



(19)中華民國智慧財產局

(12)發明說明書公告本

(11)證書號數：TW I494795 B

(45)公告日：中華民國 104 (2015) 年 08 月 01 日

(21)申請案號：098133607

(22)申請日：中華民國 98 (2009) 年 10 月 02 日

(51)Int. Cl. : G06F3/033 (2013.01)

G06F3/048 (2013.01)

(30)優先權：2008/10/26 美國

12/258,439

(71)申請人：微軟公司(美國) MICROSOFT CORPORATION (US)

美國

(72)發明人：湯森黎德 L TOWNSEND, REED L. (US)；涂霄 TU, XIAO (CN)；史考特布萊恩 D SCOTT, BRYAN D. (US)；多爾賽德透德 A TORSET, TODD A. (US)；塞可斯肯尼斯 W SYKES, KENNETH W. (US)；伯拉罕塞米爾 S PRADHAN, SAMIR S. (CA)；悌德珍妮佛 A TEED, JENNIFER A. (US)

(74)代理人：蔡坤財；李世章

(56)參考文獻：

TW M341271

TW 200825886A

CN 100374998C

US 20080042986A1

US 2007/0152984A1

US 2007/0291009A1

WO 2006/020305A2

審查人員：林彥廷

申請專利範圍項數：20 項 圖式數：7 共 46 頁

(54)名稱

多點觸控物件慣性模擬

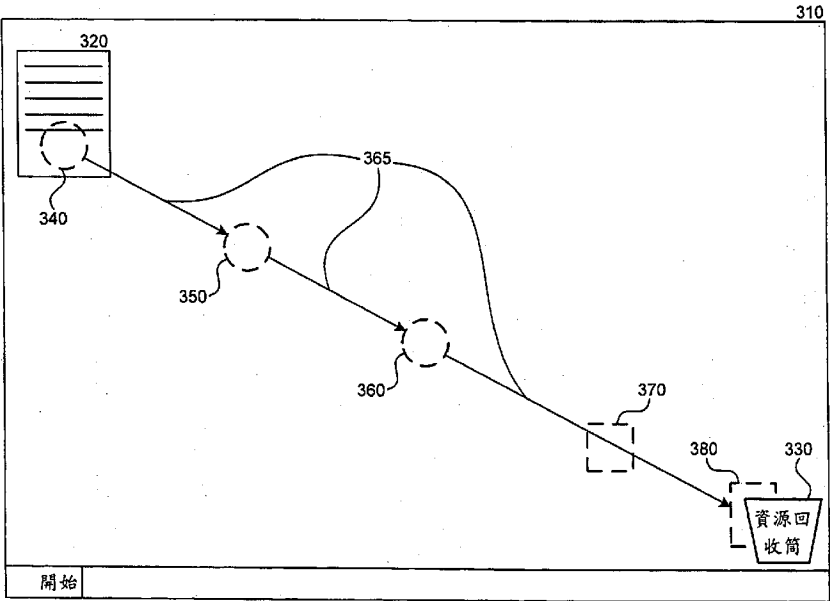
MULTI-TOUCH OBJECT INERTIA SIMULATION

(57)摘要

該慣性系統提供一種共用平台及做為應用之應用程式化介面(Application-programming interface, API)，以延伸自多種多點觸控硬體裝置接收的該輸入而模擬應用物件之真實行為。為了自然地移動，應用物件應呈現物理特性，像是彈性及減速度。當一使用者提高來自一物件的所有接觸(contact)時，該慣性系統提供額外的操縱事件到該應用，如此該應用可以如同該使用者仍在利用觸碰來移動該物件，處理該等事件。該慣性系統基於對該等物件之行為的一模擬而產生該等事件。如果該使用者移動一物件到另一物件當中，該慣性系統模擬該等物件之邊界特性。因此，該慣性系統提供對於使用多點觸控硬體操縱的應用物件之更真實的移動，且該 API 提供橫跨各種應用之操縱的一致性感覺。

The inertia system provides a common platform and application-programming interface (API) for applications to extend the input received from various multi-touch hardware devices to simulate real-world behavior of application objects. To move naturally, application objects should exhibit physical characteristics such as elasticity and deceleration. When a user lifts all contacts from an object, the inertia system provides additional manipulation events to the application so that the application can handle the events as if the user was still moving the object with touch. The inertia system generates the events based on a simulation of the behavior of the objects. If the user moves an object into another object, the inertia system simulates the boundary characteristics of the objects. Thus, the inertia system provides more realistic

movement for application objects manipulated using multi-touch hardware and the API provides a consistent feel to manipulations across applications.



- 310 . . . 顯示器
- 320 . . . 文件物件
- 330 . . . 資源回收筒
- 340 . . . 第一操縱位置
- 350 . . . 第二操縱位置
- 360 . . . 第三操縱位置
- 365 . . . 箭頭
- 370 . . . 第一慣性式操縱位置
- 380 . . . 第二慣性式操縱位置

第 3 圖

發明專利說明書

(本說明書格式、順序，請勿任意更動，※記號部分請勿填寫；惟已有申請案號者請填寫)

※ 申請案號：98133607

※ 申請日期：2009 年 10 月 2 日

※IPC 分類：

G06F 3/033

(2006.01)

G06F 3/048

(2006.01)

一、發明名稱：(中文/英文)

多點觸控物件慣性模擬/ Multi-Touch Object Inertia Simulation

二、中文發明摘要：

該慣性系統提供一種共用平台及做為應用之應用程式化介面(Application-programming interface, API)，以延伸自多種多點觸控硬體裝置接收的該輸入而模擬應用物件之真實行為。為了自然地移動，應用物件應呈現物理特性，像是彈性及減速度。當一使用者提高來自一物件的所有接觸(contact)時，該慣性系統提供額外的操縱事件到該應用，如此該應用可以如同該使用者仍在利用觸碰來移動該物件，處理該等事件。該慣性系統基於對該等物件之行為的一模擬而產生該等事件。如果該使用者移動一物件到另一物件當中，該慣性系統模擬該等物件之邊界特性。因此，該慣性系統提供對於使用多點觸控硬體操縱的應用物件之更真實的移動，且該 API 提供橫跨各種應用之操縱的一致性感覺。

三、英文發明摘要：

The inertia system provides a common platform and application-programming interface (API) for applications to extend the input received from various multi-touch hardware devices to simulate real-world behavior of application

objects. To move naturally, application objects should exhibit physical characteristics such as elasticity and deceleration. When a user lifts all contacts from an object, the inertia system provides additional manipulation events to the application so that the application can handle the events as if the user was still moving the object with touch. The inertia system generates the events based on a simulation of the behavior of the objects. If the user moves an object into another object, the inertia system simulates the boundary characteristics of the objects. Thus, the inertia system provides more realistic movement for application objects manipulated using multi-touch hardware and the API provides a consistent feel to manipulations across applications.

四、指定代表圖：

(一)本案指定代表圖為：第(3)圖。

(二)本代表圖之元件符號簡單說明：

310 顯示器

320 文件物件

330 資源回收筒

340 第一操縱位置

350 第二操縱位置

360 第三操縱位置

365 箭頭

370 第一慣性式操縱位置

380 第二慣性式操縱位置

五、本案若有化學式時，請揭示最能顯示發明特徵的化學式：

無

六、發明說明：

【發明所屬之技術領域】

本發明係關於多點觸控物件慣性模擬。

【先前技術】

一平板型 PC 或筆型電腦為一種筆記型或板狀行動電腦，其裝設有一觸控螢幕或繪圖板/螢幕複合式技術，而允許使用者利用觸控筆、數位筆、或指尖而非鍵盤或滑鼠來操作該電腦。平板型 PC 提供更為自然的輸入型式，因為速寫及手寫遠比鍵盤及滑鼠為更熟悉的輸入型式，特別是對於電腦新手而言。平板型 PC 亦為更容易使用，因為那些身體上無法打字的人即可利用平板型 PC 的額外特徵而能夠與電子世界進行互動。

多點觸控(Multi-touch 或寫為 multitouch)代表一組互動技術，其允許電腦使用者使用多指或輸入裝置(例如觸控筆)來控制繪圖應用。多點觸控實施通常包含觸控硬體(例如螢幕、平板、壁面等等)及辨識多點同時觸控點之軟體。多點觸控係相對於一次僅辨識一觸控點之傳統型觸控螢幕(例如電腦觸控板、ATM、購物互動資訊站)。多點觸控硬體可使用熱量、手指壓力、高補捉率相機、紅外光、光學補捉、可調整電磁感應、超音波接收器、

傳感器麥克風、雷射測距儀、陰影捕捉及其它機制來感測觸碰。目前已經存在有許多多點觸控介面的應用程式，而應用程式設計師與使用者提出更多的建議。一些使用係個人化的(例如 Microsoft Surface、Apple iPhone、HTC Diamond)。做為一種新的輸入方法，多點觸控提供了新的使用者經驗範例的可能性。

