小化したウィンドウと、最小化されていないウィンドウと、前記サイズ変更手段によりサイズ変更されたウィンドウとを前記表示手段に表示させてもよい。

【0011】

また、上記構成においては、前記表示制御手段は、前記表示手段に表示させる複数のウィンドウが重複しないように各ウィンドウを前記表示手段に表示させてもよい。

【0012】

また、本発明は、コンピュータを、ユーザにより操作される操作手段と、表示手段に表示されたウィンドウのサイズをユーザが前記操作手段にした操作に応じて変更するサイズ

10

20

30

40

50

変更手段と、前記サイズ変更手段によりサイズを変更されたウィンドウのサイズ変更回数を記憶する変更回数記憶手段と、記憶手段に記憶されたアプリケーションプログラムをユーザが前記操作手段にした操作に応じて実行する実行手段と、前記実行手段により実行されたアプリケーションプログラムに対応したウィンドウを表示手段に表示させる表示制御手段であって、前記実行手段がアプリケーションプログラムの実行を始める際に前記表示手段に複数のウィンドウが表示されている場合、前記表示手段に既に表示されている複数のウィンドウと、前記実行手段により実行されるアプリケーションプログラムに対応したウィンドウとを重複させることなく前記表示手段の表示領域内に表示できない場合には、前記表示手段に既に表示されている各ウィンドウのサイズ変更回数を前記変更回数記憶手段から読み出し、読み出したサイズ変更回数が最大であるウィンドウのサイズを最小化して前記表示手段に表示させる表示制御手段として機能させるプログラムを提供する。

10

【 0 0 1 3 】

また、本発明は、コンピュータを、ユーザにより操作される操作手段と、表示手段に表示されたウィンドウのサイズをユーザが前記操作手段にした操作に応じて変更するサイズ変更手段と、前記サイズ変更手段によりサイズを変更されたウィンドウのサイズ変更回数を記憶する変更回数記憶手段と、記憶手段に記憶されたアプリケーションプログラムをユーザが前記操作手段にした操作に応じて実行する実行手段と、前記実行手段により実行されたアプリケーションプログラムに対応したウィンドウを表示手段に表示させる表示制御手段であって、前記サイズ変更手段がウィンドウのサイズを変更する際に前記表示手段に複数のウィンドウが表示されている場合、表示されているウィンドウにおいて前記サイズ変更手段によりサイズ変更されないウィンドウと、前記サイズ変更手段によりサイズ変更されるウィンドウとを重複させることなく前記表示手段の表示領域内に表示できない場合には、前記表示手段に表示されているウィンドウにおいて前記サイズ変更手段によりサイズ変更されない各ウィンドウのサイズ変更回数を前記変更回数記憶手段から読み出し、読み出したサイズ変更回数が最大であるウィンドウのサイズを最小化して前記表示手段に表示させる表示制御手段として機能させるプログラムを提供する。

20

【 発明の効果 】

【 0 0 1 4 】

本発明によれば、ウィンドウ数が多くなった時にユーザにとって優先度の低いウィンドウを最小化することができる。

30

【 発明を実施するための最良の形態 】

【 0 0 1 5 】

〔 実施形態の構成 〕

図 1 は、本発明の実施形態に係る情報処理装置の一例である折畳み型の携帯電話機 1 の外觀図である。図 1 に示したように、この携帯電話機 1 は、第 1 筐体 2 A と第 2 筐体 2 B とを有する。第 1 筐体 2 A と第 2 筐体 2 B は、各々の縁端部近傍においてヒンジ 3 を介して連結されている。この携帯電話機 1 の利用者は、図 1 に示すように両筐体を開いた状態で当該携帯電話機 1 を使用して音声通話やデータ通信を行なう。

【 0 0 1 6 】

第 2 筐体 2 B には、利用者により操作される複数の操作キー群 4 や、ユーザの音声を入力するためのマイクロホン 5 が配設されている。また、第 2 筐体 2 B の内部には、当該携帯電話機 1 の動作を制御する C P U (Central Processing Unit) や C P U の作業エリアとして使用される R A M (Random Access Memory)、C P U により実行される各種プログラムを記憶する不揮発性メモリといった各種の電子部品 (図示略) が収容されている。一方、第 1 筐体 2 A には、音声通話時の音声出力するスピーカ 6、無線基地局との間で無線通信を行なうためのアンテナ 7、文字や画像を表示するディスプレイ装置 8 などが配設されている。

40

【 0 0 1 7 】

図 2 は、携帯電話機 1 のハードウェア構成を示したブロック図である。図 2 に示したように、携帯電話機 1 の各部はバス 1 0 1 により接続されている。通信部 1 0 8 はアンテナ

50

7を備えており、携帯電話機1が行う通信を中継する無線基地局と音声通信やデータ通信を行う。操作部106は、数字や文字の入力を行うためのテンキー、発呼、クリア、終話を指示する各種キーや、メニュー操作を行うためのキー、電源のオン/オフを行うための電源キー等を備えた操作キー群4を有している。携帯電話機1のユーザは、この操作部106に設けられたキーを操作して携帯電話機1へ各種指示を入力する。表示部107は、ディスプレイ装置8を備えており、文字や画像、携帯電話機1を操作するためのメニュー画面等を表示する。

