

19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 771 356**

51 Int. Cl.:

|                    |           |
|--------------------|-----------|
| <b>G06F 3/041</b>  | (2006.01) |
| <b>G06F 3/048</b>  | (2013.01) |
| <b>G06F 9/44</b>   | (2008.01) |
| <b>G06F 3/0488</b> | (2013.01) |
| <b>G06F 3/0484</b> | (2013.01) |
| <b>G06F 3/0486</b> | (2013.01) |
| <b>G06F 3/0481</b> | (2013.01) |

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

- 86 Fecha de presentación y número de la solicitud internacional: **16.10.2009 PCT/US2009/060977**
- 87 Fecha y número de publicación internacional: **29.04.2010 WO10048051**
- 96 Fecha de presentación y número de la solicitud europea: **16.10.2009 E 09822481 (9)**
- 97 Fecha y número de publicación de la concesión europea: **01.01.2020 EP 2342638**

54 Título: **Simulación de inercia de objetos multitáctiles**

30 Prioridad:

**26.10.2008 US 258439**

45 Fecha de publicación y mención en BOPI de la traducción de la patente:

**06.07.2020**

73 Titular/es:

**MICROSOFT TECHNOLOGY LICENSING, LLC  
(100.0%)  
One Microsoft Way  
Redmond, WA 98052 , US**

72 Inventor/es:

**TOWNSEND, REED L.;  
TU, XIAO;  
SCOTT, BRYAN D.;  
TORSET, TODD A.;  
SYKES, KENNETH W.;  
PRADHAN, SAMIR S. y  
TEED, JENNIFER A.**

74 Agente/Representante:

**GONZÁLEZ PECES, Gustavo Adolfo**

ES 2 771 356 T3

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín Europeo de Patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre Concesión de Patentes Europeas).

## DESCRIPCIÓN

Simulación de inercia de objetos multitáctiles

**Antecedentes**

5 Un PC tipo tableta u ordenador con lápiz, es un ordenador portátil o móvil con forma de tableta táctil, equipado con una pantalla táctil o una tecnología híbrida de tableta/pantalla de gráficos que permite al usuario operar el ordenador con un lápiz óptico, un bolígrafo digital o con la punta del dedo en lugar de un teclado o ratón. Los PC tipo tableta ofrecen una forma de entrada más natural, puesto que dibujar y escribir son una forma de entrada mucho más familiar que un teclado y un ratón, especialmente para las personas que están comenzando a usar ordenadores. Los PC tipo tableta pueden también ser más accesibles porque aquellos que no pueden escribir físicamente pueden utilizar las características adicionales de un PC tipo tableta para poder interactuar con el mundo electrónico.

10 Multi-táctil (o multitáctil) denota un conjunto de técnicas de interacción que permiten a los usuarios de ordenadores controlar aplicaciones gráficas utilizando múltiples dedos o dispositivos de entrada (por ejemplo, lápiz óptico). Las implementaciones multitáctiles incluyen por lo general hardware táctil (por ejemplo, una pantalla, mesa, pared, etc.) y software que reconoce múltiples puntos táctiles simultáneos. Multitáctil contrasta con las pantallas táctiles tradicionales (por ejemplo, panel táctil del ordenador, cajero automático, quiosco de compras) que solo reconocen un punto táctil a la vez. El hardware multitáctil puede detectar toques utilizando calor, presión de los dedos, cámaras de captura de alta velocidad, luz infrarroja, captura óptica, inducción electromagnética sintonizada, receptores ultrasónicos, micrófonos transductores, telémetros láser, captura de sombras y otros mecanismos. Existen muchas aplicaciones para interfaces multitáctiles y los diseñadores y usuarios de aplicaciones proponen aún más. Algunos usos son individualistas (por ejemplo, Microsoft Surface, Apple iPhone, HTC Diamond). Como un nuevo procedimiento de entrada, multitáctil ofrece el potencial para nuevos paradigmas de experiencia de usuario.

15 Una aplicación no puede usar hardware multitáctil sin una interfaz para que el software de la aplicación reciba información del hardware multitáctil. Desafortunadamente, cada dispositivo de hardware multitáctil incluye su propia interfaz patentada y los autores de la aplicación deben tener un conocimiento específico de un dispositivo de hardware para escribir un software que funcione con el dispositivo. Por ejemplo, un proveedor de hardware multitáctil puede proporcionar un controlador en modo kernel y una interfaz de aplicación en modo usuario a través de la que las aplicaciones de software en modo usuario pueden comunicarse con el hardware multitáctil para recibir información táctil. El autor de una aplicación escribe el software que se comunica con la interfaz de la aplicación en modo de usuario, pero el software del autor de la aplicación solo funciona con ese hardware multitáctil. Un usuario del ordenador con un dispositivo de hardware multitáctil diferente no puede usar el software del autor de la aplicación a menos que el autor de la aplicación produzca una versión diferente del software que funcione correctamente con el dispositivo informático del usuario. Esto produce un mercado potencial muy limitado para los autores de aplicaciones, reduce el incentivo para escribir aplicaciones que admitan interacciones multitáctiles y mantiene el coste de los dispositivos más populares para los que hay mayor cantidad de aplicaciones disponibles.

20 Otro problema es la dificultad de las aplicaciones para determinar las intenciones de un usuario en función de la entrada táctil recibida del hardware multitáctil. La entrada táctil puede recibirse como una lista de coordenadas donde el hardware detecta la entrada táctil en cualquier momento dado. Cada aplicación debe incluir un software para interpretar las coordenadas y determinar la intención del usuario. Además, la intención del usuario puede extenderse más allá de la entrada táctil real recibida. El usuario puede esperar que los objetos virtuales se comporten como lo hacen en el mundo físico. Por ejemplo, un usuario puede esperar poder "arrojar" un archivo de un lado del escritorio a otro moviendo su dedo. Este tipo de movimiento no es compatible con las aplicaciones multitáctiles actuales, que esperarían que el usuario arrastre su dedo desde un lado de la pantalla hasta el otro. Incluso si una aplicación proporciona soporte para este tipo de movimiento, otras aplicaciones no podrían beneficiarse y, por lo tanto, los autores de la aplicación tendrían que repetir el trabajo del primer autor de la aplicación para ofrecer la misma funcionalidad en sus aplicaciones.

25 El documento WO 2006/020305 A2 desvela un procedimiento implementado por ordenador para proporcionar un movimiento realista de objetos manipulados usando entrada multitáctil, el procedimiento comprende las etapas de:

- determinar que un usuario ha soltado un objeto de aplicación eliminando uno o más contactos de un dispositivo de entrada multitáctil; y
- 50 - procesar una simulación de movimiento del objeto de aplicación durante un período de procesamiento de inercia actual, simulando así un movimiento realista independiente de un tipo de objeto de aplicación. Dicho documento desvela también la recepción de parámetros de simulación iniciales que proporcionan un último estado del objeto de aplicación cuando un usuario ha soltado el objeto de aplicación y un motor de simulación que realiza cálculos basándose en la física para determinar el comportamiento del objeto de aplicación basándose en los parámetros de simulación inicial. Además, desvela que la simulación del movimiento del objeto de aplicación con inercia incluye el movimiento del objeto correspondiente a una transformación afín 2D del objeto.

Dichas características se desvelan también en el documento US 2006/125799 A1.

El documento US 2008/168402 A1 desvela el uso de una API de interfaz de programación de aplicaciones en el mismo

contexto de movimiento de objetos GUI manipulados mediante entrada multitáctil. Desvela la invocación de una API para procesar un movimiento del objeto de aplicación, dicho movimiento corresponde a una transformación afín 2D del objeto de aplicación; dicha API procesa dicho movimiento independientemente de un tipo de objeto de aplicación.

**Sumario**

5 Un procedimiento implementado por ordenador, un sistema informático y un medio de almacenamiento legible por ordenador de acuerdo con la invención se establecen en las reivindicaciones 1, 6 y 10, respectivamente.

El sistema de inercia proporciona una plataforma común y una interfaz de programación de aplicaciones (API) para que las aplicaciones extiendan la entrada recibida de varios dispositivos de hardware multitáctiles para simular el comportamiento de los objetos en el mundo real. Las manipulaciones recibidas por la aplicación solo describen el movimiento de un objeto basándose en el movimiento de contactos con el hardware multitáctil. Sin embargo, para moverse de forma natural, los objetos deben exhibir también características físicas como la elasticidad y la desaceleración. Cuando un usuario levanta todos los contactos de un objeto, el sistema de inercia proporciona eventos de manipulación adicionales a la aplicación para que la aplicación pueda manejar los eventos como si el usuario todavía estuviera moviendo el objeto con el tacto. Sin embargo, el sistema de inercia en realidad genera los eventos basándose en una simulación del comportamiento de los objetos. Si el usuario mueve un objeto a otro objeto, el sistema de inercia envía eventos de manipulación basándose en las características límites de los objetos. Por lo tanto, el sistema de inercia proporciona un movimiento más realista para los objetos de aplicación que un usuario manipula utilizando el hardware multitáctil y la API proporciona una sensación consistente de las manipulaciones entre aplicaciones.

20 Este Sumario se proporciona para presentar una selección de conceptos de forma simplificada que se describen más adelante en la Descripción detallada. Este Sumario no tiene la intención de identificar características clave o características esenciales de la materia objeto reivindicada, ni pretende ser utilizado para limitar el alcance de la materia objeto reivindicada.

**Breve descripción de los dibujos**

25 La Figura 1 es un diagrama de bloques que ilustra los componentes del sistema de inercia, en una realización.  
 La Figura 2 es un diagrama de flujo de datos que ilustra un entorno operativo convencional del sistema de inercia y el flujo de datos entre componentes, en una realización.  
 La Figura 3 es un diagrama de visualización que ilustra un objeto de aplicación manipulado por el tacto del usuario, en una realización. La Figura 4 es un diagrama de flujo que ilustra el procesamiento del bucle de entrada de una aplicación multitáctil que usa el sistema de inercia para manejar eventos de manipulación, en una realización.  
 30 La Figura 5 es un diagrama de flujo que ilustra el procesamiento del sistema de inercia cuando el sistema recibe una entrada táctil, en una realización.  
 La Figura 6 es un diagrama de flujo que ilustra el procesamiento de una aplicación multitáctil que usa el sistema de inercia para procesar eventos de inercia, en una realización.  
 35 La Figura 7 es un diagrama de flujo que ilustra el procesamiento del componente de simulación del sistema de procesamiento de inercia, en una realización.