一種應用不能夠使用多點觸控硬體，其不具有用於自該多點觸控硬體接收資訊的應用軟體之一介面。可惜地是，每個多點觸控硬體裝置包含其本身的專屬介面，且應用程式開發者必須具有硬體裝置之特定知識來撰寫配合該裝置使用之軟體。例如，一多點觸控硬體供應商可以提供一核心模式驅動器及一使用者模式應用程式介面，藉此使用者模式軟體應用能夠與該多點觸控硬體通訊來接收觸碰資訊。一應用開發者撰寫與該使用者模式應用程式介面通訊的軟體，但該應用開發者之軟體僅配合於該多點觸控硬體運作。利用不同多點觸控硬體裝置之電腦使用者不能夠使用該應用開發者的軟體，除非該應用開發者生產該軟體的一不同版本來正確地與該電腦使用者的裝置配合操作。此即對於應用開發者造成非常有限的可能性市場，降低了撰寫支援多點觸控互動之應用程式的誘因，並使得那些易於取得應用程式之最為常用裝置成本居高不下。

另一個問題為這些應用很難基於自多點觸控硬體所接收的觸碰輸入來判定使用者的意圖。觸碰輸入可被接收成一座標清單，其為該硬體在任何給定時間所感測到的觸碰輸入。每個應用必須包含軟體來解譯該等座標並判定該使用者的意圖。此外，使用者的意圖可能超越所接收到的實際觸碰輸入。使用者可能期待虛擬物件可以如同其在現實世界中一樣運轉。例如，一使用者可能期待能夠藉由輕彈他/她的手指而由桌面的一側「拋」一個檔案到另一側。這種移動並未由目前的多點觸控應用程式所支援，其將可期待該使用者自該螢幕的一側拖曳他/她的手指直到另一側。即使一應用提供此種移動的支援，其他應用無法受惠於此，因此應用開發者必須重複第一應用開發者的工作以於他們的應用中提供相同的功能。

【發明內容】

慣性系統提供一種共用平台及做為應用之應用程式化介面(Application-programming interface, API)，以延伸自多種多點觸控硬體裝置接收的輸入而模擬物件之真實行為。由該應用程式接收的操縱基於接觸該多點觸控硬體之移動而僅描述了一物件的移動。但是，為了自然地移動，物件亦必須呈現物理特性，例如彈性及減速度。當一使用者提高所有與物件的接觸(contact)時，該慣

性系統提供額外的操縱事件到該應用程式，如此該應用程式可以如同使用者仍在利用觸碰來移動該物件，處理該等事件。但是，該慣性系統實際上基於模擬物件之行為而產生事件。如果該使用者移動一物件到另一物件當中，該慣性系統基於該等物件之邊界特性而傳送操縱事件。因此，慣性系統對於使用者使用多點觸控硬體操縱的應用物件提供更為真實的移動，且 API 提供橫跨應用之操縱的一致性感覺。

此「發明內容」係以一簡化型式來做一選擇性觀念之介紹，其在以下的「實施方式」中會進一步加以說明。此「發明內容」並無意於識別出所主張申請標的之關鍵特徵或基本特徵，也並無意於用以限制所主張申請標的之範疇。

【實施方式】

慣性系統提供一種共用平台及做為應用之 API，以延伸自多種多點觸控硬體裝置接收的輸入而模擬物件之真實行為。例如，真實物件在當使用者停止推動它們時基本上並不會停止移動，而會呈現某種慣性及保持移動，直到磨擦力放慢它們，最後才停止。在一些具體實施例中，觸碰輸入首先經由一程序來解譯一或多個做為操縱之接觸的移動。操縱會比個別觸碰輸入更為直接地對映

到使用者意圖，並使用多點觸控接觸加入物件之基本轉換的支援。應用程式能夠使用操縱而同時支援旋轉、改變大小及平移多個物件(例如相片)。該等操縱可被描述為含有旋轉、縮放(例如放大縮小)及平移(例如搖攝(pan))資訊的二維(two-dimensional, 2D)仿射轉換。

該多點觸控硬體之每次觸碰稱之為接觸。例如，當使用者放置他/她的手指在該多點觸控硬體上時，在周圍移動他/她的手指及提高他/她的手指時，該系列的事件為單一接觸。例如，如果使用者移動兩個接觸使其更為靠近或更為遠離，該系統可以判定該使用者正在調整(例如放大或縮小)一物件。在另一例中，如果使用者以圓形運動之方式移動多個接觸，則該系統可解譯該移動為一物件的旋轉。每個應用程式可定義不同相關的物件，所以其係根據該應用程式來附加該系統的實體(稱之為操縱處理器)到每個物件，其可由使用者在該應用程式內使用觸控輸入來操縱。例如，相片瀏覽應用程式可以附加一操縱處理器到每張顯示的相片，如此該使用者可四處移動相片、縮放相片、旋轉相片等等。

由該應用程式所處理的操縱基於該接觸的移動而僅描述了一物件的移動。但是，為了自然地移動，物件亦必須呈現物理特性，例如彈性及減速度。當使用者提高所有與物件的接觸時，該慣性系統提供額外的操縱事件到

應用程式，如此該應用程式可以如同使用者仍在利用觸碰來移動該物件，處理該等事件。但是，慣性系統實際上基於模擬物件之行為而產生該等事件。例如，如果使用者提高接觸而該物件在一特定方向上具有一速度時，則慣性系統持續傳送指示該物件正在該方向上移動的事件，而當物件減速時則會隨時間改變而減慢。如果使用者移動一物件到另一物件當中，像是螢幕的邊緣，該慣性系統基於該等物件之邊界特性而傳送操縱事件。例如，如果一應用開發者定義兩個物件為有彈性，則該等兩個物件在當使用者移動該等物件到另一個當中時可能會彼此彈開。因此，慣性系統對於使用者使用多點觸控硬體操縱的應用物件提供更為真實的移動，且 API 提供橫跨應用之操縱的一致性感覺。

第 1 圖為在一具體實施例中，說明該慣性系統之組件的方塊圖。慣性系統 100 包含一硬體介面 110、一或多個操縱處理器 120、一輸入轉換組件 130、一模擬組件 140，及一應用程式介面 150。這些組件的每個將在本文更為詳細地描述。

硬體介面 110 與該硬體通訊來接收觸碰接觸及移動。硬體介面 110 可以提供數種子組件，其一起運作來提供觸碰輸入資訊。例如，作業系統可以提供多點觸控硬體製造商一共用驅動器模型，以提供它們的特定硬體之觸

碰資訊。該作業系統可解譯經由此模型接收的觸碰資訊成為視窗訊息(例如本文所述的 WM_TOUCH)，並傳遞這些訊息到該應用。因此，硬體介面 110 可以涉及硬體、硬體驅動器及作業系統層的協調。該結果為一系列傳至該慣性系統之訊息，其可識別一特定接觸(例如手指觸碰)，及不同時間該接觸的座標。例如，作業系統可提供當在多點觸控硬體上記下一新接觸的訊息，每次該接觸移動時的一訊息，及當該接觸自多點觸控硬體移開時的一訊息。

一或多個操縱處理器 120 使用輸入轉換組件 130 來解譯相關於一特定應用物件之每個接觸的移動。操縱處理器 120 可以判定使用者正在使用多個接觸來執行一單一動作。例如，使用者可用一隻手的所有五隻手指觸碰一相片，並扭轉他/她的手來指示要旋轉該相片的意圖。操縱處理器 120 接收五個個別接觸(每隻手指一個)及當該使用者旋轉他/她的手時之每個接觸的座標變化。操縱處理器 120 判定每個接觸正在抓取相同的物件並正執行相同的旋轉。該系統將通知該應用程式使用者旋轉該物件，但該應用程式可忽略該使用者係使用兩個、五個或任何特定數目的手指或其它接觸來執行該旋轉。因為該應用程式開發者可以處理那些關於該應用程式之操縱種類，且將其留給慣性系統來解譯自該多點觸控硬體接收

的每一個低階觸控輸入之意義，此即可大為簡化該應用之編寫。

操縱處理器 120 使用輸入轉換組件 130 來做出關於單獨或共同之多個接觸的已接收移動的意義之判定。例如，如果使用者正在利用兩隻手指操縱一相片，其即產生兩個相對應輸入接觸，則操縱處理器 120 使用輸入轉換組件 130 來判定該等兩個接觸之間相對移動的意義。如果該等兩個接觸為互相分開，則輸入轉換組件 130 可以判定使用者正在縮放物件來改變該物件的大小。如果該等兩個接觸旋轉，則輸入轉換組件 130 可以判定使用者正在旋轉物件。如果該等兩個接觸皆在一特定方向上滑動，則輸入轉換組件 130 可以判定使用者正在平移物件到一新的位置。雖然每種移動被單獨地討論，請注意使用者可以同時做出所有三種移動，且該輸入轉換處理器可以回報整體的轉換到該應用。例如，使用者可在一個動作中同時旋轉、縮放及平移一物件。