【0018】

R O M (Read Only Memory) 103にはI P L (Initial Program Loader)、オペレーティングシステム(以下、O Sと称する)の機能を実現させるO Sプログラム、J a v a (登録商標)の実行環境を実現するJ a v aプラットフォームプログラムが記憶されている。記憶部105は、図示を省略した不揮発性メモリを有しており、音楽の再生機能を実現させるアプリケーションプログラム(プログラム名「music」)、ゲームの機能を実現させるアプリケーションプログラム(プログラム名「game_1」、「game_2」)などの各種プログラムを記憶している。なお、これらのアプリケーションプログラムは、J a v aプログラミング言語を用いて作成されたプログラム(バイトコード)を含むJ A R (Java Archive)ファイルと、当該J A Rファイルに関するデータを記述したA D F (Application Descriptor File)とからなる。

【0019】

また、記憶部105は、変更履歴テーブルT B 1を記憶している。図3(a)は変更履歴テーブルT B 1のフォーマットを例示した図である。変更履歴テーブルT B 1は、「プログラム名」フィールド、「変更回数」フィールド、「ウィンドウサイズ」フィールドを有しており、「プログラム名」フィールドには、C P U 102が実行したアプリケーションプログラムのプログラム名が格納される。また「変更回数」フィールドの各セルには、同じレコードに格納されているプログラム名のプログラムが実行された時に表示部107に表示されるウィンドウのサイズ変更回数が格納される。また、「ウィンドウサイズ」フィールドの各セルには、同じレコードに格納されているプログラム名のプログラムが実行された時に表示部107に表示されるウィンドウのサイズ(ウィンドウの縦のサイズ×ウィンドウの横のサイズ)が格納される。

【0020】

携帯電話機1においては、図示せぬ電源から電力が供給されるとR O M 103に記憶されているI P LがC P U 102により実行される。C P U 102がI P Lを実行すると、R O M 103に記憶されているO SプログラムがC P U 102により実行され、操作部106からの入力や表示部107への画面出力といった入出力機能、記憶部105へのアクセス、記憶部105に記憶されている各種プログラムの実行、音声通信機能やデータ通信機能など、携帯電話機としての基本的な機能が実現する。

【0021】

また、O Sプログラムが実行されると、携帯電話機1においては、複数のプログラムを並行して実行するマルチタスク機能やG U I (Graphical User Interface)機能が実現する。携帯電話機1は、ユーザからの指示に応じてアプリケーションプログラムを実行すると、ユーザとアプリケーションプログラムとのインターフェースとなるウィンドウを、実行したプログラム毎に表示部107に表示する。そして、ユーザが操作部106を操作することにより、ユーザからの入力引き渡されるウィンドウを切り替えることができ、ウィンドウにフォーカスがある状態においては、ユーザが操作部106を操作することにより、ウィンドウの表示位置や表示サイズを変更することができる。

【0022】

また、C P U 102は、O Sプログラムを実行した後、J a v aプラットフォームプログラムを実行する。すると、図4に例示したJ a v a実行環境114がO S上に実現し、記憶部105の不揮発性メモリには、第1ストレージ115と第2ストレージ116とが確保される。J a v a実行環境114は、クラスライブラリ117、J V M (Java Virtu

10

20

30

40

50

al Machine) 1 1 8 および J A M (Java Application Manager) 1 1 9 からなる。クラスライブラリ 1 1 7 は、特定の機能を有するプログラムモジュール(クラス)群を 1 つのファイルに結合したものである。J V M 1 1 8 は、J a v a アプリケーションプログラムとして提供されるバイトコードを解釈して実行する機能を有する。J A M 1 1 9 は、J a v a アプリケーションプログラムのダウンロードやインストール、起動・終了などを管理する機能を有する。

【 0 0 2 3 】

第 1 ストレージ 1 1 5 は、J A M 1 1 9 の管理の下にダウンロードされる J a v a アプリケーションプログラム(J a r ファイルと A D F)を格納する領域である。第 2 ストレージ 1 1 6 は、J a v a アプリケーションプログラムの実行の際に生成されたデータをその終了後に格納しておくための領域であり、インストールされた J a v a アプリケーションプログラム毎に個別の格納領域が割り当てられるようになっている。そして、ある J a v a アプリケーションプログラムに割り当てられた格納領域のデータは、その J a v a アプリケーションプログラムが実行されている間のみ書き換え可能となっており、別の J a v a アプリケーションプログラムが書き換えを行い得ないようになっている。

10

【 0 0 2 4 】

この J a v a 実行環境 1 1 4 が実現している状態において、ユーザが操作部 1 0 6 を操作して J a v a アプリケーションプログラムの実行を指示すると、第 1 ストレージに格納されている J a v a アプリケーションプログラムが J V M 1 1 8 により解釈されて実行され、プログラムに対応したウィンドウが表示部 1 0 7 に表示される。

20

【 0 0 2 5 】

[実施形態の動作]

(アプリケーションプログラムの初回実行時の動作)

ユーザが操作部 1 0 6 のキーを操作してアプリケーションプログラムの実行を携帯電話機 1 に対して指示すると、C P U 1 0 2 は、実行を指示されたアプリケーションプログラムを実行する。そして C P U 1 0 2 は、アプリケーションプログラムを実行すると、ユーザとアプリケーションプログラムとのインターフェースとなるウィンドウを表示部 1 0 7 に表示するが、ここで図 5 に示した処理を行う。