**Descripción detallada**

El sistema de inercia proporciona una plataforma y API común para que las aplicaciones extiendan la entrada recibida de varios dispositivos de hardware multitáctiles para simular el comportamiento de los objetos en el mundo real. Por ejemplo, los objetos del mundo real no suelen dejar de moverse cuando un usuario deja de empujarlos, sino que exhiben algo de inercia y continúan moviéndose hasta que la fricción los frena finalmente hasta detenerse. En algunas realizaciones, la entrada táctil pasa primero por un procedimiento para interpretar el movimiento de uno o más contactos como manipulaciones. Las manipulaciones se asignan más directamente a las intenciones del usuario que las entradas táctiles individuales y agregan soporte para la transformación básica de objetos utilizando múltiples contactos táctiles. Una aplicación puede utilizar manipulaciones para admitir el giro, el cambio de tamaño y la traslación de varios objetos (por ejemplo, fotos) al mismo tiempo. Las manipulaciones pueden describirse como transformaciones afines bidimensionales (2D) que contienen información de giro, escala (por ejemplo, zoom) y traslación (por ejemplo, panorámica).

Cada toque del hardware multitáctil se denomina contacto. Por ejemplo, cuando un usuario coloca su dedo en el hardware multitáctil, mueve su dedo y levanta el dedo, esa serie de eventos es un solo contacto. Por ejemplo, si el usuario mueve dos contactos más juntos o más separados, el sistema puede determinar que el usuario está escalando (por ejemplo, acercándose a o alejándose de) un objeto. Como otro ejemplo, si el usuario mueve múltiples contactos en un movimiento circular, entonces el sistema puede interpretar el movimiento como un giro de un objeto. Cada aplicación puede definir objetos que son relevantes de forma diferente, por lo que depende de la aplicación adjuntar una instancia del sistema (denominado procesador de manipulación) a cada objeto que un usuario puede manipular mediante la entrada táctil dentro de la aplicación. Por ejemplo, una aplicación de exploración de fotos puede adjuntar un procesador de manipulación a cada foto mostrada, de modo que el usuario pueda mover las fotos, escalarlas, girarlas, etc.

Las manipulaciones manejadas por la aplicación solo describen el movimiento de un objeto basándose en el movimiento de los contactos. Sin embargo, para moverse de forma natural, los objetos deben exhibir también características físicas como la elasticidad y la desaceleración. Cuando un usuario levanta todos los contactos de un objeto, el sistema de inercia proporciona eventos de manipulación adicionales a la aplicación para que la aplicación pueda manejar los eventos como si el usuario todavía estuviera moviendo el objeto con el tacto. Sin embargo, el sistema de inercia en realidad genera los eventos basándose en una simulación del comportamiento de los objetos. Por ejemplo, si el usuario ha levantado los contactos mientras el objeto tenía una velocidad en una dirección particular, entonces el sistema de inercia continúa enviando eventos que indican que el objeto se está moviendo en esa dirección, disminuyendo con el tiempo a medida que el objeto desacelera. Si el usuario mueve un objeto en otro objeto, como el borde de la pantalla, el sistema de inercia envía eventos de manipulación basándose en las características límites de los objetos. Por ejemplo, si el autor de una aplicación define dos objetos como elásticos, entonces los dos objetos pueden rebotar entre sí cuando un usuario mueve los objetos entre sí. Por lo tanto, el sistema de inercia proporciona un movimiento más realista para los objetos de aplicación que un usuario manipula utilizando el hardware multitáctil y la API proporciona una sensación consistente de las manipulaciones entre aplicaciones.

La Figura 1 es un diagrama de bloques que ilustra los componentes del sistema de inercia, en una realización. El sistema 100 de inercia incluye una interfaz 110 de hardware, uno o más procesadores 120 de manipulación, un componente 130 de transformación de entrada, un componente 140 de simulación y una interfaz 150 de aplicación. Cada uno de estos componentes se describe con más detalle en el presente documento.

La interfaz 110 de hardware se comunica con el hardware para recibir contactos y movimientos táctiles. La interfaz 110 de hardware puede incluir varios subcomponentes que trabajan juntos para proporcionar información de entrada táctil. Por ejemplo, el sistema operativo puede proporcionar un modelo de controlador común para los fabricantes de hardware multitáctil para proporcionar información táctil para su hardware particular. El sistema operativo puede traducir la información táctil recibida a través de este modelo en mensajes de ventana (por ejemplo, WM\_TOUCH descrito en el presente documento) y pasar estos mensajes a la aplicación. Por lo tanto, la interfaz 110 de hardware puede implicar la coordinación del hardware, un controlador de hardware y una capa del sistema operativo. El resultado es una serie de mensajes al sistema de inercia que identifican un contacto particular (por ejemplo, el toque de un dedo) y las coordenadas del contacto a lo largo del tiempo. Por ejemplo, el sistema operativo puede proporcionar un mensaje cuando se establece un nuevo contacto en el hardware multitáctil, un mensaje cada vez que el contacto se mueve y un mensaje cuando el contacto se levanta del hardware multitáctil.

Uno o más procesadores 120 de manipulación usan el componente 130 de transformación de entrada para interpretar el movimiento de cada contacto asociado con un objeto de aplicación particular. El procesador 120 de manipulación puede determinar que un usuario está usando múltiples contactos para realizar una sola acción. Por ejemplo, un usuario podría tocar una foto con los cinco dedos de una mano y girar su mano para indicar una intención de girar la foto. El procesador 120 de manipulación recibe cinco contactos separados (uno para cada dedo) y el cambio en las coordenadas de cada contacto a medida que el usuario gira su mano. El procesador 120 de manipulación determina que cada contacto está agarrando el mismo objeto y realizando el mismo giro. El sistema informará a la aplicación que el usuario ha girado el objeto, pero la aplicación puede ignorar si el usuario usó dos, cinco o cualquier número particular de dedos u otros contactos para realizar el giro. Esto simplifica enormemente la creación de la aplicación porque el autor de la aplicación puede manejar esos tipos de manipulaciones que son relevantes para la aplicación y dejar que el sistema de inercia interprete el significado de cada entrada táctil de bajo nivel recibida del hardware multitáctil.

El procesador 120 de manipulación usa el componente 130 de transformación de entrada para hacer determinaciones sobre el significado de los movimientos recibidos de varios contactos, tanto solos como en concierto. Por ejemplo, si un usuario está manipulando una foto con dos dedos, lo que crea dos contactos de entrada correspondientes, entonces el procesador 120 de manipulación usa el componente 130 de transformación de entrada para determinar el significado de los movimientos relativos entre los dos contactos. Si los dos contactos se separan, entonces el componente 130 de transformación de entrada puede determinar que el usuario está escalando el objeto para cambiar el tamaño del objeto. Si los dos contactos giran, entonces el componente 130 de transformación de entrada puede determinar que el usuario está girando el objeto. Si los dos contactos se deslizan en una dirección particular, entonces el componente 130 de transformación de entrada puede determinar que el usuario está desplazando el objeto a una nueva ubicación. Aunque cada tipo de movimiento se describe por separado, tenga en cuenta que un usuario puede realizar los tres tipos de movimientos al mismo tiempo, y el procesador de transformación de entrada puede informar la transformación general a la aplicación. Por ejemplo, un usuario puede girar, escalar y desplazar un objeto todo en un solo movimiento.

El componente 140 de simulación simula el movimiento continuo de un objeto de aplicación después de que el usuario deja de tocar el objeto basándose en los parámetros de inicialización y en las restricciones definidas para el objeto. Una aplicación puede inicializar el componente 140 de simulación con el estado final del procesador 120 de manipulación asociado con el objeto. La aplicación puede definir también varias características del objeto, como el comportamiento de los límites del objeto. El componente 140 de simulación usa técnicas basándose en la física para simular el comportamiento del objeto durante un período después de que el usuario suelta el objeto. Por ejemplo, el componente 140 de simulación puede continuar activando notificaciones a la aplicación en la misma forma que los eventos de manipulación recibidos por la aplicación mientras el usuario estaba moviendo el objeto. La aplicación puede centrarse en reaccionar al movimiento del objeto en lugar de preocuparse por las acciones (físicas o de usuario) que causaron el movimiento del objeto. Los expertos en la materia reconocerán numerosas técnicas bien conocidas para

simular el comportamiento físico equivalente de los objetos virtuales en el software.

La interfaz 150 de aplicación se comunica con la aplicación para recibir información y proporcionar transformaciones de manipulación a la aplicación. La interfaz 150 de aplicación recibe información de inicialización de la aplicación. La información de inicialización puede especificar qué tipos de transformaciones admite el objeto de aplicación para un objeto particular y el procesador de manipulación asociado, así como los datos de inicialización para el componente 140 de simulación cuando el usuario ya no está moviendo el objeto. Por ejemplo, algunos objetos de aplicación pueden admitir escalado pero no giro. La información de inicialización puede especificar también un punto de pivote del objeto. El sistema de inercia proporciona transformaciones de manipulación a la aplicación a través de la interfaz de la aplicación. Por ejemplo, cuando el sistema de inercia recibe una entrada táctil de bajo nivel que el sistema interpreta como una transformación reconocida (por ejemplo, un giro), el sistema activa un evento para notificar a la aplicación sobre la manipulación. La aplicación procesa la transformación de manipulación para modificar el objeto en función de la transformación. Por ejemplo, si el usuario ha girado el objeto, entonces la aplicación puede almacenar la nueva orientación del objeto para usar la próxima vez que la aplicación muestre el objeto. Como otro ejemplo, si el objeto continúa girando después de que el usuario lo soltó basándose en los cálculos del componente 140 de simulación, entonces la aplicación puede almacenar la nueva orientación del objeto.