模擬組件 140 基於對於物件定義的初始化參數及限制，在該使用者停止觸碰該物件之後即模擬一應用物件之持續移動。一應用程式可以利用相關於物件之操縱處理器 120 的最後狀態來初始化模擬組件 140。該應用程式亦可以定義該物件之多種特性，像是該物件的邊界必須如何運作。模擬組件 140 使用基於物理的技術來模擬

在該使用者釋放該物件之後的一段時間該物件之行為。例如，當該使用者正在移動該物件時，模擬組件 140 可用與該應用程式接收的操縱事件相同的型式來繼續發出通知至該應用程式。則該應用程式可聚焦於對物件移動的反應，而非考慮是哪些動作(使用者或物理動作)造成該物件的移動。本技術專業人士將可瞭解到有多種已為吾人所熟知的技術來模擬軟體中虛擬物件之等效物理行為。

應用程式介面 150 與該應用程式進行通訊來接收資訊，並提供操縱轉換到該應用程式。應用程式介面 150 自該應用接收初始化資訊。該初始化資訊可以指定該應用物件對於一特定物件的支援之哪種轉換，以及相關的操縱處理器，還有當使用者不再移動該物件時模擬組件 140 之初始化資料。例如，一些應用物件可以支援縮放，但非旋轉。初始化資訊亦可指定物件的一樞轉點 (pivot point)。該慣性系統經由應用程式介面提供操縱轉換到該應用程式。例如，當該慣性系統接收該系統解譯成一辨識的轉換(例如旋轉)之低階觸碰輸入，該系統發出一事件來通知該應用程式關於該操縱。該應用程式處理該操縱轉換以基於轉換來修正該物件。例如，如果使用者旋轉物件，則該應用程式可以儲存該物件的新定位以於下次應用程式顯示該物件時使用。在另一例中，如果物

件基於模擬組件 140 之計算在該使用者將其釋放之後持續地旋轉，則該應用程式可以儲存該物件的新定位。

在該系統上被實施的運算裝置可以包括一中央處理器、記憶體、輸入裝置(例如鍵盤及指向裝置)、輸出裝置(例如顯示裝置)及儲存裝置(例如磁碟機)。該記憶體及儲存裝置為電腦可讀取媒體，可以實施該系統之電腦可執行指令來進行編碼，其代表含有該等指令之電腦可讀取媒體。此外，資料結構與訊息結構可被儲存或藉由一資料傳輸媒體傳送，像是在一通訊鏈結上的一信號。可使用多種通訊鏈結，像是網際網路、區域網路、廣域網路、點對點撥接連線、行動電話網路等等。

該系統之具體實施例可實施在多種操作環境中，其中包含個人電腦、伺服器電腦、掌上型或膝上型裝置、多處理器系統、微處理器式系統、可程式化消費性電子產品、數位相機、網路 PC、微型電腦、大型電腦、包含任何上述系統或裝置的分散式運算環境等等。該等電腦系統可為行動電話、個人數位助理、智慧型電話、個人電腦、可程式化消費性電子產品、數位相機等等。

該系統可在由一或多部電腦或其它裝置所執行之電腦可執行指令(像是程式模組)的一般性內容中描述。概言之，程式模組包含例式、程式、物件、組件、資料結構等等，其可執行特定工作或實施特定的抽象資料類型。

基本上，該等程式模組的功能在多種具體實施例中可視需要組合或分散。

第 2 圖為在一具體實施例中，用以說明慣性系統之典型操作環境及組件之間的資料流動之資料流程圖。一多點觸控硬體裝置經由一硬體介面產生輸入 210。例如，該硬體可以經由硬體製造商所提供的一軟體驅動器來傳送輸入 210 到一作業系統。該硬體介面提供輸入事件 220 到一應用程式 230。例如，一應用程式可以通知作業系統該應用程式 230 支援多點觸控使用者輸入，並註冊以接收關於多點觸控使用者輸入的訊息。應用程式 230 接收低階觸碰輸入資訊做為輸入變化 240，並轉送輸入變化 240 到一操縱系統 250。例如，該輸入變化 240 可以使用指示每個接觸的目前位置與其他移動特性之一組座標來描述與該硬體之一或多個觸碰接觸之每個移動。操縱系統 250 解譯輸入變化 240，並通知一或多個操縱事件 260 之應用程式 230 來指示該使用者正在對一顯示的物件所執行的較高階操縱。例如，如果該等接觸之移動指示該使用者想要旋轉物件，操縱事件 260 指示旋轉的角度。

當使用者完成移動該物件(例如當該應用程式接收到每個觸碰一物件之接觸已經自該觸碰硬體上移除的通知)時，應用程式 230 傳送初始化資訊 270 到慣性系統 280。

慣性系統 280 判定該物件的下一位置，並提供類似於操縱事件 260 之慣性事件 290，其為操縱系統 250 當使用者正在移動該物件時所提供者。應用程式 230 亦提供一驅動計時器來定期地呼叫慣性系統 280 來經由慣性事件 290 提供該物件的下一位置。應用程式 230 以類似於操縱事件的方式處理該等慣性事件。

雖然該圖式說明該應用程式首先接收觸碰輸入，並傳送該觸碰輸入到操縱系統與慣性，在一些具體實施例中，這些系統直接自硬體介面接收觸碰輸入，解譯該觸碰輸入，並提供解譯的操縱事件到該應用程式。同樣地，該應用程式可能不知道獨立慣性系統 280 在一使用者停止利用觸碰移動一物件之後提供慣性事件，而是有可能在該使用者正在移動該物件期間及當該物件基於慣性正在移動之後，自一介面接收事件。此代表一種替代架構，其提供類似結果功能，但對於該應用程式針對輸入之處理給予較少控制。例如，該應用程式可能無法定義該系統附加個別操縱處理器之個別的應用程式物件。本文所述的 RTS 嵌入程式為該系統之此替代架構之一示例。

第 3 圖為在一具體實施例中，用以說明由使用者觸碰所操縱的應用物件之示意圖。一應用程式可對於許多這種物件同時顯示及接收觸碰輸入。例如，作業系統層應用可以顯示儲存在該使用者之電腦桌面上的一或多個文

件物件。在顯示器 310 中，文件物件 320 在使用者的桌面上代表該使用者想要使用觸碰拖曳到資源回收筒 330 之文件。該使用者執行文件物件 320 之輕彈動作，使得系統處理一第一操縱位置 340、第二操縱位置 350 及第三操縱位置 360。當該使用者剛開始利用一或多隻手指(即接觸)觸碰文件物件 320 時，該應用程式接收第一操縱位置 340。當使用者滑動他/她的手指跨過螢幕時，該應用程式接收第二操縱位置 350。當使用者提起他/她的手指離開螢幕時，該應用程式接收第三操縱位置。箭頭 365 代表文件物件 320 的移動向量。

若沒有慣性，文件物件 320 將停止在第三操縱位置 360，其有可能並非該使用者想要的。該慣性系統提供額外的操縱位置到該應用程式，如同基於當使用者釋放文件物件 320 時文件物件 320 的速度，而該使用者仍然觸碰及移動文件物件 320。當該應用程式初始化慣性系統及首次呼叫該慣性系統的處理功能時，該應用程式接收第一慣性式操縱位置 370。當該應用程式繼續呼叫慣性系統的處理功能時，該應用程式接收第二慣性式操縱位置 380。因為文件物件 320 之最終操縱位置 380 在資源回收筒 330 之上，該應用程式處理該等兩個物件之間的接觸(例如藉由放置文件物件 320 在資源回收筒 330 中)。在以圖示說明的示例中，雖然慣性系統使文件物件

320 之移動減速，文件物件 320 基於該使用者在開始時，文件物件 320 之使用者移動的高初始速度而仍能夠移動橫跨顯示器 310 一段相當的距離。

第 4 圖為在一具體實施例中，用以說明使用慣性系統來處理操縱事件的一多點觸控應用程式之輸入迴圈處理的流程圖。在方塊 410 中，該應用程式接收低階觸碰輸入。例如，作業系統或慣性系統之實體自多點觸控硬體接收觸碰接觸資訊，並轉送該觸碰接觸資訊到該應用程式。在方塊 420 中，該應用程式識別該輸入所套用到的物件。例如，該應用程式可以藉由比較該等座標與由該應用程式顯示的每個應用物件之座標而命中測試(hit test)該接收到的輸入之座標。如果該觸碰輸入在一已顯示的應用物件之邊界內，則該應用程式判定該觸碰輸入套用到該物件。在方塊 430 中，該應用程式傳送該接收到的觸碰輸入及關於該識別的應用物件之資訊到一操縱 API 來引動慣性系統(參見第 5 圖)。例如，該應用程式可以產生每個應用物件之數值識別，並在每次觸碰輸入對應於該物件時傳送該數值識別到該慣性系統。