【 0 0 2 6 】

まず、C P U 1 0 2 は、実行を指示されたアプリケーションプログラムが過去に実行されたことがあるか否かを判断する(ステップ S A 1)。具体的には、C P U 1 0 2 は、実行を指示されたアプリケーションプログラムのプログラム名を変更履歴テーブル T B 1 において検索する。ここで、このプログラム名が「プログラム名」フィールドに格納されていない場合には、C P U 1 0 2 は、実行を指示されたアプリケーションプログラムが過去に実行されていないと判断し(ステップ S A 1; N O)、実行を指示されたアプリケーションプログラムのプログラム名とプログラムの実行により表示されるウィンドウの情報を変更履歴テーブル T B 1 に格納する(ステップ S A 2)。

30

【 0 0 2 7 】

例えば、実行を指示されたプログラム名が「music」である場合、C P U 1 0 2 は、このプログラム名「music」を図 3 (a) に示したように変更履歴テーブル T B 1 の「プログラム名」フィールドに格納し、「music」が格納されたレコードの「変更回数」フィールドのセルには、変更回数として「0」を格納する。そして、予め定められたウィンドウのサイズ(「e x f」)をプログラム名「music」が格納されたレコードの「ウィンドウサイズ」フィールドのセルに格納する。

40

【 0 0 2 8 】

次に C P U 1 0 2 は、表示部 1 0 7 に他のプログラムのウィンドウ(他に実行中のアプリケーションプログラムのウィンドウ)が表示されているか否かを判断する(ステップ S A 3)。ここで、ウィンドウが一つも表示されていない場合(アプリケーションプログラムの実行を指示された時点で他にアプリケーションプログラムを実行していない場合)には、C P U 1 0 2 はステップ S A 3 で N O と判断する。そして、実行を指示されたアプリ

50

ケーションプログラムのウィンドウサイズを変更履歴テーブルTB1から取得する(ステップSA4)。ここでは、CPU102は、プログラム名「music」が格納されているレコードに格納されたウィンドウサイズ「e×f」を変更履歴テーブルTB1から取得する。そして、プログラム名「music」のプログラムのウィンドウサイズを、取得したウィンドウサイズ(e×f)にして表示部107に表示する(ステップSA9)。

【0029】

次に、ユーザが操作部106を操作し、プログラム名「music」のプログラムのウィンドウのサイズ変更を携帯電話機1に対して指示すると、CPU102は、図6に示した処理を行い、ウィンドウのサイズを変更する処理を行う。まず、CPU102は、サイズ変更を指示されたウィンドウの情報を変更履歴テーブルTB1に格納する(ステップSB1)。具体的には、CPU102は、サイズ変更を指示されたウィンドウのプログラム名が格納されているレコードの変更回数に1を加算する。ここでは、プログラム名「music」のプログラムのウィンドウのサイズ変更が指示されたため、CPU102は、変更履歴テーブルTB1のプログラム名「music」のレコードに格納されている変更回数「0」に1を加算する。また、CPU102は、サイズ変更後のウィンドウサイズを変更履歴テーブルTB1に格納する。ここで、プログラム名「music」のプログラムのウィンドウのサイズが「e×f」から「e1×f1」に変更された場合、この変更後のウィンドウサイズ「e1×f1」をプログラム名「music」のレコードの「ウィンドウサイズ」のセルに格納する。この結果、変更履歴テーブルTB1は図3(b)に示したようになる。

【0030】

次にCPU102は、サイズ変更を指示されたウィンドウ以外のウィンドウが表示部107に表示されているか否かを判断する(ステップSB2)。ここで、サイズ変更を指示されたウィンドウ以外には、ウィンドウが一つも表示されていない場合には、CPU102はステップSB2でNOと判断する。そして、サイズ変更を指示されたプログラム名「music」のプログラムのウィンドウサイズを変更履歴テーブルTB1から取得し(ステップSB3)、ウィンドウサイズを取得したサイズ(e1×f1)にして表示部107に表示する(ステップSB8)。

【0031】

(複数のアプリケーション実現時の動作)

次に、アプリケーションプログラムを複数実行した時の動作について説明する。なお、以下の説明においては、図3(b)に示した変更履歴テーブルTB1が記憶部105に記憶されており、アプリケーションプログラムが一つも実行されていない状態からアプリケーションプログラムを順次実行していった場合を想定して動作の説明を行う。

【0032】

まず、ユーザが操作部106のキーを操作してプログラム名「game_1」のアプリケーションプログラムの実行を携帯電話機1に対して指示すると、CPU102は、図5に示した処理を行う。まず、CPU102は、実行を指示されたアプリケーションプログラムが過去に実行されたことがあるか否かを判断するため、プログラム名「game_1」を変更履歴テーブルTB1において検索する。ここで、図3(b)に示した変更履歴テーブルTB1にはプログラム名「game_1」が格納されているため、CPU102によりプログラム名「game_1」が検索され、ステップSA1でYESと判断される。

【0033】

次に、CPU102は、表示部107に他のプログラムのウィンドウが表示されているか否かを判断する(ステップSA3)。ここで、ウィンドウが一つも表示されていないと(ステップSA3; NO)、CPU102は、実行を指示されたアプリケーションプログラムのウィンドウサイズを変更履歴テーブルTB1から取得する(ステップSA4)。

ここでは、CPU102は、プログラム名「game_1」が格納されているレコードに格納されたウィンドウサイズ「a×b」を変更履歴テーブルTB1から取得する。そして、プログラム名「music」のプログラムのウィンドウサイズを、取得したウィンドウサイズ(a×b)にして表示部107に表示する(ステップSA9)。