El dispositivo informático en el que se implementa el sistema puede incluir una unidad central de procesamiento, memoria, dispositivos de entrada (por ejemplo, teclado y apuntadores), dispositivos de salida (por ejemplo, dispositivos de visualización) y dispositivos de almacenamiento (por ejemplo, unidades de disco). Los dispositivos de memoria y almacenamiento son medios legibles por ordenador que pueden estar codificados con instrucciones ejecutables por ordenador que implementan el sistema, lo que significa un medio legible por ordenador que contiene las instrucciones. Además, las estructuras de datos y las estructuras de mensajes pueden almacenarse o transmitirse a través de un medio de transmisión de datos, como una señal en un enlace de comunicación. Se pueden utilizar varios enlaces de comunicación, como Internet, una red de área local, una red de área amplia, una conexión de acceso telefónico de punto a punto, una red de teléfono celular, etc.

Las realizaciones del sistema pueden implementarse en varios entornos operativos que incluyen ordenadores personales, ordenadores de servidor, dispositivos portátiles o de mano, sistemas multiprocesadores, sistemas basados en microprocesadores, electrónica de consumo programable, cámaras digitales, PC en red, miniordenadores, ordenadores centrales, entornos de computación distribuidos que incluyen cualquiera de los sistemas o dispositivos anteriores, y así sucesivamente. Los sistemas informáticos pueden ser teléfonos celulares, asistentes digitales personales, teléfonos inteligentes, ordenadores personales, productos electrónicos de consumo programables, cámaras digitales, etc.

El sistema puede describirse en el contexto general de instrucciones ejecutables por ordenador, tales como módulos de programa, ejecutados por uno o más ordenadores u otros dispositivos. En general, los módulos de programa incluyen rutinas, programas, objetos, componentes, estructuras de datos, etc. que realizan tareas particulares o implementan tipos de datos abstractos particulares. Normalmente, la funcionalidad de los módulos de programa puede combinarse o distribuirse según se desee en diversas realizaciones.

La Figura 2 es un diagrama de flujo de datos que ilustra un entorno operativo convencional del sistema de inercia y el flujo de datos entre componentes, en una realización. Un dispositivo de hardware multitáctil produce entradas 210 a través de una interfaz de hardware. Por ejemplo, el hardware puede enviar las entradas 210 a un sistema operativo a través de un controlador de software proporcionado por el fabricante del hardware. La interfaz de hardware proporciona eventos 220 de entrada a una aplicación 230. Por ejemplo, una aplicación puede informar al sistema operativo que la aplicación 230 admite entrada de usuario multitáctil y registrarse para recibir mensajes relacionados con la entrada de usuario multitáctil. La aplicación 230 recibe información de entrada táctil de bajo nivel como cambios 240 de entrada y reenvía los cambios 240 de entrada a un sistema 250 de manipulación. Por ejemplo, los cambios 240 de entrada pueden describir cada movimiento de uno o más contactos táctiles con el hardware utilizando un conjunto de coordenadas que indican la posición actual de cada contacto y otras características de movimiento. El sistema 250 de manipulación interpreta los cambios 240 de entrada y notifica a la aplicación 230 el uno o más eventos 260 de manipulación que indican manipulaciones de nivel superior que el usuario está realizando en un objeto visualizado. Por ejemplo, si el movimiento de los contactos indica que el usuario tiene la intención de girar el objeto, los eventos 260 de manipulación indican un grado de giro.

Cuando el usuario termina de mover el objeto (por ejemplo, cuando la aplicación recibe una notificación de que cada contacto que toca un objeto se ha eliminado del hardware táctil), la aplicación 230 envía información 270 de inicialización al sistema 280 de inercia. El sistema 280 de inercia determina la siguiente posición del objeto y proporciona eventos 290 de inercia similares a los eventos 260 de manipulación que el sistema 250 de manipulación ha proporcionado cuando el usuario estaba moviendo el objeto. La aplicación 230 proporciona también un temporizador de activación para llamar periódicamente al sistema 280 de inercia para proporcionar la siguiente posición del objeto a través de los eventos 290 de inercia. La aplicación 230 procesa los eventos de inercia de manera similar a los eventos de manipulación.

Aunque el diagrama ilustra que la aplicación recibe primero la entrada táctil y pasa la entrada táctil al sistema de manipulación y la inercia, en algunas realizaciones, estos sistemas reciben entrada táctil directamente desde la interfaz

del hardware, interpretan la entrada táctil y proporcionan eventos de manipulación interpretados a la aplicación. Del mismo modo, la aplicación puede no saber que un sistema 280 de inercia separado proporciona eventos de inercia después de que un usuario deja de mover un objeto con el tacto, sino que puede recibir eventos de una interfaz durante el tiempo en que el usuario está moviendo el objeto y después cuando el objeto se está moviendo basándose en la inercia. Esto representa una arquitectura alternativa que proporciona una funcionalidad resultante similar pero le da a la aplicación menos control sobre el procesamiento de la entrada. Por ejemplo, la aplicación puede no ser capaz de definir objetos de aplicación individuales a los que el sistema conecta procesadores de manipulación individuales. El complemento RTS descrito aquí es un ejemplo de esta arquitectura alternativa para el sistema.

La Figura 3 es un diagrama de visualización que ilustra un objeto de aplicación manipulado por el tacto del usuario, en una realización. Una aplicación puede mostrar y recibir simultáneamente entradas táctiles para muchos de estos objetos. Por ejemplo, una aplicación de shell del sistema operativo puede mostrar uno o más objetos de documentos almacenados en el escritorio del ordenador del usuario. En la pantalla 310, el objeto 320 de documento representa un documento en el escritorio del usuario que el usuario desea arrastrar a la papelera 330 de reciclaje usando el tacto. El usuario realiza un movimiento de desplazamiento del objeto 320 de documento que da como resultado que el sistema procese una primera ubicación 340 de manipulación, una segunda ubicación 350 de manipulación y una tercera ubicación 360 de manipulación. La aplicación recibe la primera ubicación 340 de manipulación cuando el usuario toca inicialmente el objeto 320 de documento con uno o más dedos (es decir, contactos). La aplicación recibe la segunda ubicación 350 de manipulación cuando el usuario desliza sus dedos por la pantalla. La aplicación recibe la tercera ubicación de manipulación cuando el usuario levanta los dedos de la pantalla. Las flechas 365 representan los vectores del movimiento del objeto 320 de documento.

Sin inercia, el objeto 320 de documento se detendría en la tercera ubicación 360 de manipulación, que probablemente no sea lo que el usuario pretende. El sistema de inercia proporciona ubicaciones de manipulación adicionales a la aplicación como si el usuario todavía estuviera tocando y moviendo el objeto 320 de documento basándose en la velocidad 320 del objeto del documento cuando el usuario suelta el objeto 320 del documento. La aplicación recibe la primera ubicación de manipulación basándose en la inercia 370 cuando la aplicación inicializa el sistema de inercia y llama a la función de procesamiento del sistema de inercia por primera vez. La aplicación recibe la segunda ubicación de manipulación basándose en la inercia 380 a medida que la aplicación continúa llamando a la función de procesamiento del sistema de inercia. Debido a que la ubicación 380 de manipulación final del objeto 320 de documento está sobre la papelera 330 de reciclaje, la aplicación procesa el contacto entre los dos objetos (por ejemplo, colocando el objeto 320 de documento en la papelera 330 de reciclaje). En el ejemplo ilustrado, aunque el sistema de inercia desacelera el movimiento del objeto 320 de documento, el objeto 320 de documento todavía puede moverse una distancia considerable a través de la pantalla 310 en función de la alta velocidad inicial del movimiento del usuario del objeto 320 de documento al principio.

La Figura 4 es un diagrama de flujo que ilustra el procesamiento del bucle de entrada de una aplicación multitáctil que usa el sistema de inercia para manejar eventos de manipulación, en una realización. En el bloque 410, la aplicación recibe una entrada táctil de bajo nivel. Por ejemplo, un sistema operativo o instancia del sistema de inercia recibe información de contacto táctil del hardware multitáctil y reenvía la información de contacto táctil a la aplicación. En el bloque 420, la aplicación identifica el objeto al que se aplica la entrada. Por ejemplo, la aplicación puede probar las coordenadas de la entrada recibida comparando las coordenadas con las coordenadas de cada objeto de aplicación que se muestra en la aplicación. Si la entrada táctil está dentro de los límites de un objeto de aplicación visualizado, entonces la aplicación determina que la entrada táctil se aplica a ese objeto. En el bloque 430, la aplicación envía la entrada táctil recibida y la información sobre el objeto de aplicación identificado a una API de manipulación para invocar el sistema de inercia (véase Figura 5). Por ejemplo, la aplicación puede crear un identificador numérico para cada objeto de aplicación y pasar el identificador numérico al sistema de inercia cada vez que la entrada táctil corresponde a ese objeto.

En el bloque 440, la aplicación recibe un evento de manipulación del sistema de inercia que describe una o más manipulaciones del objeto de aplicación identificado. Por ejemplo, la aplicación puede recibir un evento que describe una transformación afin 2D del objeto de aplicación. Tenga en cuenta que el bloque 440 se ilustra en serie después del bloque 430 para simplificar la ilustración. En la práctica, la aplicación puede recibir muchos eventos de entrada táctil antes de que el sistema de inercia notifique a la aplicación con un evento de manipulación. No hay necesariamente un mapeo uno a uno de los eventos de entrada táctil a los eventos de manipulación. Debido a que los eventos de manipulación representan una interpretación de nivel superior de las entradas táctiles de bajo nivel, las entradas táctiles múltiples pueden formar un solo evento de manipulación. En el bloque 450, la aplicación maneja el evento de manipulación recibido. Por ejemplo, si el evento de manipulación recibido es una rotación, entonces la aplicación puede girar el objeto de aplicación en la pantalla y almacenar la nueva ubicación de los objetos de aplicación para su uso cuando el objeto de aplicación se muestre nuevamente. El sistema de inercia libera a la aplicación de realizar pasos específicos para un dispositivo de hardware multitáctil particular o incluso de saber qué dispositivo de hardware está proporcionando la entrada multitáctil. Además, el sistema de inercia libera a la aplicación del procesamiento del movimiento de contacto individual y permite que la aplicación se centre en el procesamiento de transformaciones a nivel del objeto de aplicación.