在方塊 440 中，該應用程式自慣性系統接收一操縱事件，其描述該識別的應用物件之一或多個操縱。例如，該應用程式可以接收描述該應用物件之 2D 仿射轉換之一事件。請注意到方塊 440 係在方塊 430 之後依序圖示，

以用於簡化說明。實際上，該應用程式在慣性系統通知該應用程式一操縱事件之前，可以接收許多觸碰輸入事件。不需要有觸碰輸入事件對操縱事件之一對一對映。因為操縱事件代表低階觸碰輸入之較高階解譯，多個觸碰輸入可構成一單一操縱事件。在方塊 450 中，該應用程式處理接收到的操縱事件。例如，如果該收到的操縱事件為一旋轉，則該應用程式可以在螢幕上旋轉應用物件，並儲存該等應用物件之新的位置來以用於當該應用物件被再次顯示時。慣性系統使得該應用程式可以免於在一特殊多點觸控硬體裝置所執行之特定步驟，或甚至可免於知道哪一個硬體裝置正在提供該多點觸控輸入。此外，該慣性系統使得該應用程式免於處理個別的接觸移動，並允許該應用程式聚焦在處理該應用物件層級的轉換。

在方塊 460 中，該應用程式等待下一個觸碰輸入。例如，該應用程式可以呼叫提供訊息 API 之作業系統，該訊息為像是 Microsoft Windows 上的 GetMessage，其等待下一個訊息來傳遞到該應用程式的訊息佇列。在決策方塊 470 中，如果該應用程式接收到下一個觸控輸入，則該應用程式經迴路至方塊 410 來處理該輸入，否則該應用程式經迴路至方塊 460 來繼續等待另外的輸入。當

應用程式關閉時，該應用程式離開該輸入迴圈(未顯示出)。

第 5 圖為在一具體實施例中，用以說明當該系統接收觸控輸入時，慣性系統之處理的流程圖。在方塊 505 中，該系統接收觸控輸入及識別該觸碰輸入所相關的一應用物件之資訊。例如，該觸碰輸入可以包含一或多次觸碰接觸之座標或其他位置資訊，且該應用物件資訊可以包含一識別碼，其為該應用程式指定給該觸碰輸入位在多點觸控硬體之上的一特定顯示物件。在方塊 510 中，該系統識別相關於該應用物件之一操縱處理器。在決策方塊 520 中，如果該系統先前並未使一操縱處理器與該應用物件產生相關，則該系統繼續到方塊 530，否則該系統繼續到方塊 540。在方塊 530 中，該系統產生一操縱處理器，且將其與應用物件產生相關，然後繼續到方塊 540。

在決策方塊 540 中，如果接收的觸碰輸入指示該應用程式接收一新接觸(例如一向下觸碰事件)，則該系統繼續到系統 550，否則該系統繼續到方塊 560。例如，使用者可以進行手指與一螢幕上物件的初始接觸，或放置另一隻手指(即接觸)於先前觸碰過的物件上。在方塊 550 中，該系統加入新的接觸到相關於操縱處理器之接觸清單，然後繼續到方塊 560。在決策方塊 560 中，如果接

收的觸碰輸入指示該應用程式接收到一觸碰接觸被移除(例如向上觸碰事件)的通知，則系統繼續到方塊 570，否則系統繼續到方塊 580。例如，該使用者可自一先前觸碰過的物件移開一或多隻手指。在方塊 570 中，系統自相關於該操縱處理器的接觸清單中移除該接觸，然後繼續到方塊 580。在方塊 580 中，該系統處理觸碰輸入來判定由該觸碰輸入所代表的任何操縱。例如，觸碰移動可以指示一旋轉或平移操縱，而觸碰接觸移除可指示操縱完成。在方塊 590 中，該系統發出一操縱事件來傳送描述該操縱的轉換資訊到該應用程式。例如該系統可以提供該物件之角旋轉度數到該應用程式。在方塊 590 之後，這些步驟即結束。

第 6 圖為在一具體實施例中，用以說明使用該慣性系統來處理慣性事件的一多點觸控應用程式之處理的流程圖。在方塊 610 中，該應用程式判定使用者已經釋放一物件。例如，在第 4 圖的方塊 450 中之操縱事件的處理之後，該應用程式可接收到操縱已完成的指示，或者接收到該使用者已經提起觸碰一應用物件之所有接觸的指示。在方塊 620 中，該應用程式初始化慣性系統。例如，該應用程式可傳送一基準給正在處理該物件之移動及其他初始化資訊的操縱處理器。在方塊 630 中，該應用程式設定一計時器，其將會驅動該慣性系統之慣性處理期

間。例如，該應用程式可以設定一計時器，其將會在每 100 毫秒發出以處理一物件之下一個移動增量。在方塊 640 中，該應用程式呼叫該慣性系統之處理功能(參見第 7 圖)。例如，該慣性系統可以提供一「處理」功能，其可由該應用程式呼叫來通知慣性系統，自從最後模擬期間起算已經是時候來執行該期間的模擬。

在方塊 650 中，該應用程式接收一或多個慣性事件，其描述基於模擬的慣性之該物件的操縱(例如旋轉、平移及/或縮放)。例如，如果該物件在一特定方向上行進，該應用程式可以接收描述在該方向上一平移操縱的慣性事件。在另一例中，如果該物件當使用者釋放它時正在變大，該應用程式可以接收描述一縮放操縱之慣性事件。請注意到方塊 650 係在方塊 640 之後依序圖示，用於簡化說明。實際上，該應用程式在慣性系統通知該應用程式一慣性事件之前，會數次呼叫該慣性處理功能。並不需要一對一的呼叫對映到該處理功能及慣性事件。另一方面，該慣性系統在單一呼叫到該處理功能之後，可以通知該應用程式有多個慣性事件。

在方塊 660 中，該應用程式基於在該特定應用之內容中操縱的意義(即效果)來處理該接收的慣性事件。例如，如果該接收的慣性事件為旋轉，則該應用程式可以在螢幕上旋轉該應用物件，並儲存該等應用物件之新的

位置，以當應用程式再次顯示該應用物件時來使用。在決策方塊 670 中，如果該等慣性事件已完成，則這些步驟結束，否則該系統繼續到方塊 680。慣性系統可以通知該應用程式一特定模擬的操縱已完成，做為來自該處理功能或經由該等通知提供給該應用程式之傳回值(例如經由一組件物件模型(Component Object Model, COM)事件介面)。在方塊 680 中，該應用程式等待計時器的下一次發動，然後經迴路至方塊 640 來呼叫該慣性系統處理功能。

第 7 圖為在一具體實施例中，用以說明該慣性處理系統之模擬組件的處理之流程圖。在方塊 710 中，該組件接收初始模擬參數。例如，一應用程式或操縱處理器在當使用者停止觸碰該物件時，可以提供一應用物件之最後狀態。在方塊 720 中，該組件初始化一模擬引擎，其執行根據物理的計算來判定基於該等參數之一物件的行為。例如，該模擬引擎可以對於由使用者觸碰輸入藉由動作所設定之應用物件提供實際減速或彈性行為。在方塊 730 中，該組件接收來自該應用程式之一處理呼叫。該應用程式或其他組件藉由重複地以固定間隔呼叫一處理功能來驅動該模擬處理來使該模擬向前行進。該模擬組件亦可以在內部產生計時器。

在方塊 740 中，該組件基於該等初始參數、任何先前處理，及從最後處理呼叫所經過的時間來模擬該物件之移動。該處理呼叫亦可以提供一時間標記，其指示該應用程式想要使用該模擬的時間。此允許該應用程式以非即時方式來模擬應用行為(例如對於應用測試或偵錯)。在決策方塊 750 中，如果該移動已完成，則該組件繼續到方塊 760，否則該組件繼續到方塊 770。該組件可以基於一些因素，像是該物件是否仍在移動，或該物件移動是否已經低於某個臨界值，而判定該移動已完成。在方塊 760 中，該組件在下一個慣性事件上設定一完成旗標。在方塊 770 中，該組件發出一慣性事件來傳送描述目前移動(例如做為一操縱)的轉換資訊到該應用程式。例如系統可以提供該物件之角旋轉度數到該應用程式。在方塊 770 之後，這些步驟即結束。

在一些具體實施例中，該慣性系統自該應用程式接收物件限制。例如，該應用程式可以定義一物件之彈性、摩擦係數(以判定物件如何減速)、該物件之邊界特性等等。例如，一應用程式開發者可以定義使用者可以移動的固態物件及一彈性應用視窗邊緣，如此移動到該視窗邊緣中的物件在當使用者釋放它們時，即自該視窗邊緣彈回。

在一些具體實施例中，慣性系統接收來自一操縱系統之初始物件狀態資訊，其正在追蹤當使用者正在利用觸碰輸入操縱該物件時該物件的移動。例如，該操縱系統可以追蹤每個物件之目前位置、該物件之歷史移動、該物件之線性及角速度等等。該應用程式開發者可以提供該操縱之輸出到慣性系統來初始化該慣性系統，如此該慣性系統基於適當的物理及該物件之特性而平滑地繼續該物件之過去的移動並將其速度減慢。