10

20

30

40

50

【 0 0 3 4 】

次に、ユーザが操作部 1 0 6 のキーを操作してプログラム名「game_2」のアプリケーションプログラムの実行を携帯電話機 1 に対して指示すると、C P U 1 0 2 は、実行を指示されたアプリケーションプログラムのプログラム名「game_2」を変更履歴テーブル T B 1 において検索する。ここで、図 3 (b) に示した変更履歴テーブル T B 1 にはプログラム名「game_2」が格納されているため、ステップ S A 1 で Y E S と判断される。

【 0 0 3 5 】

次に、C P U 1 0 2 は、表示部 1 0 7 に他のプログラムのウィンドウが表示されているか否かを判断する（ステップ S A 3 ）。ここでは、既にプログラム名「game_1」のアプリケーションプログラムが実行されてウィンドウが 1 つ表示されているため、C P U 1 0 2 はステップ S A 3 で Y E S と判断する。C P U 1 0 2 は、ステップ S A 3 で Y E S と判断すると、プログラム名「game_2」が格納されているレコードに格納されたウィンドウサイズ「c × d」を変更履歴テーブル T B 1 から読み出す（ステップ S A 5 ）。

【 0 0 3 6 】

そして、C P U 1 0 2 は、既に表示されているウィンドウと、実行を指示されたプログラムのウィンドウとを重ねずに表示部 1 0 7 に表示させられるか否かを判断する（ステップ S A 6 ）。具体的には、C P U 1 0 2 は、表示部 1 0 7 の表示領域のサイズと同じサイズの仮想表示領域を生成し、この仮想表示領域に各プログラムのウィンドウを配置していく。まず、C P U 1 0 2 は、図 7 に示したようにプログラム名「game_1」のプログラムのウィンドウ W 1 を仮想表示領域に配置する。次に C P U 1 0 2 は、実行を指示されたプログラム名「game_2」のプログラムのウィンドウ W 2 の配置する。ここで、図 7 に示したように、仮想表示領域において、各プログラムのウィンドウが重ならないようにウィンドウを配置できる場合（ステップ S A 6 ; Y E S ）、C P U 1 0 2 は、仮想表示領域の配置と同様にして、プログラム名「game_1」のプログラムのウィンドウと、プログラム名「game_2」のプログラムのウィンドウを表示部 1 0 7 に表示させる（ステップ S A 9 ）。

【 0 0 3 7 】

次に、ユーザが操作部 1 0 6 のキーを操作してプログラム名「music」のアプリケーションプログラムの実行を携帯電話機 1 に対して指示すると、C P U 1 0 2 は、実行を指示されたアプリケーションプログラムのプログラム名「music」を変更履歴テーブル T B 1 において検索する。ここで、図 3 (b) に示した変更履歴テーブル T B 1 にはプログラム名「music」が格納されているため、ステップ S A 1 で Y E S と判断される。

【 0 0 3 8 】

次に、C P U 1 0 2 は、表示部 1 0 7 に他のプログラムのウィンドウが表示されているか否かを判断する（ステップ S A 3 ）。ここでは、既にプログラム名「game_1」のアプリケーションプログラムと、プログラム名「game_2」のアプリケーションプログラムとが実行されてウィンドウが 2 つ表示されているため、C P U 1 0 2 はステップ S A 3 で Y E S と判断する。C P U 1 0 2 は、ステップ S A 3 で Y E S と判断すると、プログラム名「music」が格納されているレコードに格納されたウィンドウサイズ「e 1 × f 1」を変更履歴テーブル T B 1 から読み出す（ステップ S A 5 ）。

【 0 0 3 9 】

そして、C P U 1 0 2 は、既に表示されているウィンドウと、実行を指示されたプログラムのウィンドウとを重ねずに表示部 1 0 7 に表示させられるか否かを判断する（ステップ S A 6 ）。C P U 1 0 2 は、表示部 1 0 7 の表示領域のサイズと同じサイズの仮想表示領域を生成し、この仮想表示領域に各ウィンドウを配置していく。まず、図 8 に示したように、C P U 1 0 2 はプログラム名「game_1」のプログラムのウィンドウ W 1 を仮想表示領域に配置し、次にプログラム名「game_2」のプログラムのウィンドウ W 2 を仮想表示領域に配置する。次に C P U 1 0 2 は、実行を指示されたプログラム名「music」のプログラムのウィンドウ W 3 の配置を試みる。ここで、図 8 に示したように、各プログラムのウィンドウが重ならないように配置できない場合には（仮想表示領域においてウィンドウで重なる部分（図 8 においてハッチングがかけられている部分）が生じる場合）（ステップ

S A 6 ; N O)、C P U 1 0 2 は最小化するウィンドウのプログラムを選択する (ステップ S A 7)。

【 0 0 4 0 】

具体的には、C P U 1 0 2 は、先に表示しているウィンドウのプログラム名「game_1」のレコードとプログラム名「game_2」のレコードに格納されている変更回数を変更履歴テーブル T B 1 から読み出し、実行中のプログラムにおいてウィンドウのサイズ変更回数の多いプログラムを、ウィンドウを最小化するプログラムとして選択する。例えば、図 3 (b) に示した変更履歴テーブル T B 1 の場合、「game_1」のプログラムのウィンドウのサイズ変更回数は「4」、「game_2」のプログラムのウィンドウのサイズ変更回数は「3」であり、プログラム名「game_1」のプログラムのウィンドウサイズ変更回数のほうが多いので、ウィンドウを最小化するプログラムとして、プログラム名が「game_1」であるプログラムが選択される。