En el bloque 460, la aplicación espera la siguiente entrada táctil. Por ejemplo, la aplicación puede llamar a un API de mensaje proporcionado por el sistema operativo, como GetMessage en Microsoft Windows que espera a que se envíe

el siguiente mensaje a la cola de mensajes de la aplicación. En el bloque de decisión 470, si la aplicación recibe la siguiente entrada táctil, la aplicación realiza un bucle en el bloque 410 para procesar la entrada, de lo contrario, la aplicación realiza un bucle en el bloque 460 para continuar esperando una entrada adicional. Cuando la aplicación se cierra, la aplicación sale del bucle de entrada (no mostrado).

5 La Figura 5 es un diagrama de flujo que ilustra el procesamiento del sistema de inercia cuando el sistema recibe una entrada táctil, en una realización. En el bloque 505, el sistema recibe entrada táctil junto con información que identifica un objeto de aplicación con el que está asociada la entrada táctil. Por ejemplo, la entrada táctil puede incluir coordenadas u otra información de ubicación de uno o más contactos táctiles, y la información del objeto de aplicación puede incluir un identificador que la aplicación ha asignado a un objeto en particular en el que la entrada táctil está sobre el hardware multitáctil. En el bloque 510, el sistema identifica un procesador de manipulación asociado con el objeto de aplicación. En el bloque 520 de decisión, si el sistema no ha asociado previamente un procesador de manipulación con el objeto de aplicación, entonces el sistema continúa en el bloque 530, de lo contrario, el sistema continúa en el bloque 540. En el bloque 530, el sistema crea un procesador de manipulación y lo asocia con el objeto de aplicación, luego continúa en el bloque 540.

15 En el bloque 540 de decisión, si la entrada táctil recibida indica que la aplicación recibió un nuevo contacto (por ejemplo, un evento de contacto), entonces el sistema continúa en el bloque 550, de lo contrario, el sistema continúa en el bloque 560. Por ejemplo, un usuario puede hacer contacto inicial de un dedo con un objeto en pantalla o colocar otro dedo (es decir, contacto) en un objeto previamente tocado. En el bloque 550, el sistema agrega el nuevo contacto a la lista de contactos asociados con el procesador de manipulación, y luego continúa en el bloque 560. En el bloque 560 de decisión, si la entrada táctil recibida indica que la aplicación recibió notificación de que se eliminó un contacto táctil (por ejemplo, un evento de retoque), entonces el sistema continúa en el bloque 570, de lo contrario, el sistema continúa en el bloque 580. Por ejemplo, el usuario puede levantar uno o más dedos de un objeto previamente tocado. En el bloque 570, el sistema elimina el contacto de la lista de contactos asociados con el procesador de manipulación y luego continúa en el bloque 580. En el bloque 580, el sistema procesa la entrada táctil para determinar cualquier manipulación representada por la entrada táctil. Por ejemplo, el movimiento táctil puede indicar una manipulación de giro o traslación, mientras que la eliminación del contacto táctil puede indicar la finalización de una manipulación. En el bloque 590, el sistema activa un evento de manipulación para enviar información de transformación que describe la manipulación a la aplicación. Por ejemplo, el sistema puede proporcionar un grado de rotación angular del objeto a la aplicación. Después del bloque 590, estas etapas concluyen.

30 La Figura 6 es un diagrama de flujo que ilustra el procesamiento de una aplicación multitáctil que usa el sistema de inercia para procesar eventos de inercia, en una realización. En el bloque 610, la aplicación determina que el usuario ha soltado un objeto. Por ejemplo, después del manejo de un evento de manipulación en el bloque 450 de la Figura 4, la aplicación puede recibir una indicación de que la manipulación está completa o que el usuario ha levantado todos los contactos que estaban tocando un objeto de aplicación. En el bloque 620, la aplicación inicializa el sistema de inercia. Por ejemplo, la aplicación puede pasar una referencia al procesador de manipulación que manejaba el movimiento del objeto y otra información de inicialización. En el bloque 630, la aplicación establece un temporizador que controlará el período de procesamiento de inercia del sistema de inercia. Por ejemplo, la aplicación puede establecer un temporizador que se activará cada 100 milisegundos para procesar el siguiente incremento de movimiento de un objeto. En el bloque 640, la aplicación llama a la función de procesamiento del sistema de inercia (véase Figura 7). Por ejemplo, el sistema de inercia puede proporcionar una función de "procedimiento" que la aplicación llama para informar al sistema de inercia que es hora de realizar la simulación para el período desde el último período de simulación.

45 En el bloque 650, la aplicación recibe uno o más eventos de inercia que describen manipulaciones del objeto (por ejemplo, giro, traslación y/o escala) en función de la inercia simulada. Por ejemplo, si el objeto viajaba en una dirección particular, la aplicación puede recibir un evento de inercia que describe una manipulación de traducción en esa dirección. Como otro ejemplo, si el objeto se estaba expandiendo cuando el usuario lo liberó, la aplicación puede recibir un evento de inercia que describe una manipulación de escala. Tenga en cuenta que el bloque 650 se ilustra en serie después del bloque 640 para simplificar la ilustración. En la práctica, la aplicación puede llamar a la función de procesamiento de inercia varias veces antes de que el sistema de inercia notifique a la aplicación con un evento de inercia. No hay necesariamente un mapeo uno a uno de las llamadas a la función de procesamiento y los eventos de inercia. Por otro lado, el sistema de inercia puede notificar la aplicación de múltiples eventos de inercia después de una sola llamada a la función de procesamiento.

55 En el bloque 660, la aplicación maneja el evento de inercia recibido en función del significado (por ejemplo, un efecto) de la manipulación en el contexto de la aplicación particular. Por ejemplo, si el evento de inercia recibido es un giro, entonces la aplicación puede rotar el objeto de aplicación en la pantalla y almacenar los objetos de aplicación en una nueva ubicación para usar cuando la aplicación vuelve a mostrar el objeto de aplicación. En el bloque 670 de decisión, si los eventos de inercia están completos, estas etapas concluyen, de lo contrario, el sistema continúa en el bloque 680. El sistema de inercia puede informar a la aplicación que una manipulación simulada particular se completa como un valor de retorno de la función del procedimiento o mediante las notificaciones proporcionadas a la aplicación (por ejemplo, a través de una interfaz de evento del Modelo de Objetos Componentes (COM)). En el bloque 680, la aplicación espera la próxima activación del temporizador, luego realiza un bucle en el bloque 640 y llama a la función de procesamiento del sistema de inercia.

La Figura 7 es un diagrama de flujo que ilustra el procesamiento del componente de simulación del sistema de procesamiento de inercia, en una realización. En el bloque 710, el componente recibe parámetros de simulación iniciales. Por ejemplo, una aplicación o procesador de manipulación puede proporcionar el estado final de un objeto de aplicación cuando un usuario dejó de tocar el objeto. En el bloque 720, el componente inicializa un motor de simulación que realiza cálculos basándose en la física para determinar el comportamiento de un objeto en función de los parámetros. Por ejemplo, el motor de simulación puede proporcionar un comportamiento realista de desaceleración o elasticidad para los objetos de aplicación puestos en movimiento por la entrada táctil del usuario. En el bloque 730, el componente recibe una llamada de procedimiento desde la aplicación. La aplicación u otro componente impulsan el procedimiento de simulación llamando repetidamente a una función de procesamiento a intervalos regulares para avanzar la simulación. El componente de simulación también puede generar internamente el temporizador.

En el bloque 740, el componente simula el movimiento del objeto en función de los parámetros iniciales, cualquier procesamiento anterior y el tiempo transcurrido desde la última llamada de procedimiento. La llamada de procedimiento puede proporcionar también una marca de tiempo que indica el tiempo que la aplicación desea que use la simulación. Esto permite que la aplicación simule el comportamiento de la aplicación en otro momento que no sea en tiempo real (por ejemplo, para pruebas o depuración de aplicaciones). En el bloque de decisión 750, si el movimiento se completa, entonces el componente continúa en el bloque 760, de lo contrario, el componente continúa en el bloque 770. El componente puede determinar que el movimiento se completa en función de factores como si el objeto todavía se está moviendo o si el movimiento del objeto ha caído por debajo de un cierto umbral. En el bloque 760, el componente establece un indicador de finalización en el próximo evento de inercia. En el bloque 770, el componente activa un evento de inercia para enviar información de transformación que describe el movimiento actual (por ejemplo, como una manipulación) a la aplicación. Por ejemplo, el sistema puede proporcionar un grado de rotación angular del objeto a la aplicación. Después del bloque 770, estas etapas concluyen.

En algunas realizaciones, el sistema de inercia recibe restricciones de objeto de aplicación. Por ejemplo, la aplicación puede definir la elasticidad de un objeto, el coeficiente de fricción (para determinar cómo se desacelera un objeto), las características límite del objeto, etc. Por ejemplo, el autor de una aplicación puede definir objetos rígidos que el usuario puede mover y un borde de ventana de aplicación que rebota, de modo que los objetos que se mueven al borde de la ventana rebotan desde el borde de la ventana cuando el usuario los suelta.

En algunas realizaciones, el sistema de inercia recibe información del estado del objeto inicial de un sistema de manipulación que estaba rastreando el movimiento del objeto cuando el usuario manipulaba el objeto con entrada táctil. Por ejemplo, el sistema de manipulación puede rastrear la posición actual de cada objeto, el movimiento histórico del objeto, la velocidad lineal y angular del objeto, y así sucesivamente. El autor de la aplicación puede proporcionar la salida de la manipulación al sistema de inercia para inicializar el sistema de inercia, de modo que el sistema de inercia pueda continuar sin problemas el movimiento pasado del objeto y ralentizarlo basándose en la física y las características apropiadas del objeto.

En algunas realizaciones, el sistema de inercia recibe límites en el movimiento de objetos desde la aplicación. Por ejemplo, el autor de la aplicación puede definir un límite superior en la distancia que un objeto puede moverse una vez que un usuario lo suelta. Como otro ejemplo, la aplicación puede definir un límite superior sobre cuánto tiempo puede moverse el objeto una vez que un usuario lo suelta. Estos y otros límites permiten al autor de la aplicación ajustar el sistema de inercia para adaptarse a los tipos de objetos manipulados por la aplicación y mejorar la experiencia del usuario con la aplicación.