在一些具體實施例中，該慣性系統接收來自該應用程式對於物件移動的限制。例如，該應用程式開發者可以定義一物件於一旦使用者釋放該物件時，它能夠移動之距離的上限。在另一例中，該應用程式可以定義該物件於一旦使用者釋放該物件時，它能夠移動多長距離的上限。這些及其他限制允許該應用程式開發者調整該慣性系統，來符合由該應用程式操縱之物件的種類，並增進該使用者對於該應用程式的經驗。

在一些具體實施例中，該慣性系統對於物件在低於一預先定義的臨界值來移動時，不會提供額外的移動。臨界值可藉由該應用程式來組態。例如，該慣性系統在使用者釋放該物件之後，可使得一特定物件具有低於該系統不會繼續將該物件移動的線性或角速度。如果該物件在使用者釋放它時並未非常快地移動，該使用者可以預

期該物件將會原地不動，且不再繼續移動。該臨界值允許該應用程式或該慣性系統之開發者在提供一良好的使用者經驗的操縱後可以判定該移動層級。

在一些具體實施例中，慣性系統自該應用程式接收指令來遞增模擬移動。例如，該慣性系統可以提供一「處理(Process)」或「進行工作(DoWork)」功能，其由該應用程式呼叫來指示該慣性系統來執行整體模擬之一部份。該慣性系統可以預期該應用程式來設定一計時器，或另外定期地呼叫該功能來使得該慣性系統能根據一自然時間軸模擬隨時間的移動。該應用程式可以藉由改變該應用程式呼叫該功能的頻率來影響由該慣性系統所提供的操縱事件之特性。在其它具體實施例中，慣性系統使用一內部計時器來以固定時程提供操縱事件，直到每個物件已經停止移動(例如由於減速或其它模擬的力量)。

在一些具體實施例中，慣性系統為一訊息式作業系統之一部份，且該系統接收關於該作業系統自硬體接收之觸碰輸入的訊息。例如，使用類似於滑鼠訊息之 WM_MOUSEMOVE 的範例，Microsoft Windows 的未來版本可以提供一 WM_TOUCH 訊息，其含有自多點觸控硬體接收的低階觸碰移動資訊。該作業系統亦可提供較細微的微粒訊息，像是 WM_TOUCHDOWN(當利用多點

觸控硬體產生一新接觸時)、WM_TOUCHMOVE(當一既有接觸移動時)、及 WM_TOUCHUP(當一接觸自該多點觸控硬體提起時)。接收一 WM_TOUCH 相關的訊息之應用程式可以引動慣性系統，並傳送訊息到該慣性系統來解譯及處理。該應用程式則接收較高階的事件，其基於接收到的低階觸碰移動資訊代表由該使用者所想要的該操縱之慣性系統的解譯。

在一些具體實施例中，慣性系統自特殊化的硬體(像是即時觸控筆)接收低階觸碰移動資訊。例如，微軟平板型個人電腦軟體開發工具箱 (Microsoft Tablet PC Software Development Kit, SDK)提供應用程式開發者可利用掛鉤延伸的一即時觸控筆(Real-time stylus, RTS)組件。RTS 掛鉤接收來自該 RTS 硬體的輸入，並可對該接收到的輸入執行處理。該慣性系統可以提供一掛鉤，使一應用程式置入到該 RTS 組件中來自動地處理 RTS 及其它輸入以操縱應用物件，以如本文所述。該 RTS 掛鉤提供一不同的方式給慣性系統來接收輸入，但該慣性系統解譯輸入，並發出事件給該應用程式，其描述由輸入所意味的操縱，如前所述。使用者可以使用觸控筆與觸碰輸入之組合。例如，該使用者可以利用觸控筆畫出一物件，然後使用他/她的手指旋轉該物件。

在一些具體實施例中，該慣性系統為一應用程式可以引動來提供一共用使用者介面之一共用控制項的一部份。Microsoft Windows 提供顯示清單、樹狀、按鈕等等之共用控制項。類似地，該慣性系統可以提供一多點觸控式控制項，用於以本文所述的方式來操縱應用物件。例如，該系統可以提供一分散控制項，其允許使用者顯示一或多個物件，並操縱該等物件。該分散控制項處理低階觸碰輸入的處理，並使該輸入與一特定應用物件產生相關，且該應用程式接收來自該控制項之事件來處理該等應用物件之操縱。例如，如果該控制項指示使用者調整一物件之大小，則該應用程式可以儲存該等物件之新尺寸。

在一些具體實施例中，該慣性系統在三維中執行本文所述的處理。雖然本文所述為二維多點觸控硬體，本技術專業人士將可瞭解到本文所述之系統的處理，可同樣地成功套用到三維(three-dimensional, 3D)操縱，如果該硬體可用於提供三維的座標移動。例如，偵測壓力或使用相機來偵測使用者手指的 3D 移動的硬體，可以提供三維的移動之座標到該慣性系統，並且該慣性系統則可以產生在多個 3D 方向上描述物件之操縱(例如旋轉、縮放及平移)的 3D 轉換。

下表定義一 API，其由該慣性系統提供給應用程式，以在該等物件之使用者觸碰式移動之後，提供慣性式移動到應用物件。

特性:	
BoundaryBottom	限制該目標物件朝向該螢幕的底部能夠移動多遠
BoundaryLeft	限制該目標物件朝向該螢幕的左方能夠移動多遠
BoundaryRight	限制該目標物件朝向該螢幕的右方能夠移動多遠
BoundaryTop	限制該目標物件朝向該螢幕的上方能夠移動多遠
DesiredAngularDeceleration	指定每毫秒該目標物件將停止旋轉以弧度為單位的期望速率
DesiredDecleration	指定平移操作將減速之期望速率
DesiredDisplacement	指定該物件將行進之期望距離
DesiredExpansion	指定在該物件的平均半徑之期望變化
DesiredExpansionDeceleration	指定該物件將停止擴大之速率
ElasticMarginBottom	指定彈回該目標物件之底部區域

ElasticMarginLeft	指定彈回該目標物件之最左方區域
ElasticMarginRight	指定彈回該目標物件之最右方區域
InitialAngularVelocity	指定當移動開始時該目標之旋轉
InitialOriginX	取得或放置指定一目標物件之水平位置之特性。此特性指定具有慣性之一目標的開始水平位置。
InitialOriginY	取得或放置指定一目標物件之垂直位置之特性。此特性指定具有慣性之一目標的開始垂直位置。
InitialRadius	指定在該物件被改變之前該目標之邊緣與其中心的距離
InitialTimestamp	指定具有慣性之一目標物件之開始時間標記
InitialVelocityX	指定該目標物件在該水平軸上之初始移動
InitialVelocityY	指定該目標物件在該垂直軸上之初始移動
方法:	

HRESULT Reset();	利用初始時間標記初始化該處理器
HRESULT Process([out] BOOL* completed);	對給定時刻執行計算，並根據是否完成外插來引發 Delta 或 Completed 事件。如果外插在先前時刻完成，該方法即無進行作業。
HRESULT ProcessTime([in] DWORD timestamp, [out] BOOL* completed);	對給定時刻執行計算，並根據是否完成外插來引發 Delta 或 Completed 事件。如果外插在先前時刻完成，該方法即無進行作業。
HRESULT Complete();	引發該 Completed 事件
HRESULT CompleteTime([in] DWORD timestamp);	對給定時刻進行處理，並引發 Completed 事件
事件:	
HRESULT ManipulationStarted([in] FLOAT x, [in] FLOAT y);	處理當一操縱開始時的事件
HRESULT ManipulationDelta([in] FLOAT x, [in] FLOAT y, [in] FLOAT translationDeltaX, [in] FLOAT translationDeltaY,	處理當一操縱物件改變時發生的事件

<pre> [in] FLOAT scaleDelta, [in] FLOAT expansionDelta, [in] FLOAT rotationDelta, [in] FLOAT cumulativeTranslationX, [in] FLOAT cumulativeTranslationY, [in] FLOAT cumulativeScale, [in] FLOAT cumulativeExpansion, [in] FLOAT cumulativeRotation); </pre>	
<pre> HRESULT ManipulationCompleted([in] FLOAT x, [in] FLOAT y, [in] FLOAT cumulativeTranslationX, [in] FLOAT cumulativeTranslationY, [in] FLOAT cumulativeScale, [in] FLOAT cumulativeExpansion, [in] FLOAT cumulativeRotation); </pre>	處理當操縱完成時的事件

在上表中，該慣性系統基於使用者移動，在與一應用程式在先前接收事件的相同介面上提供該等列出的事件。

由前所述，應瞭解該慣性系統之特定具體實施例已經在本文為了圖示目的來描述，但可在不背離本發明之精神及範疇之下進行不同的修正。例如，雖然該系統已經在多點觸控操縱之背景中描述，該系統提供的慣性之模擬可以用於其他背景中，像是遊戲及其他經常使用模擬