10

【 0 0 4 1 】

C P U 1 0 2 は、ウィンドウを最小化するプログラムを選択すると、このプログラムのウィンドウを最小化し、最小化した時のウィンドウサイズ (a 1 × b 1) をプログラム名「game_1」のレコードのウィンドウサイズのセルに格納する (ステップ S A 8)。そして、C P U 1 0 2 は、処理の流れをステップ S A 6 へ戻す。

【 0 0 4 2 】

C P U 1 0 2 は、処理の流れをステップ S A 6 へ戻すと、各プログラムのウィンドウを重ねずに表示部 1 0 7 に表示させられるか否か再度判断する (ステップ S A 6)。まず、C P U 1 0 2 は、上述した処理で最小化されたプログラム名「game_1」のプログラムのウィンドウ W 4 を仮想表示領域に配置し、次にプログラム名「game_2」のプログラムのウィンドウ W 2 を仮想表示領域に配置する。そして、C P U 1 0 2 は、実行を指示されたプログラム名「music」のプログラムのウィンドウ W 3 の配置を試みる。ここで、図 9 に示したように、仮想表示領域において、各プログラムのウィンドウが重ならないように配置できる場合には (ステップ S A 6 ; Y E S)、C P U 1 0 2 は、仮想表示領域の配置と同様にして、各プログラムのウィンドウを表示部 1 0 7 に表示させる (ステップ S A 9)。

20

【 0 0 4 3 】

なお、再度ステップ S A 6 へ処理を戻した時にステップ S A 6 で N O と判断した場合、C P U 1 0 2 は、ウィンドウを最小化するプログラムを選択するが、ここでは、この時点でウィンドウが最小化されていないプログラムのウィンドウのサイズ変更回数のみが変更履歴テーブル T B 1 から読み出される。そして、ウィンドウが最小化されていないプログラムの中でウィンドウのサイズ変更回数が最大のプログラムがウィンドウが最小化されるプログラムとして選択される。

30

【 0 0 4 4 】

このように本実施形態によれば、新たにウィンドウを表示する際にウィンドウが重ならないように配置できない場合には、サイズ変更回数の多いウィンドウがまず最小化される。サイズ変更回数が少ないウィンドウは、そのサイズが適切であるからサイズ変更がされないと考えられ、ウィンドウを整理する際にはそのサイズのまま残しておきたいという要望があるが、本実施形態では、このような要望に応えることができる。

40

【 0 0 4 5 】

(複数ウィンドウ表示時にウィンドウのサイズ変更をした時の動作)

次に、複数のウィンドウが表示されている状態からウィンドウのサイズを変更した時の動作について説明する。なお、以下の説明においては、図 1 0 に示したように表示部 1 0 7 にプログラム名「game_1」のプログラムのウィンドウ W 1 1、プログラム名「game_2」のプログラムのウィンドウ W 2 1、プログラム名「music」のプログラムのウィンドウ W 3 1 が表示されており、図 3 (a) に示した変更履歴テーブル T B 1 が記憶部 1 0 5 に記憶されている状態から、ウィンドウのサイズを変更する時の動作について説明する。

【 0 0 4 6 】

まず、ユーザが操作部 1 0 6 のキーを操作し、プログラム名「music」のプログラムの

50

ウィンドウW 3 1のサイズ変更を携帯電話機 1 に対して指示すると、C P U 1 0 2 は、ウィンドウW 3 1のサイズを変更する。そしてC P U 1 0 2 は、サイズを変更されたウィンドウの情報を変更履歴テーブルT B 1に格納する(ステップS B 1)。具体的には、C P U 1 0 2 は、サイズ変更を指示されたウィンドウのプログラム名が格納されているレコードの変更回数に1を加算する。ここでは、プログラム名「music」のプログラムのウィンドウW 3 1のサイズ変更が指示されたため、C P U 1 0 2 は、変更履歴テーブルT B 1のプログラム名「music」のレコードに格納されている変更回数「0」に1を加算する。また、C P U 1 0 2 は、サイズ変更後のウィンドウサイズを変更履歴テーブルT B 1に格納する。ここで、プログラム名「music」のプログラムのウィンドウW 3 1のサイズが「e × f」から「e 1 × f 1」に変更された場合、この変更後のウィンドウサイズ「e 1 × f 1」をプログラム名「music」のレコードの「ウィンドウサイズ」のセルに格納する。この結果、変更履歴テーブルT B 1は図3 (b) に示したようになる。

【0047】

次にC P U 1 0 2 は、サイズ変更を指示されたウィンドウ以外のウィンドウが表示部 1 0 7 に表示されているか否かを判断する(ステップS B 2)。ここで、サイズ変更を指示されたウィンドウ以外にも、表示部 1 0 7 にはウィンドウが表示されているため、C P U 1 0 2 はステップS B 2でY E S と判断する。そして、サイズ変更を指示されたプログラム名「music」のプログラムのウィンドウサイズ(e 1 × f 1)を変更履歴テーブルT B 1から取得する(ステップS B 4)。