En algunas realizaciones, el sistema de inercia no proporciona movimiento adicional para objetos con movimiento por debajo de un umbral predefinido. El umbral puede ser configurable por la aplicación. Por ejemplo, el sistema de inercia puede tener una velocidad lineal o angular de un objeto en particular por debajo de la que el sistema no continuará el movimiento del objeto después de que el usuario lo suelte. Si el objeto no se mueve muy rápido cuando el usuario lo suelta, el usuario puede esperar que el objeto se quede quieto y no continúe moviéndose. El umbral permite que la aplicación o el autor del sistema de inercia determinen el nivel de movimiento después de la manipulación que proporciona una buena experiencia de usuario.

En algunas realizaciones, el sistema de inercia recibe instrucciones para simular movimientos incrementales desde la aplicación. Por ejemplo, el sistema de inercia puede proporcionar una función de "procedimiento" o "DoWork" que la aplicación llama para indicar al sistema de inercia que realice una parte de la simulación general. El sistema de inercia puede esperar que la aplicación establezca un temporizador o, de lo contrario, llame periódicamente a la función para que el sistema de inercia simule el movimiento en el tiempo de acuerdo con una línea de tiempo natural. La aplicación puede afectar las características de los eventos de manipulación proporcionados por el sistema de inercia al variar la frecuencia con la que la aplicación llama a la función. En otras realizaciones, el sistema de inercia usa un temporizador interno para proporcionar eventos de manipulación en un horario regular hasta que cada objeto haya dejado de moverse (por ejemplo, debido a la desaceleración u otras fuerzas simuladas).

En algunas realizaciones, el sistema de inercia es parte de un sistema operativo basado en mensajes, y el sistema recibe mensajes relacionados con la entrada táctil que el sistema operativo recibe del hardware. Por ejemplo, al usar un paradigma similar a WM\_MOUSEMOVE para mensajes de ratón, las versiones futuras de Microsoft Windows pueden proporcionar un mensaje WM\_TOUCH que contiene información de movimiento táctil de bajo nivel recibida

del hardware multitáctil. El sistema operativo también puede proporcionar mensajes específicos, como WM\_TOUCHDOWN (cuando se hace un nuevo contacto con el hardware multitáctil), WM\_TOUCHMOVE (cuando se mueve un contacto existente) y WM\_TOUCHUP (cuando se levanta un contacto del hardware multitáctil). Una aplicación que recibe un mensaje relacionado con WM\_TOUCH puede invocar el sistema de inercia y pasar el mensaje al sistema de inercia para su interpretación y procesamiento. A continuación, la aplicación recibe eventos de nivel superior que representan la interpretación del sistema de inercia de la manipulación prevista por el usuario en función de la información de movimiento táctil de bajo nivel recibida.

En algunas realizaciones, el sistema de inercia recibe información de movimiento táctil de bajo nivel de hardware especializado, como un lápiz óptico en tiempo real. Por ejemplo, el Kit de desarrollo de software (SDK) de Microsoft PC tipo tableta proporciona un componente de lápiz óptico en tiempo real (RTS) que los autores de aplicaciones pueden ampliar con ganchos. Los ganchos RTS reciben entradas del hardware RTS y pueden realizar el procesamiento en la entrada recibida. El sistema de inercia puede proporcionar un gancho que una aplicación puede insertar en el componente RTS para procesar automáticamente RTS y otras entradas para manipular los objetos de aplicación como se describe en el presente documento. El enlace RTS proporciona una forma diferente para que el sistema de inercia reciba la entrada, pero el sistema de inercia interpreta la entrada y activa eventos a la aplicación que describen manipulaciones implicadas por la entrada como se ha descrito anteriormente. Un usuario puede usar una combinación de lápiz y entrada táctil. Por ejemplo, el usuario puede dibujar un objeto con el lápiz y girar después el objeto con los dedos.

En algunas realizaciones, el sistema de inercia es parte de un control común que una aplicación puede invocar para proporcionar una interfaz de usuario común. Microsoft Windows proporciona controles comunes para mostrar listas, árboles, botones, etc. Del mismo modo, el sistema de inercia puede proporcionar un control multitáctil para manipular objetos de aplicación de las formas descritas en el presente documento. Por ejemplo, el sistema puede proporcionar un control de dispersión que permite al usuario mostrar uno o más objetos y manipular los objetos. El control de dispersión maneja el procesamiento de la entrada táctil de bajo nivel y asocia la entrada con un objeto de aplicación particular, y la aplicación recibe eventos del control para manejar las manipulaciones de los objetos de aplicación. Por ejemplo, si el control indica que el usuario ha cambiado el tamaño de un objeto, entonces la aplicación puede almacenar el nuevo tamaño de los objetos.

En algunas realizaciones, el sistema de inercia realiza el procesamiento descrito en el presente documento en tres dimensiones. Aunque en el presente documento se describe un hardware multitáctil bidimensional, los expertos en la materia reconocerán que el procesamiento del sistema descrito en el presente documento puede aplicarse igualmente bien a manipulaciones tridimensionales (3D) si el hardware está disponible para proporcionar un movimiento coordinado en tres dimensiones. Por ejemplo, el hardware que detecta presión o usa cámaras para detectar el movimiento 3D de los dedos de un usuario podría proporcionar las coordenadas de movimiento en la tercera dimensión al sistema de inercia, y el sistema de inercia podría producir transformaciones 3D que describan manipulaciones (por ejemplo, giro, escala y traslación) de objetos en múltiples direcciones 3D.

La siguiente tabla define una API que el sistema de inercia proporciona a las aplicaciones para proporcionar movimiento basándose en la inercia a los objetos de aplicación después del movimiento táctil de los objetos por parte del usuario.

| <b>Propiedades:</b>            |   |
|--------------------------------|---|
| Límite inferior                | Limita qué tan lejos hacia la parte inferior de la pantalla puede moverse el objeto diana.          |
| Límite izquierdo               | Limita qué tan lejos hacia la izquierda de la pantalla puede moverse el objeto diana.               |
| Límite derecho                 | Limita qué tan lejos hacia la derecha de la pantalla puede moverse el objeto diana.                 |
| Límite superior                | Limita qué tan lejos hacia la parte superior de la pantalla puede moverse el objeto diana.          |
| Desaceleración angular deseada | Especifica la velocidad deseada para que el objeto diana deje de girar en radianes por milisegundo. |
| Desaceleración deseada         | Especifica la velocidad deseada a la que las operaciones de traducción se desacelerarán.            |

(continuación)

|   |   |
|---|---|
| <b>Propiedades:</b>   |   |
| Desplazamiento deseado  | Especifica la distancia deseada que recorrerá el objeto.  |
| Expansión deseada   | Especifica el cambio deseado en el radio promedio del objeto.   |
| Desaceleración de expansión deseada   | Especifica la velocidad a la que el objeto dejará de expandirse.  |
| Margen elástico inferior  | Especifica la región inferior para rebotar el objeto diana.   |
| Margen elástico izquierdo   | Especifica la región más a la izquierda para hacer rebotar el objeto diana.   |
| Margen elástico derecho   | Especifica la región más a la derecha para rebotar el objeto diana.   |
| Velocidad angular inicial   | Especifica el giro del objetivo cuando comienza el movimiento.  |
| Origen X inicial  | Obtiene o coloca la propiedad que designa la posición horizontal de un objeto diana. Esta propiedad especifica la ubicación horizontal inicial para un objetivo con inercia.  |
| Origen Y inicial  | Obtiene o coloca la propiedad que designa la ubicación vertical de un objeto diana. Esta propiedad especifica la ubicación vertical inicial para un objetivo con inercia.   |
| Radio inicial   | Especifica la distancia desde el borde del objetivo hasta su centro antes de cambiar el objeto.   |
| Marca de tiempo inicial   | Especifica la marca de tiempo de inicio para un objeto diana con inercia.   |
| Velocidad X inicial   | Especifica el movimiento inicial del objeto diana en el eje horizontal.   |
| Velocidad Y inicial   | Especifica el movimiento inicial del objeto diana en el eje vertical.   |
| <b>Procedimiento:</b>   |   |
| HRESULT Reset ();   | Inicializa el procesador con la marca de tiempo inicial.  |
| HRESULT Process([out] BOOL* completed);   | Realiza cálculos para el tick dado y puede aumentar el evento Delta o Completado dependiendo de si la extrapolación se ha completado o no. Si la extrapolación terminó en el tick anterior, el procedimiento es no operativo. |
| HRESULT ProcessTime([in] DWORD timestamp, [out] B); OOL* completed);  | Realiza cálculos para el tick dado y puede aumentar el evento Delta o Completado dependiendo de si la extrapolación se ha completado o no. Si la extrapolación terminó en el tick anterior, el procedimiento es no operativo. |
| HRESULT Complete();   | Genera el evento completado.  |
| <b>Procedimiento:</b>   |   |
| HRESULT CompleteTime([in] DWORD timestamp);   | Procesa la marca dada y genera el evento Completado.  |
| <b>Eventos:</b>   |   |
| HRESULT ManipulationStarted([in] FLOAT x, [in] FLOAT y);  | Maneja el evento para cuando comienza una manipulación.   |
| HRESULT ManipulationDelta([in] FLOAT x, [in] FLOAT y, [in] FLOAT translationDeltaX, [in] FLOAT translationDeltaY, [in] FLOAT scaleDelta, [in] FLOAT expansionDelta, [in] FLOAT rotationDelta, [in] FLOAT cumulativeTranslationX, [in] FLOAT cumulativeTranslationY, [in] FLOAT cumulativeScale, [in] FLOAT cumulativeExpansion, [in] FLOAT cumulativeRotation); | Maneja eventos que suceden cuando cambia un objeto manipulado   |

(continuación)

| Propiedades:  |   |
|---|---|
| HRESULT<br>ManipulationCompleted(<br>[in] FLOAT x,<br>[in] FLOAT y,<br>[in] FLOAT<br>cumulativeTranslationX,<br>[in] FLOAT<br>cumulativeTranslationY,<br>[in] FLOAT<br>cumulativeScale,<br>[in] FLOAT<br>cumulativeExpansion,<br>[in] FLOAT<br>cumulativeRotation<br>); | Maneja el evento cuando finaliza la manipulación. |

En la tabla anterior, el sistema de inercia puede proporcionar los eventos enumerados en la misma interfaz en la que una aplicación estaba recibiendo eventos previamente en función del movimiento del usuario.