的領域。因此，本發明除了隨附申請專利範圍外，並不受限。

【圖式簡單說明】

第 1 圖為在一具體實施例中，說明該慣性系統之組件的方塊圖。

第 2 圖為在一具體實施例中，說明該慣性系統之典型操作環境及組件之間的資料流動之資料流程圖。

第 3 圖為在一具體實施例中，用以說明由使用者觸碰所操縱的一應用物件之示意圖。

第 4 圖為在一具體實施例中，用以說明使用慣性系統來處理操縱事件的一多點觸控應用之輸入迴圈處理的流程圖。

第 5 圖為在一具體實施例中，用以說明當系統接收觸控輸入時該慣性系統之處理的流程圖。

第 6 圖為在一具體實施例中，說明使用該慣性系統來處理慣性事件的一多點觸控應用之處理的流程圖。

第 7 圖為在一具體實施例中，說明該慣性處理系統之模擬組件的處理之流程圖。

【主要元件符號說明】

100 慣性系統

110 硬體介面

120 操縱處理器

130 輸入轉換組件

140 模擬組件	290 慣性事件
150 應用程式介面	310 顯示器
210 來自硬體介面的輸入	320 文件物件
220 輸入事件	330 資源回收筒
230 應用	340 第一操縱位置
240 輸入變化	350 第二操縱位置
250 操縱系統	360 第三操縱位置
260 操縱事件	365 箭頭
270 初始化資訊	370 第一慣性式操縱位置
280 慣性系統	380 第二慣性式操縱位置

104年1月7日	修正 對線 再(本)
----------	------------------

P34~39

七、申請專利範圍：

1. 一種在包含一觸控敏感輸入表面及一顯示器之一計算裝置上執行之方法，該方法係用於經由該觸控敏感輸入表面提供使用多點觸控輸入 (multi-touch input) 而被操縱之物件在該顯示器上之真實移動，該方法包括下列步驟：

判定一使用者藉由自一多點觸控輸入裝置移除一或更多個接觸而已經釋放執行於該計算裝置上之一應用程式之一應用物件，該應用物件顯示於該顯示器上；

在該判定後，調用執行於該計算裝置上之一慣性 API(應用程式介面)以在一目前慣性處理期間處理該應用物件之移動的一模擬，其中該慣性 API 提供可由該計算裝置上之任意應用程式調用之一與應用程式無關 (application-independent) 的平台，該慣性 API 係模擬無關於該應用物件之一類型之慣性物件之移動；

藉由該應用程式接收在該判定後藉由該慣性 API 所產生之一慣性事件，其中該慣性事件根據模擬的慣性描述該物件之一操縱；及

根據在該應用程式之一內容中之該操縱之一效果而藉由修改該應用物件來處理所接收到的該慣性

事件，從而顯示該應用物件於該顯示器上。

2. 如申請專利範圍第 1 項所述之方法，更包括以下步驟：在判定該使用者釋放該物件之後，其係於該使用者釋放該物件之前，藉由傳送一基準到正在處理該物件之該移動的一操縱處理器來初始化該慣性系統。
3. 如申請專利範圍第 1 項所述之方法，更包括以下步驟：在判定該使用者釋放該物件之後，其係當該使用者釋放該物件時，藉由傳送描述該物件的一狀態之一或更多個參數來初始化該慣性系統。
4. 如申請專利範圍第 1 項所述之方法，其中接收該慣性事件之步驟包括以下步驟：接收描述該應用物件之一 2D 仿射轉換（2D affine transform）之資訊。
5. 如申請專利範圍第 1 項所述之方法，其中由該慣性事件描述的該操縱包含：一旋轉操縱、一平移操縱、及一縮放操縱當中至少一項。
6. 如申請專利範圍第 1 項所述之方法，其中該接收的慣性事件為一旋轉，且其中處理該接收的慣性事件之步

驟包括以下步驟：在一顯示器上旋轉該應用物件。

7. 如申請專利範圍第 1 項所述之方法，更包括以下步驟：設定一計時器，其決定該慣性處理期間的一長度，且其中該應用程式在該計時器每次啟動時調用該慣性 API。
8. 如申請專利範圍第 7 項所述之方法，更包括以下步驟：自該慣性 API 接收該模擬已完成之一指示，且當該指示被接收時，使該計時器失效。
9. 如申請專利範圍第 1 項所述之方法，其中接收該慣性事件之步驟包括以下步驟：經由一 COM 事件介面接收一通知。
10. 一種電腦可讀取儲存媒體，以用於控制包含一顯示器及一觸控敏感輸入表面之一計算裝置的指令編碼該電腦可讀取儲存媒體，以經由該觸控敏感輸入表面模擬先前藉由多點觸控輸入而移動的一應用物件在顯示器上之移動，該應用物件係為執行於該計算裝置上之一應用程式之部分，該電腦可讀取儲存媒體所藉方法包括下列步驟：

藉由停止與該觸控敏感輸入表面之接觸而接收一或更多個初始模擬參數，該一或更多個初始模擬參數提供對應至一使用者釋放該應用物件之該應用物件的一狀態；

初始化一模擬引擎，該模擬引擎執行計算，以根據該等初始模擬參數來模擬該應用物件的慣性；

接收一目前模擬期間之一指示；

根據該等初始參數、藉由該模擬引擎之任何先前之慣性模擬、及自從任何先前模擬期間以來之已經經過的時間而藉由該模擬引擎來模擬該應用物件的慣性移動；及

發出一慣性事件來傳送描述該應用物件之一目前移動的轉換資訊至該應用程式，根據該轉換資訊而顯示該應用物件於該顯示器上。

11. 如申請專利範圍第 10 項所述之電腦可讀取媒體，其中該模擬引擎模擬在藉由該使用者觸碰該物件而使其移動後之該應用物件的真實減速行為。

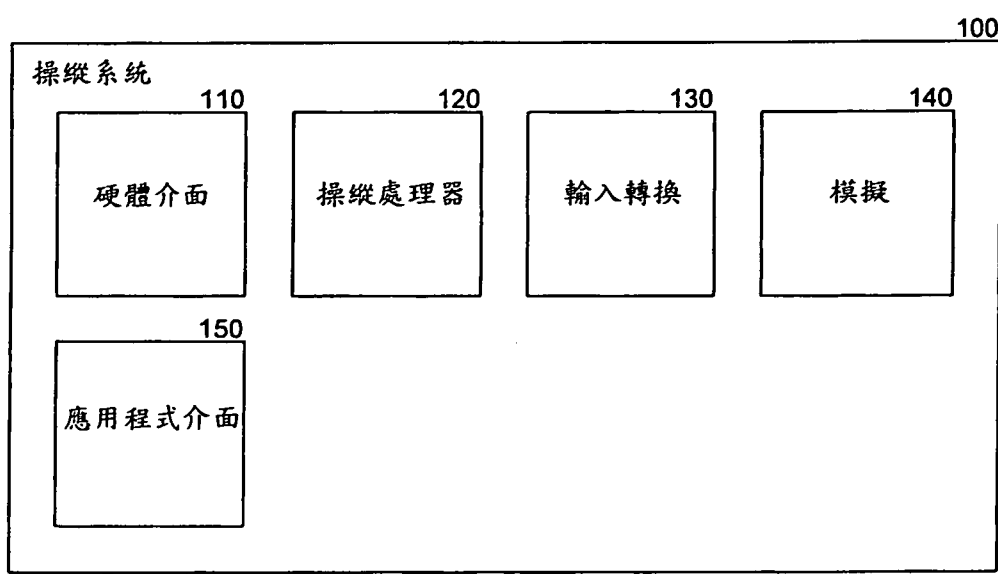
12. 如申請專利範圍第 10 項所述之電腦可讀取媒體，其中該模擬引擎模擬當該應用物件在一顯示器上重疊於另一元件時之該應用物件的真實彈性行為。

13. 如申請專利範圍第 10 項所述之電腦可讀取媒體，其中接收該目前模擬期間之該指示的步驟，包括以下步驟：自該應用程式接收到一模擬處理功能之呼叫。
14. 如申請專利範圍第 10 項所述之電腦可讀取媒體，更包括判定：該模擬是否完成，且如果該模擬已完成，則通知該應用程式該模擬已完成。
15. 一種用於處理來自多點觸控硬體之觸控輸入之計算裝置，該計算裝置包含：
- 一模組，該模組接收來自一多點觸控輸入裝置之用於任意應用程式之低階輸入，及解譯該等低階輸入至由該等任意應用程式所接收之操縱，其中至少一些操縱係基於個別之複數個該等低階輸入，該等操作包含 2D 仿射轉換；
 - 一應用程式，該應用程式包含顯示於該顯示器上之一圖形物件，該圖形物件係與接觸該多點觸控輸入裝置之一使用者互動，使得經由該模組接收針對該圖形物件之對應低階輸入，及該應用程式接收由該模組所產生之對應操縱，該等對應操縱係由該應用程式所使用以顯示該圖形物件，其中當該使用者停止接觸該