【0048】

次にC P U 1 0 2 は、サイズ変更後のウィンドウW 3 2 と他のウィンドウとを重ねずに表示部 1 0 7 に表示させられるか否かを判断する(ステップS B 5)。具体的には、C P U 1 0 2 は、表示部 1 0 7 の表示領域のサイズと同じサイズの仮想表示領域を生成し、この仮想表示領域に各ウィンドウを配置していく。まず、C P U 1 0 2 は、サイズ変更がされなかったプログラム名「game_1」のプログラムのウィンドウW 1 1 を、仮想表示領域に配置し、次に、サイズ変更がされなかったプログラム名「game_2」のプログラムのウィンドウW 2 1 を仮想表示領域へ配置する。次にC P U 1 0 2 は、プログラム名「music」のプログラムのウィンドウW 3 2 の仮想表示領域への配置を試みる。ここで、図1 1 に示したように、サイズ変更がされたプログラム名「music」のプログラムのウィンドウW 3 2 を配置した場合に、各プログラムのウィンドウが重ならないように配置できない場合には(仮想表示領域においてウィンドウで重なる部分(図1 1 においてハッチングがかけられた部分)が生じる場合)(ステップS B 5 ; N O)、C P U 1 0 2 は最小化するウィンドウのプログラムを選択する(ステップS B 6)。

【0049】

具体的には、C P U 1 0 2 は、サイズ変更がされていないプログラムのプログラム名「game_1」のレコードとプログラム名「game_2」のレコードに格納されている変更回数を変更履歴テーブルT B 1から読み出し、実行中のプログラムにおいてウィンドウサイズの変更回数の多いプログラムを最小化するプログラムとして選択する。例えば、図3 (b) に示した変更履歴テーブルT B 1の場合、「game_1」のプログラムのウィンドウのサイズ変更回数は「4」、「game_2」のプログラムのウィンドウのサイズ変更回数は「3」であり、プログラム名「game_1」のプログラムのウィンドウサイズ変更回数のほうが多いので、ウィンドウを最小化するプログラムとして、プログラム名が「game_1」であるプログラムが選択される。

【0050】

C P U 1 0 2 は、ウィンドウを最小化するプログラムを選択すると、このプログラムのウィンドウを最小化し、最小化した時のウィンドウサイズ「a 1 × b 1」をプログラム名「game_1」のレコードのウィンドウサイズのセルに格納する(ステップS B 7)。そして、C P U 1 0 2 は処理の流れをステップS B 5 へ戻す。

【0051】

C P U 1 0 2 は、処理の流れをステップS B 5 へ戻すと、各プログラムのウィンドウを

重ねずに表示部 107 に表示させられるか否か再度判断する（ステップ S B 5）。まず、C P U 102 は、図 12 に示したように、上述した処理でウィンドウが最小化されたプログラム名「game_1」のプログラムウィンドウ W 12 を仮想表示領域に配置し、次にプログラム名「game_2」のプログラムのウィンドウ W 2 1 を仮想表示領域に配置する。そして、C P U 102 は、プログラム名「music」のプログラムのウィンドウ W 3 2 の配置を試みる。ここで、図 12 に示したように、仮想表示領域において、各プログラムのウィンドウが重ならないように配置できる場合には（ステップ S B 5；Y E S）、C P U 102 は、仮想表示領域の配置と同様にして、各プログラムのウィンドウを表示部 107 に表示させる（ステップ S B 8）。

【0052】

10

なお、再度ステップ S B 5 へ処理を戻した時にステップ S B 5 で N O と判断した場合、C P U 102 は、ウィンドウを最小化するプログラムを選択するが、ここでは、この時点でウィンドウが最小化されていないプログラムのウィンドウのサイズ変更回数のみが変更履歴テーブル T B 1 から読み出される。そして、ウィンドウが最小化されていないプログラムの中でウィンドウのサイズ変更回数が最大のプログラムがウィンドウが最小化されるプログラムとして選択される。

【0053】

以上説明したように本実施形態によれば、複数のウィンドウが表示されている状態においてウィンドウのサイズが変更された時に、ウィンドウが重ならないように配置できない場合には、サイズ変更回数の多いウィンドウがまず最小化される。サイズ変更回数が少ないウィンドウは、そのサイズが適切であるからサイズ変更がされないと考えられ、ウィンドウを整理する際には、そのサイズのまま残しておきたいという要望があるが、本実施形態では、このような要望に応えることができる。

20

【0054】

[変形例]

以上、本発明の実施形態について説明したが、本発明は上述した実施形態に限定されることなく、他の様々な形態で実施可能である。例えば、上述の実施形態を以下のように変形して本発明を実施してもよい。

【0055】

上述した動作を行う情報処理装置は、パーソナルコンピュータ装置や、P D A（Personal Digital Assistants）など、複数のプログラムを並行して実行し、プログラム毎にユーザとのインターフェースとなるウィンドウを表示するものであれば、携帯電話機に限定されず、他のコンピュータ装置であってもよい。

30

【0056】

上述した実施形態においては、ステップ S A 8 の処理が終了した後、ステップ S A 6 に処理の流れを戻すのではなく、ステップ S A 9 へ処理の流れを変更するようにしてもよい。また、上述した実施形態においては、ステップ S B 7 の処理が終了した後、ステップ S B 5 に処理の流れを戻すのではなく、ステップ S B 8 へ処理の流れを変更するようにしてもよい。

【0057】

40

上述した実施形態においては、サイズの変更回数の多いウィンドウが最小化されているが、最も小さい所定のサイズに最小化するのではなく、ウィンドウをアイコン（プログラムの機能を表わす絵柄）化して表示するようにしてもよい。