5 De lo anterior, se apreciará que las realizaciones específicas del sistema de inercia se han descrito en el presente documento con fines ilustrativos, pero que pueden realizarse diversas modificaciones sin desviarse del ámbito de la invención tal como se define en las reivindicaciones adjuntas. Por ejemplo, aunque el sistema se ha descrito en el contexto de manipulaciones multitáctiles, el sistema proporciona una simulación de inercia que podría usarse en otros contextos, como juegos y otras áreas donde la simulación se usa comúnmente. Por consiguiente, la invención no está limitada excepto por las reivindicaciones adjuntas.

10

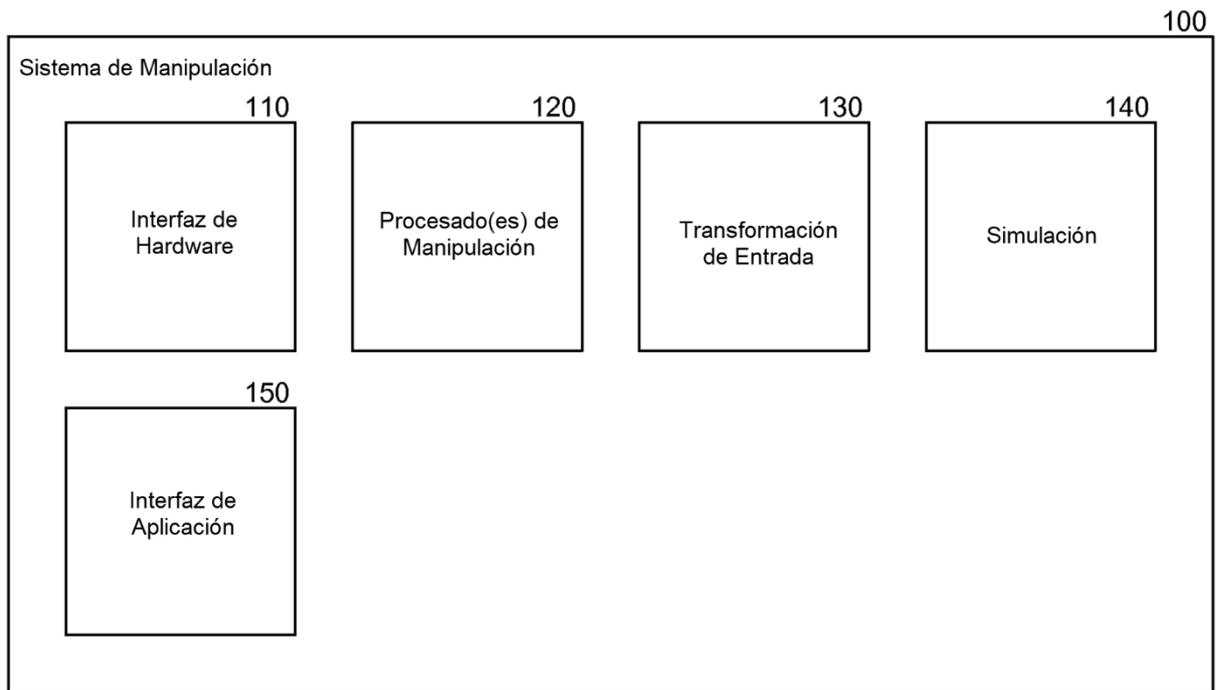
**REIVINDICACIONES**

1. Un procedimiento implementado por ordenador para proporcionar un movimiento realista de objetos manipulados mediante entrada multitáctil, comprendiendo el procedimiento las siguientes etapas realizadas por una aplicación:
  - 5 determinar (610) que un usuario ha liberado un objeto de aplicación definido por la aplicación eliminando uno o más contactos de un dispositivo de entrada multitáctil;
  - invocar (640) una interfaz de programación de aplicaciones API de inercia para procesar una simulación de movimiento del objeto de aplicación durante un período de procesamiento de inercia actual, en el que la API de inercia proporciona una plataforma independiente de la aplicación para simular movimientos realistas independientes de un tipo de objeto de aplicación;
  - 10 recibir (650) un evento de inercia generado por la API de inercia que describe una manipulación del objeto basándose en la inercia simulada, en el que recibir el evento de inercia comprende recibir información que describe una transformación afín 2D del objeto de aplicación; y
  - manejar (660) el evento de inercia recibido basándose en un efecto de la manipulación en un contexto de la aplicación modificando el objeto de aplicación, y
  - 15 comprendiendo además la API de inercia que recibe, desde la aplicación, una marca de tiempo que indica el tiempo que la aplicación desea que use la simulación.
  
2. El procedimiento de la reivindicación 1, que comprende además, después de determinar que el usuario ha liberado el objeto, inicializar el sistema de inercia pasando una referencia a un procesador de manipulación que estaba manejando el movimiento del objeto antes de que el usuario libere el objeto o después de determinar que el usuario ha liberado el objeto, inicializar el sistema de inercia pasando uno o más parámetros que describen un estado del objeto cuando el usuario ha liberado el objeto.
  
3. El procedimiento de la reivindicación 1, en el que la manipulación descrita por el evento de inercia incluye al menos una de una manipulación de giro, una manipulación de traslación y una manipulación de escala o en el que el evento de inercia recibido es un giro y en el que manejar el evento de inercia recibido comprende girar la aplicación objeto en una pantalla.
  
4. El procedimiento de la reivindicación 1, que comprende además establecer un temporizador que determina una duración del período de procesamiento de inercia, y en el que la aplicación invoca la API de inercia en cada activación del temporizador, en particular comprendiendo además recibir una indicación de la API de inercia de que la simulación se ha completado y el temporizador expira cuando se recibe la indicación.
  
- 30 5. El procedimiento de la reivindicación 1, en el que recibir el evento de inercia comprende recibir una notificación a través de una interfaz de evento COM.
  
6. Un sistema informático para manejar la entrada táctil desde hardware multitáctil, comprendiendo el sistema:
  - una interfaz (110) de hardware configurada para comunicarse con el hardware multitáctil para recibir información de contacto táctil y movimientos de los contactos táctiles;
  - 35 uno o más procesadores (120) de manipulación configurados para gestionar la interpretación del movimiento de cada contacto asociado con un objeto de aplicación particular definido por una aplicación;
  - un componente (130) de transformación de entrada configurado para interpretar un significado de los movimientos recibidos de varios contactos para producir manipulaciones de objetos de aplicación;
  - un componente (140) de simulación configurado para simular el movimiento continuo del objeto de aplicación después de que un usuario deja de tocar el objeto; y
  - 40 una interfaz (150) de aplicación configurada para comunicarse con la aplicación para recibir información de movimiento de contacto y proporcionar transformaciones afines 2D a la aplicación,
  - en el que el componente (140) de simulación está configurado además para recibir, desde la aplicación, una marca de tiempo que indica un tiempo que la aplicación desea que use la simulación.
  
- 45 7. El sistema de la reivindicación 6, en el que el componente de simulación está configurado además para recibir parámetros de inicialización desde una aplicación a través de la interfaz de la aplicación, en particular en el que los parámetros de inicialización son proporcionados por un procesador de manipulación que estaba manejando el objeto de aplicación antes de que el usuario dejara de tocar el objeto.
  
8. El sistema de la reivindicación 6, en el que el componente de simulación está configurado además para recibir restricciones de límites del objeto de aplicación o en el que el componente de simulación está configurado además para aplicar leyes de la física para determinar el movimiento continuo del objeto basándose en una velocidad y dirección de movimiento anteriores del objeto.
  
- 50 8. El sistema de la reivindicación 6, en el que el componente de simulación está configurado además para recibir restricciones de límites del objeto de aplicación o en el que el componente de simulación está configurado además para aplicar leyes de la física para determinar el movimiento continuo del objeto basándose en una velocidad y dirección de movimiento anteriores del objeto.
  
9. El sistema de la reivindicación 6, en el que el componente de transformación de entrada y el componente de simulación generan eventos en un formato similar para que la aplicación pueda manejar eventos de ambos componentes de forma similar.
  
- 55 10. Un medio de almacenamiento legible por ordenador codificado con instrucciones para controlar un sistema

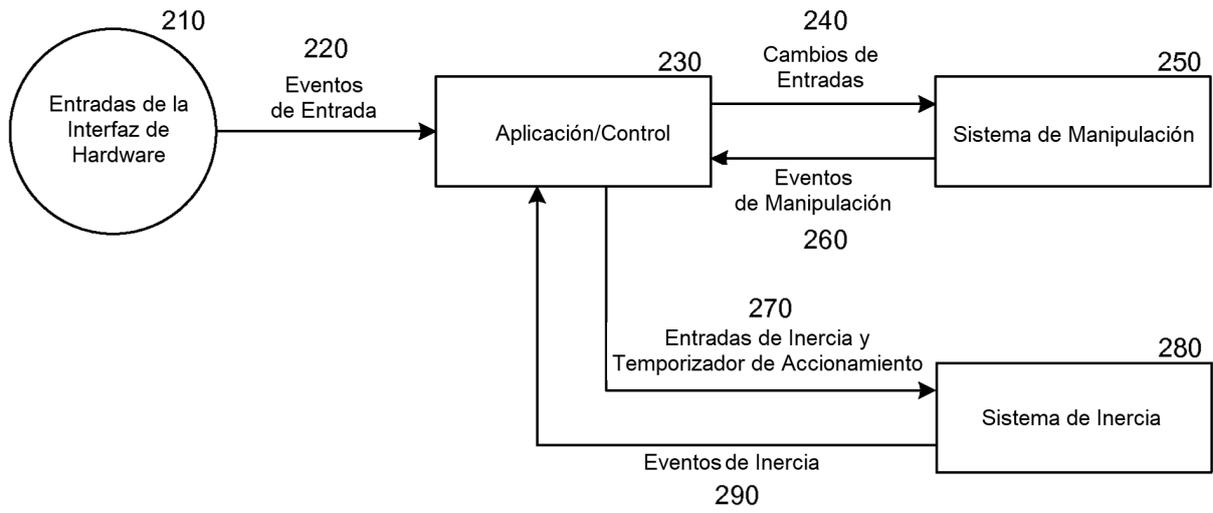
informático para simular el movimiento de un objeto de aplicación definido por una aplicación y previamente movido por entrada multitáctil, mediante un procedimiento que comprende las etapas siguientes realizadas por un componente (140) de simulación:

- 5 recibir (710) uno o más parámetros de simulación iniciales que proporcionan un último estado del objeto de aplicación cuando un usuario suelta el objeto de aplicación;  
inicializar (720) un motor de simulación que realiza cálculos basándose en la física para determinar el comportamiento del objeto de aplicación basándose en los parámetros de simulación iniciales;  
10 recibir (730), desde la aplicación, una indicación de que ha pasado el período de simulación actual para hacer avanzar una simulación;  
15 Simular (740) el movimiento del objeto de aplicación basándose en los parámetros iniciales, cualquier procesamiento previo y el tiempo transcurrido desde cualquier período de simulación anterior; y  
activar (770) un evento de inercia para enviar información de transformación afín 2D que describe un movimiento actual del objeto de aplicación a la aplicación,  
comprendiendo además recibir, desde la aplicación, una marca de tiempo que indica el tiempo que la aplicación desea que use la simulación.