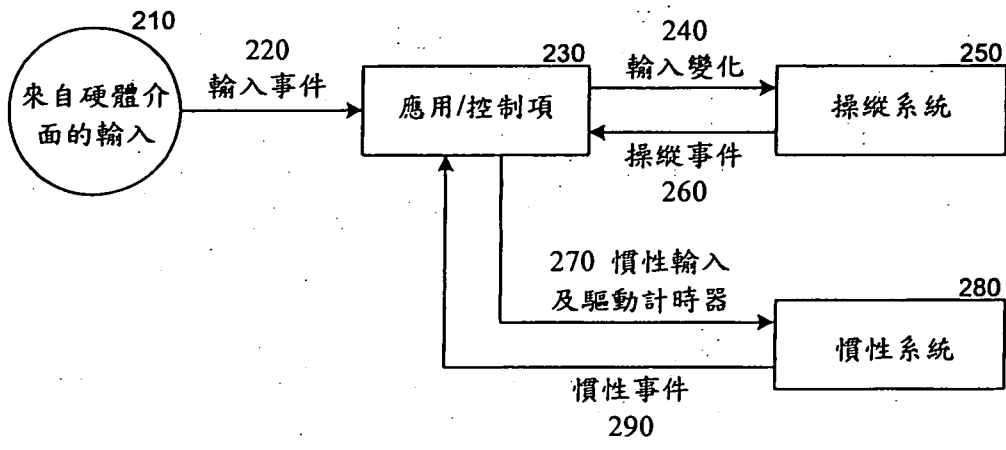
多點觸控輸入裝置時，該模組藉由模擬減少該物件之慣性而持續產生用於該圖形物件之操縱。

16. 如申請專利範圍第 15 項所述之計算裝置，其中模組根據在該使用者停止接觸該多點觸控輸入裝置前之該物件之一狀態，而模擬該減少慣性。
17. 如申請專利範圍第 16 項所述之計算裝置，其中該狀態包含該應用物件之一速度及方向。
18. 如申請專利範圍第 15 項所述之計算裝置，其中該等低階輸入包含該使用者正接觸中之該多點觸控輸入裝置之位置。
19. 如申請專利範圍第 15 項所述之計算裝置，其中該模組使該應用程式接收該圖形物件已停止之一通知。
20. 如申請專利範圍第 15 項所述之計算裝置，其中該等操縱在該使用者停止接觸該多點觸控輸入裝置前及在該使用者停止接觸該多點觸控輸入裝置後皆具有一相同格式及至該應用程式之一相同通訊通道。