【図面の簡単な説明】

【0058】

【図 1】本発明の一実施形態に係る携帯電話機の外觀図である。

【図 2】同携帯電話機のハードウェア構成を示したブロック図である。

【図 3】同携帯電話機が記憶している変更履歴テーブル T B 1 のフォーマットを示した図である。

【図 4】携帯電話機において実現する J a v a 実行環境の構成を示した図である。

50

【図 5】CPU 102 が行う処理の流れを示したフローチャートである。

【図 6】CPU 102 が行う処理の流れを示したフローチャートである。

【図 7】仮想表示領域におけるウィンドウの配置を示した図である。

【図 8】仮想表示領域におけるウィンドウの配置を示した図である。

【図 9】仮想表示領域におけるウィンドウの配置を示した図である。

【図 10】表示部 107 におけるウィンドウの配置例を示した図である。

【図 11】仮想表示領域におけるウィンドウの配置を示した図である。

【図 12】仮想表示領域におけるウィンドウの配置を示した図である。

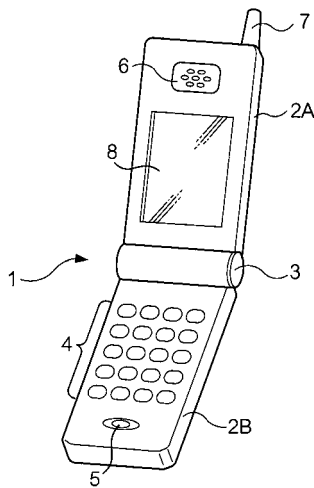
【符号の説明】

【0059】

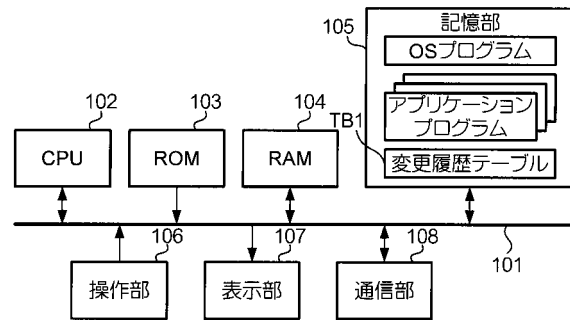
10

1・・・携帯電話機、2A・・・第1筐体、2B・・・第2筐体、3・・・ヒンジ、4・・・操作キー群、5・・・マイクロホン、6・・・スピーカ、7・・・アンテナ、8・・・ディスプレイ装置、101・・・バス、102・・・CPU、103・・・ROM、104・・・RAM、105・・・記憶部、106・・・操作部、107・・・表示部、108・・・通信部

【図 1】



【図 2】



【図 3】

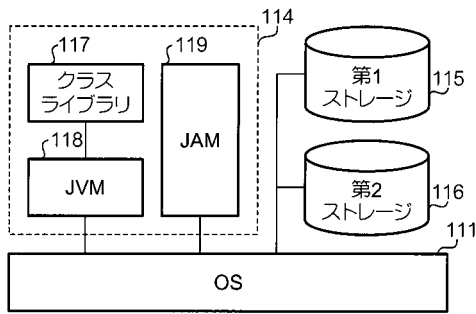
(a)

プログラム名	変更回数	ウィンドウサイズ (縦サイズ×横サイズ)
game_1	4	a×b
game_2	2	c×d
music	0	e×f

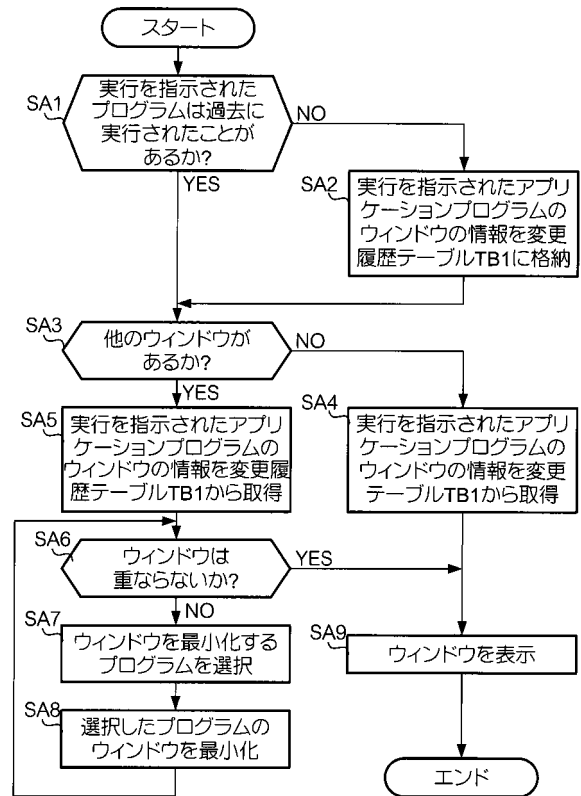
(b)