11. El medio legible por ordenador de la reivindicación 10, en el que el motor de simulación simula un comportamiento de desaceleración realista para el objeto de aplicación después de que el usuario lo puso en movimiento al tocar el objeto.
- 20 12. El medio legible por ordenador de la reivindicación 10, en el que el motor de simulación simula un comportamiento de elasticidad realista para el objeto de aplicación cuando el objeto de aplicación se superpone con otro elemento en una pantalla.
13. El medio legible por ordenador de la reivindicación 10, en el que recibir la indicación de que ha pasado el período de simulación actual comprende recibir una llamada de la aplicación a una función de procesamiento de simulación.
- 25 14. El medio legible por ordenador de la reivindicación 11, que comprende además determinar si la simulación se ha completado y si la simulación se ha completado, informar a la aplicación que la simulación se ha completado.



**Figura 1**



**Figura 2**

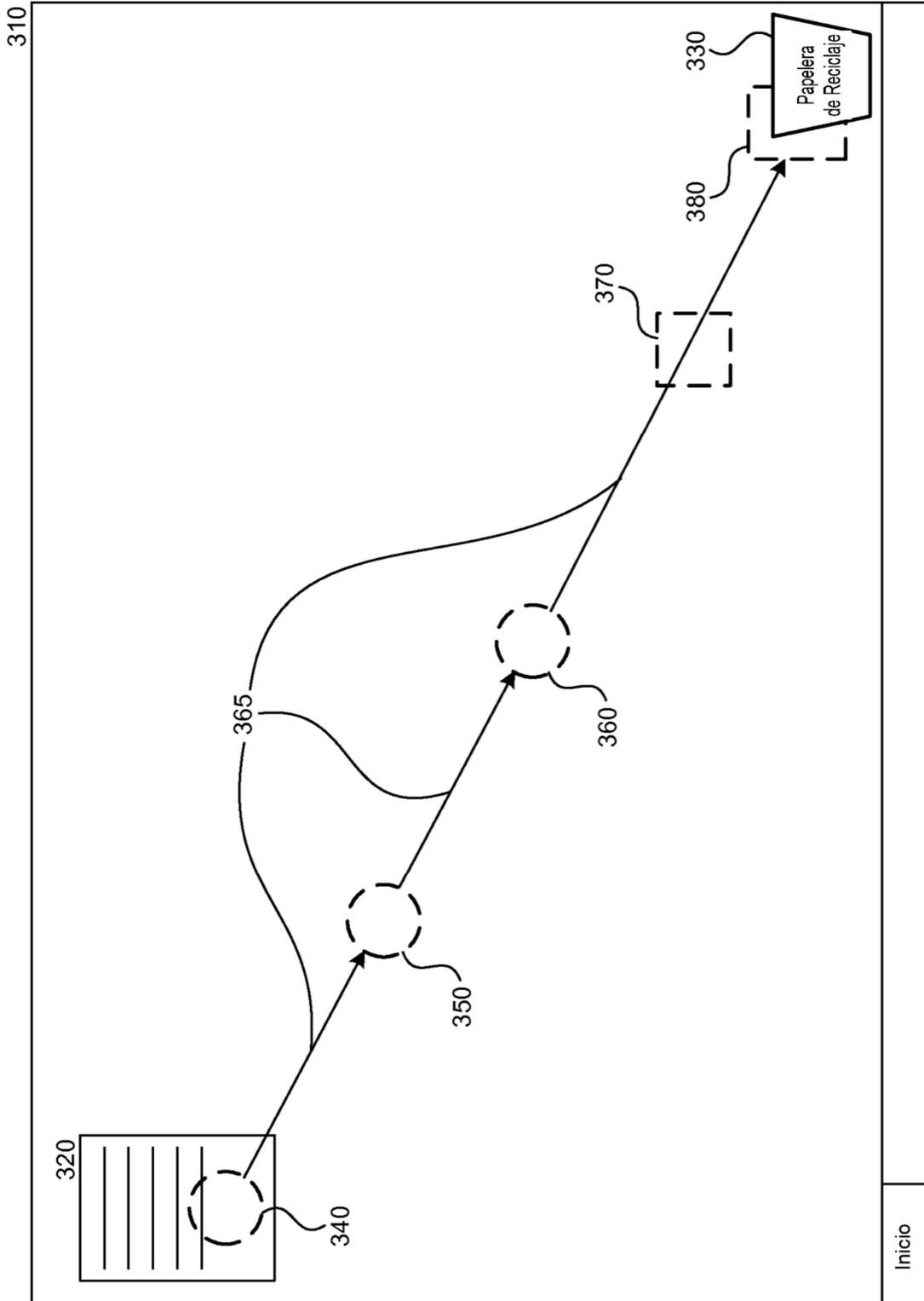
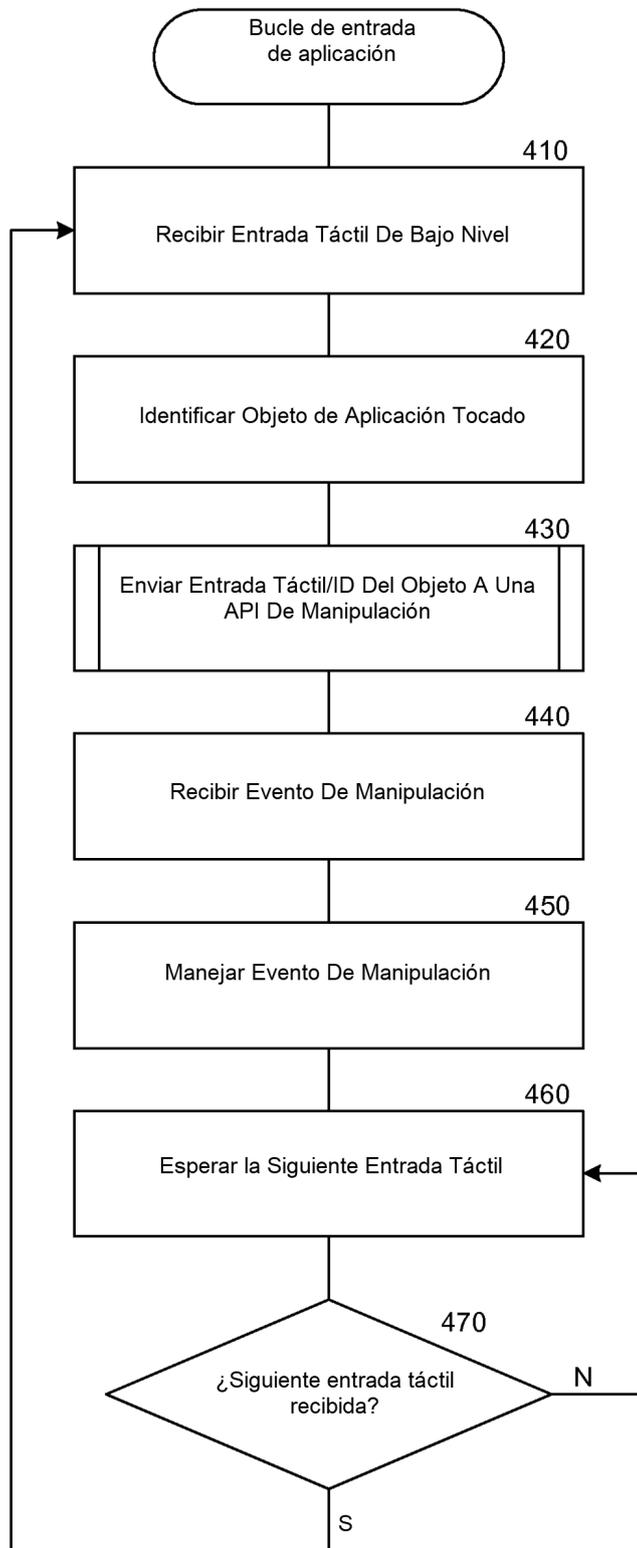
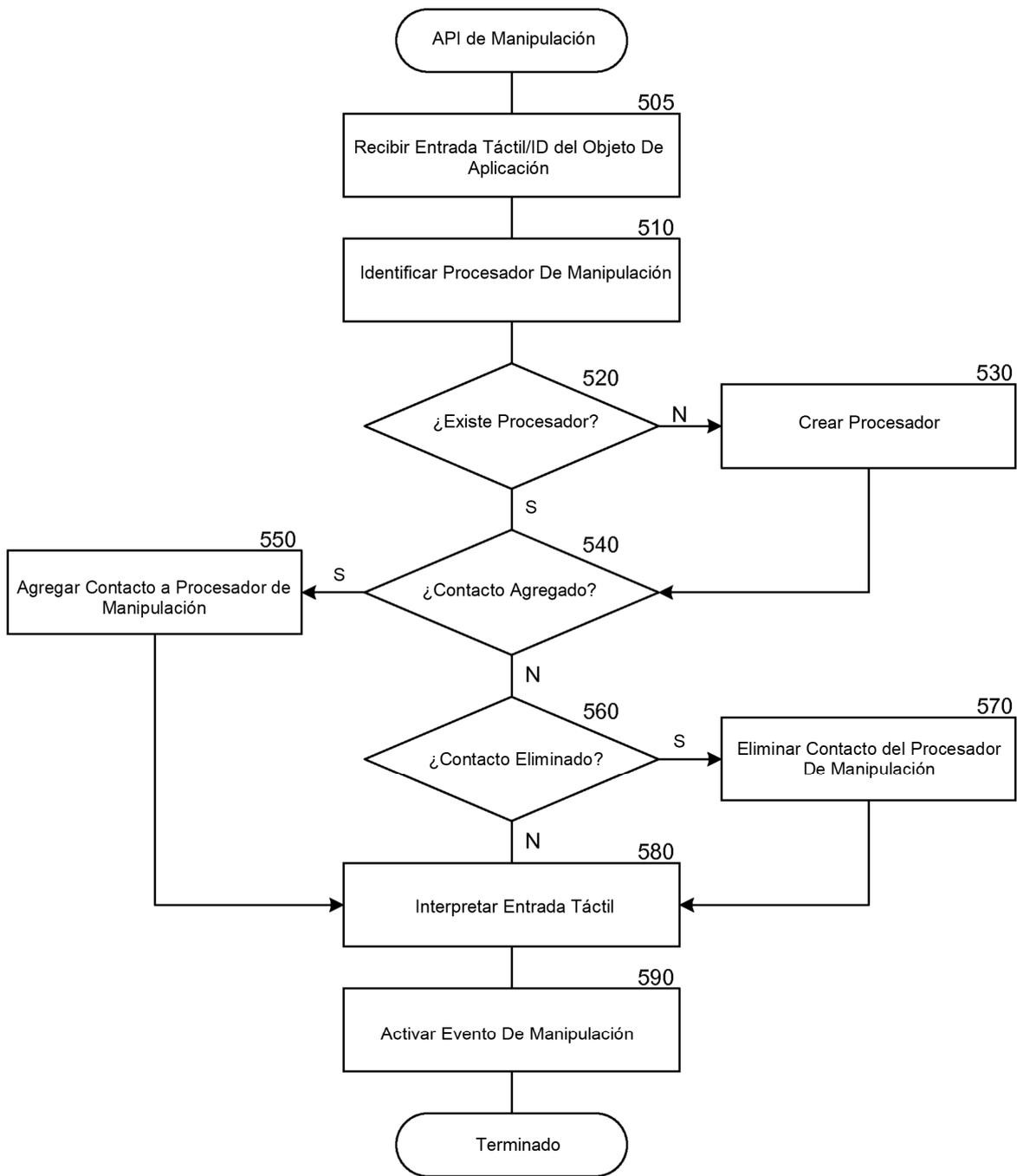