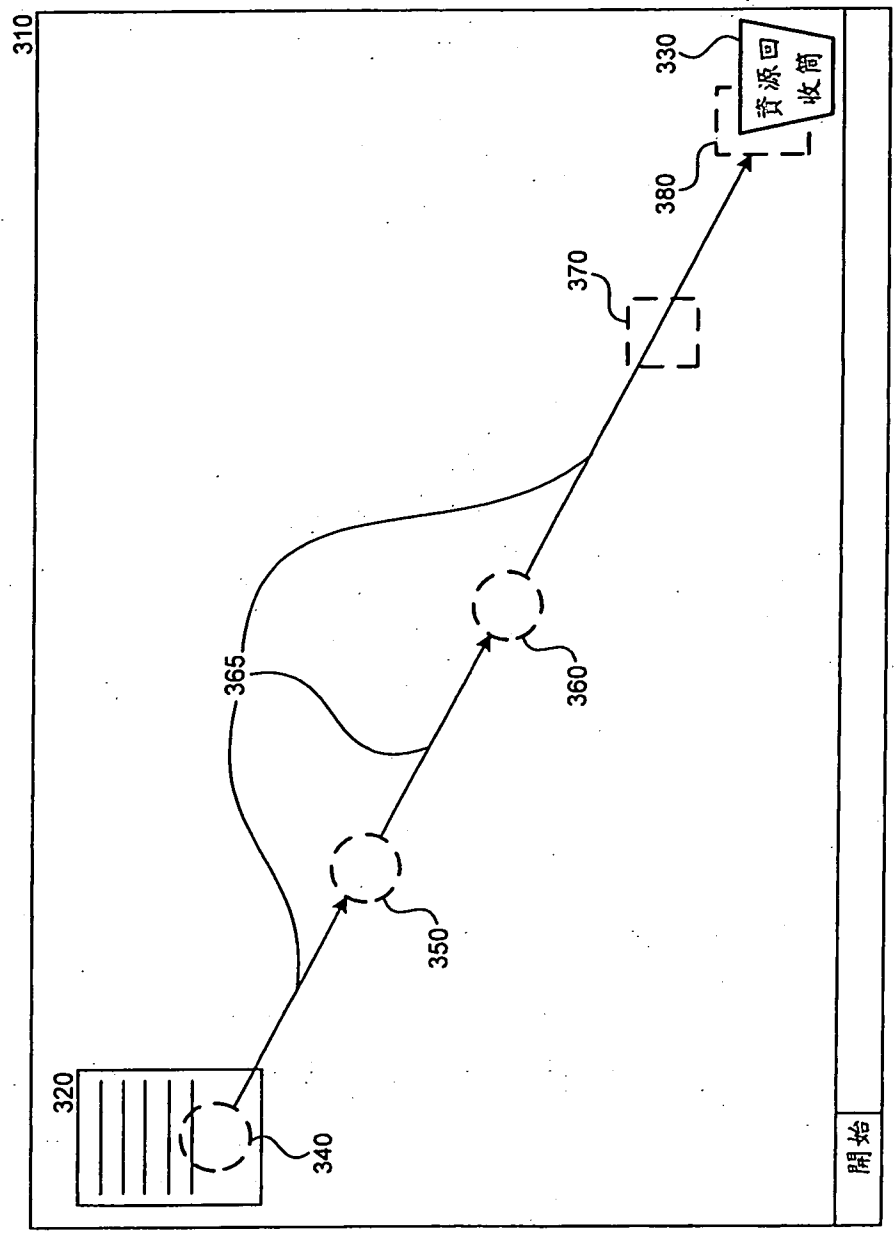
八、圖式：



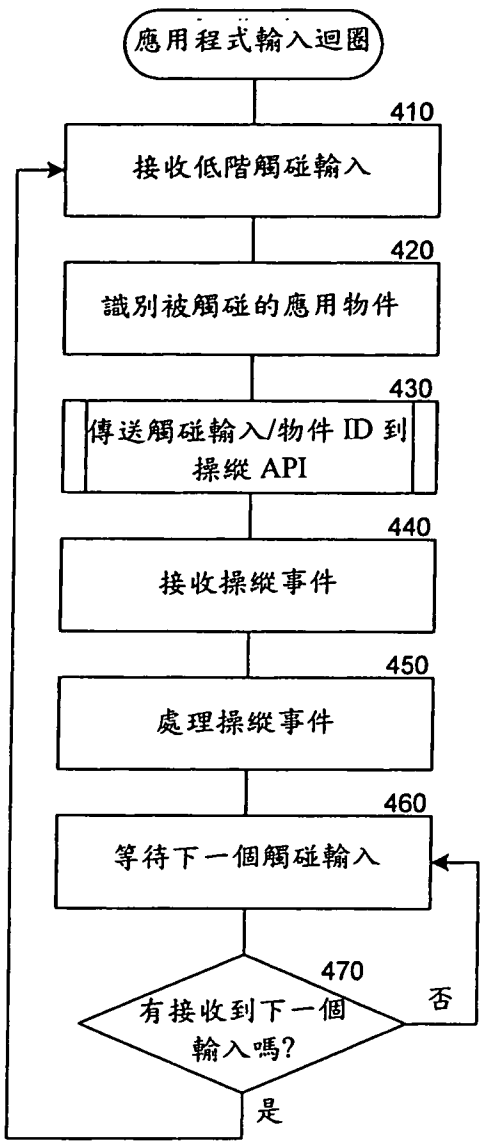
第 1 圖



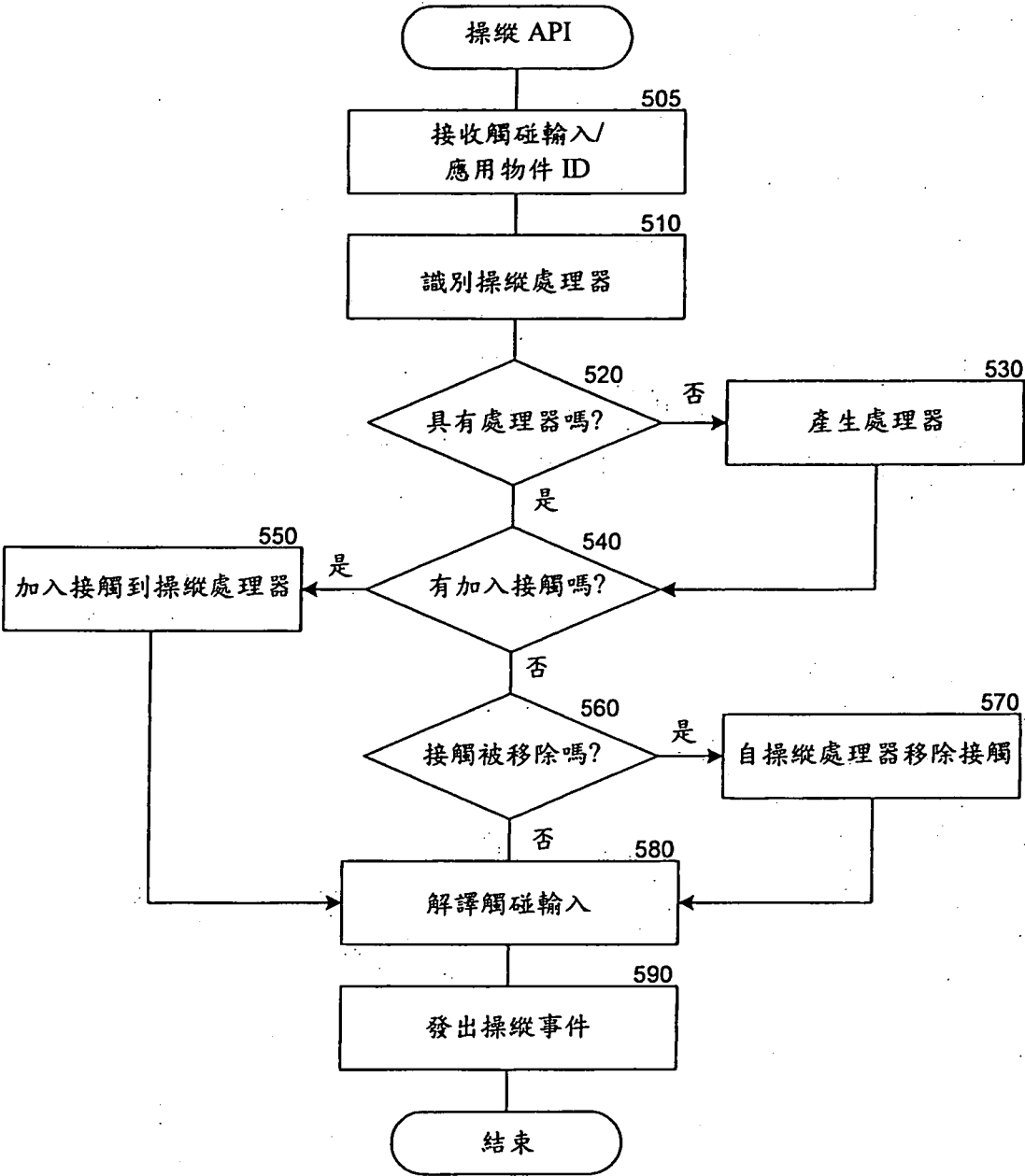
第 2 圖



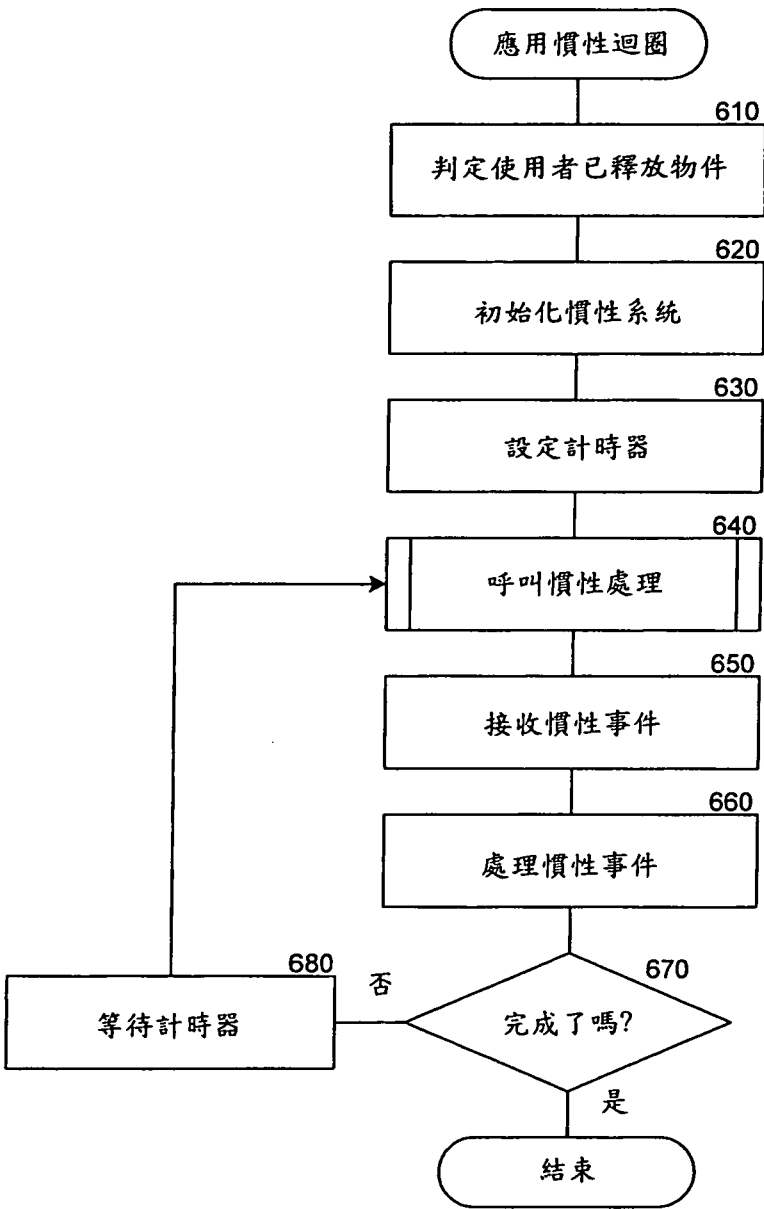
第 3 圖



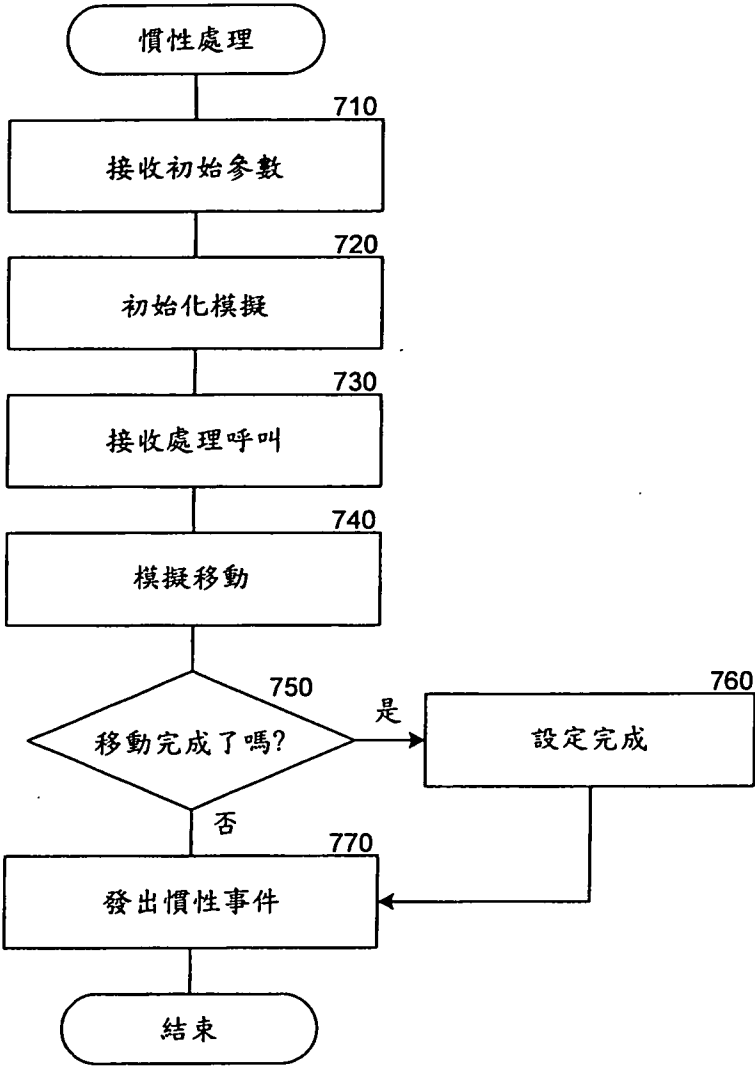
第 4 圖



第 5 圖



第 6 圖



第 7 圖