プログラム名	変更回数	ウィンドウサイズ (縦サイズ×横サイズ)
game_1	4	a×b
game_2	2	c×d
music	1	e1×f1

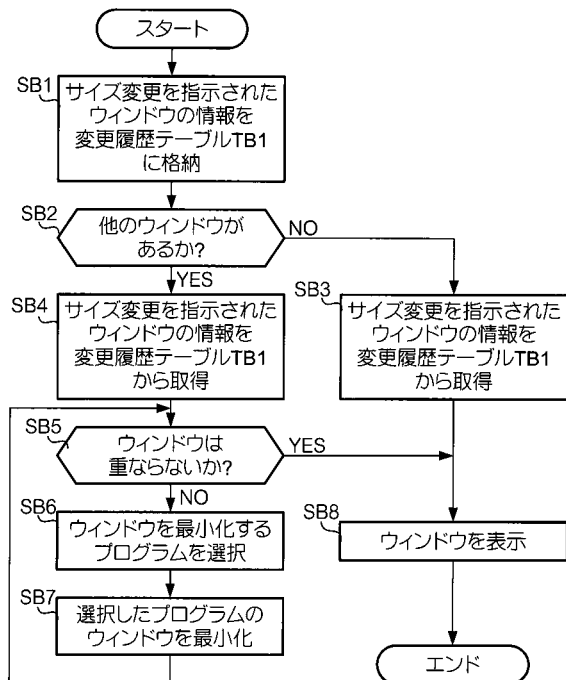
【図 4】



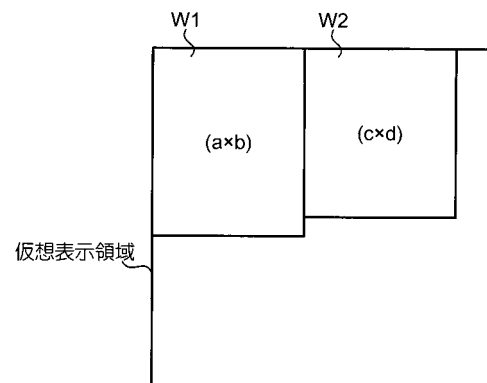
【図 5】



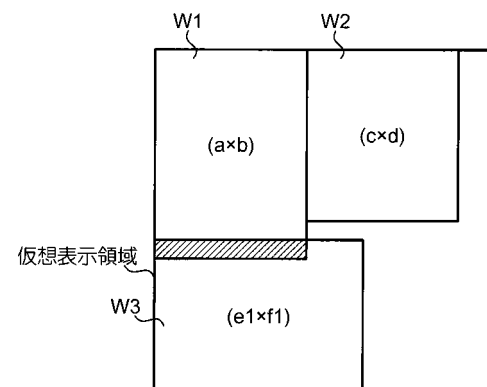
【図 6】



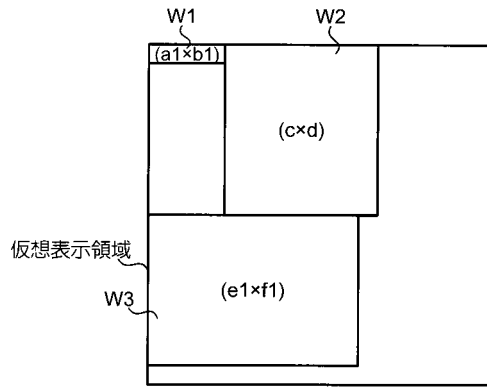
【図 7】



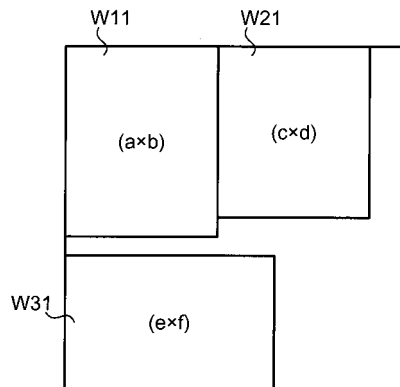
【図 8】



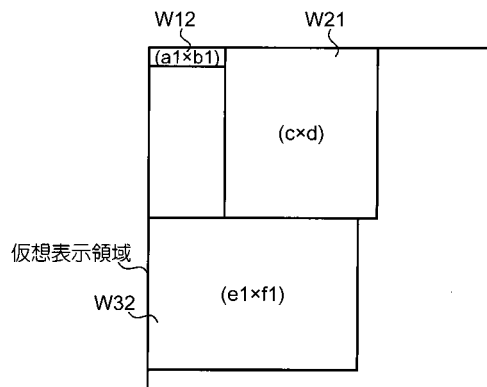
【図 9】



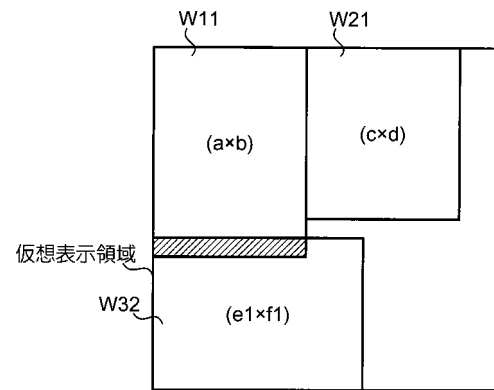
【図 10】



【図 12】



【図 11】



フロントページの続き

(51)Int.Cl. F I テーマコード(参考)
G 0 9 G 5/36 5 2 0 E

(72)発明者 村上 圭一
東京都千代田区永田町二丁目 1 1 番 1 号 株式会社エヌ・ティ・ティ・ドコモ内

(72)発明者 恩田 靖
東京都千代田区永田町二丁目 1 1 番 1 号 株式会社エヌ・ティ・ティ・ドコモ内

(72)発明者 山田 和宏
東京都千代田区永田町二丁目 1 1 番 1 号 株式会社エヌ・ティ・ティ・ドコモ内

(72)発明者 神谷 大
東京都千代田区永田町二丁目 1 1 番 1 号 株式会社エヌ・ティ・ティ・ドコモ内

F ターム(参考) 5C082 AA21 BB25 CA32 CA52 CA62 CA76 CA81 CB01 CB06 DA87

MM10

5E501 AA04 AB03 FA06 FB04