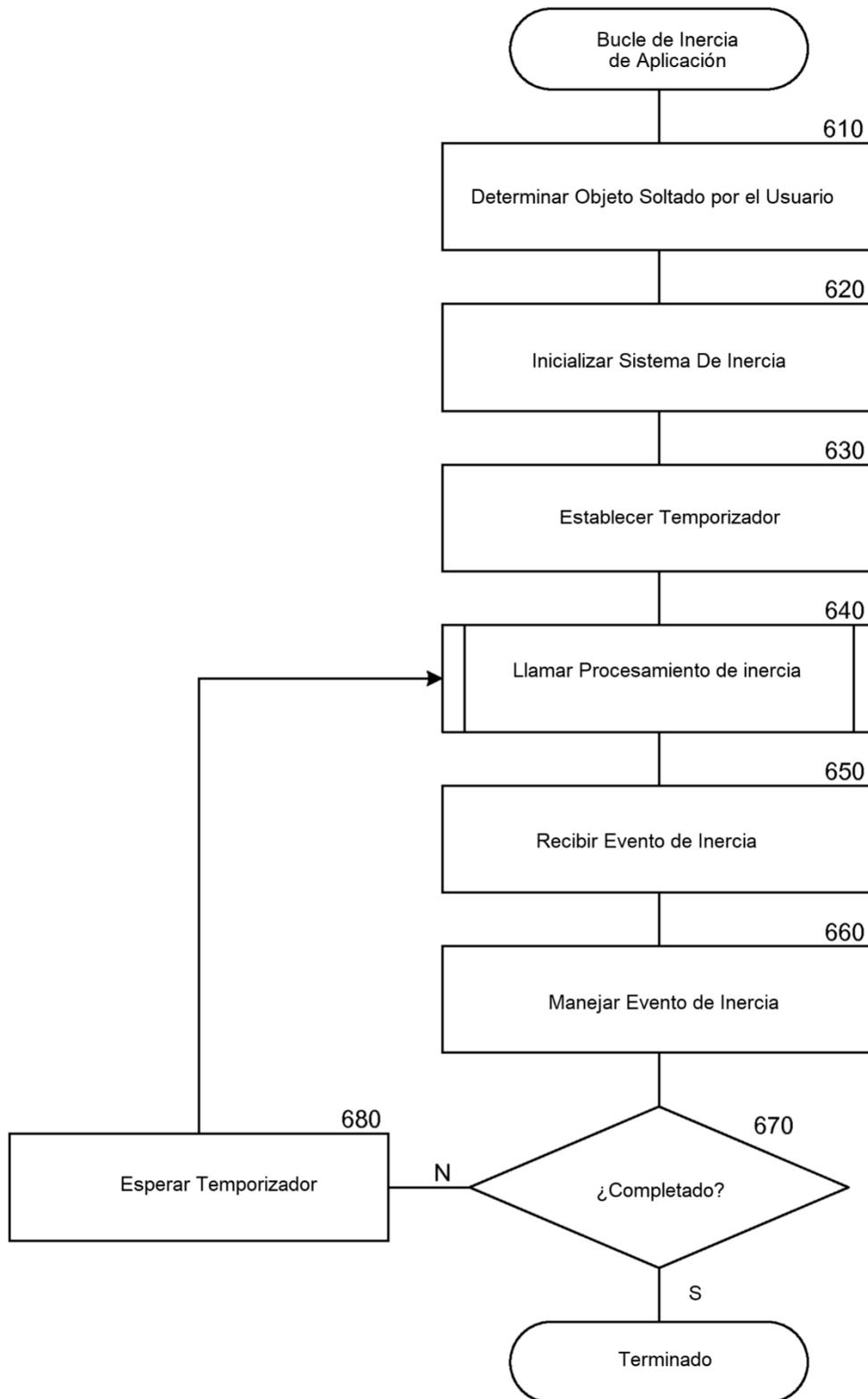
Figura 3



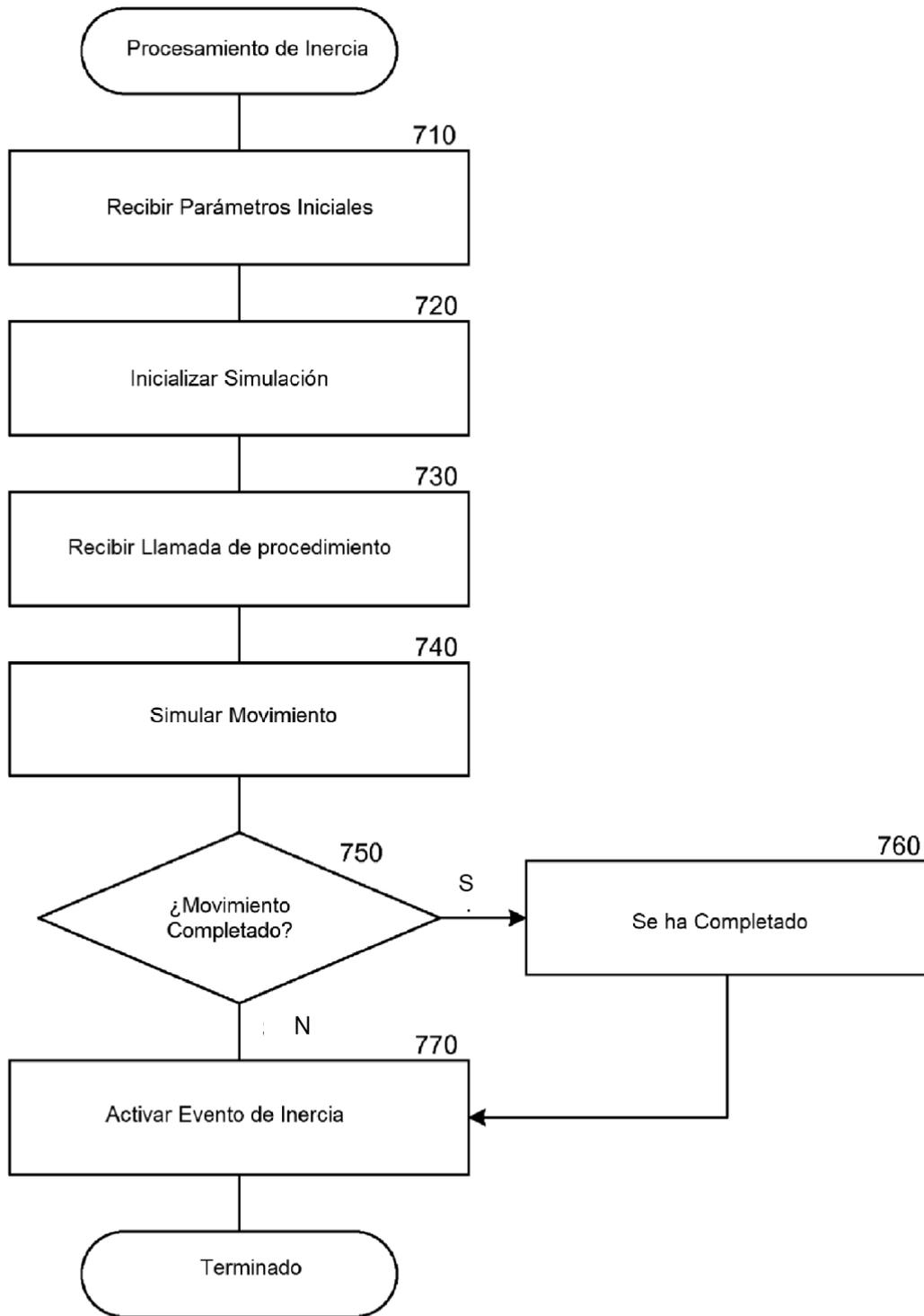
**Figura 4**



**Figura 5**



**Figura 6**



**Figura 7**