

ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(21)(22) Заявка: 2011116315/08, 16.10.2009

(24) Дата начала отсчета срока действия патента:
16.10.2009

Приоритет(ы):

(30) Конвенционный приоритет:
26.10.2008 US 12/258,439

(43) Дата публикации заявки: 27.10.2012 Бюл. № 30

(45) Опубликовано: 27.11.2014 Бюл. № 33

(56) Список документов, цитированных в отчете о
поиске: US 2007/0157089 A1, 05.07.2007. US
2008/0168403 A1, 10.07.2008. US 2007/0152984
A1, 05.07.2007. US 2007/0268269 A1, 22.11.2007.
RU 2008113045 A, 20.08.2008(85) Дата начала рассмотрения заявки РСТ на
национальной фазе: 25.04.2011(86) Заявка РСТ:
US 2009/060977 (16.10.2009)(87) Публикация заявки РСТ:
WO 2010/048041 (29.04.2010)Адрес для переписки:
129090, Москва, ул.Б.Спасская, 25, строение 3,
ООО "Юридическая фирма Городисский и
Партнеры", А.В.Миц

(72) Автор(ы):

ТАУНСЕНД Рид Л. (US),
ТУ Сяо (US),
СКОТТ Брайан Д. (US),
ТОРСЕТ Todd A. (US),
САЙКС Кеннет В. (US),
ПРАДХАН Самир С. (US),
ТИД Дженнифер А. (US)

(73) Патентообладатель(и):

МАЙКРОСОФТ КОРПОРЕЙШН (US)

C 2
6 6
8 8
4 4
3 3
2 2
R UR U
2 5 3 4 4 8 6 C 2

(54) МОДЕЛИРОВАНИЕ ИНЕРЦИИ ОБЪЕКТОВ ПРИ ВВОДЕ МНОЖЕСТВЕННЫМ КАСАНИЕМ

(57) Реферат:

Изобретение относится к области управления графическим приложением пользователя с использованием технологии множественного касания (multi-touch). Техническим результатом является обеспечение более реалистичного движения объектов приложений, управляемых с использованием аппаратного обеспечения multi-touch. Способ содержит этапы, на которых: определяют, что пользователь высвободил объект приложения, путем удаления одного или более контактов, связанных с вводом множественным касанием, с поверхности сенсорного ввода, при этом объект приложения отображается на

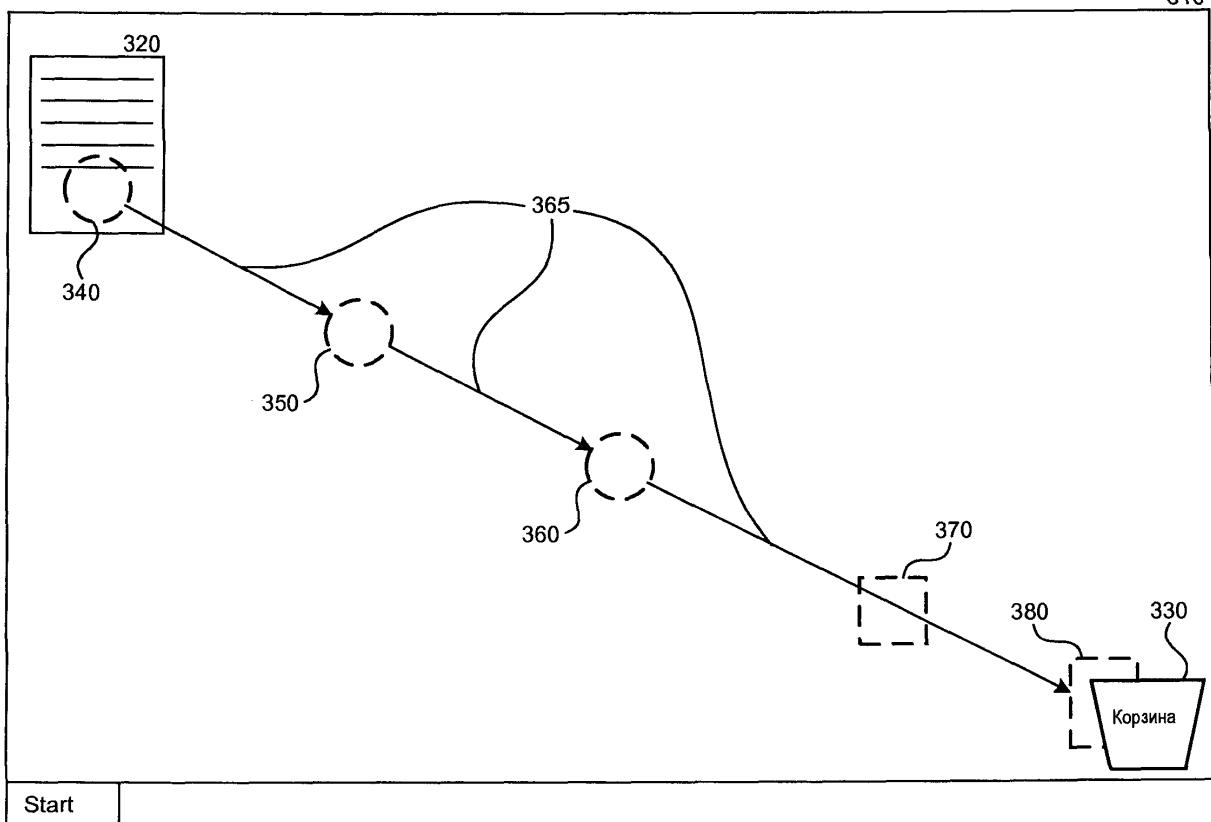
дисплее; после данного определения вызывают инерционный интерфейс прикладного программирования (API), исполняющийся на компьютерном устройстве, для обработки моделирования перемещения объекта приложения для текущего периода обработки инерции; принимают посредством приложения инерционное событие, сгенерированное инерционным API после упомянутого определения, причем инерционное событие описывает воздействие на объект на основе смоделированной инерции; и обрабатывают принимаемое инерционное событие на основе

R U 2 5 3 4 4 8 6 C 2

эффекта упомянутого воздействия в контексте приложения путем изменения объекта приложения и отображения объекта приложения

на дисплее соответственным образом. 3 н. и 17 з.п. ф-лы, 7 ил., 1 табл.

310



ФИГ. 3

R U 2 5 3 4 4 8 6 C 2

FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY

(12) ABSTRACT OF INVENTION

(21)(22) Application: 2011116315/08, 16.10.2009

(24) Effective date for property rights:
16.10.2009

Priority:

(30) Convention priority:
26.10.2008 US 12/258,439

(43) Application published: 27.10.2012 Bull. № 30

(45) Date of publication: 27.11.2014 Bull. № 33

(85) Commencement of national phase: 25.04.2011

(86) PCT application:
US 2009/060977 (16.10.2009)(87) PCT publication:
WO 2010/048041 (29.04.2010)Mail address:
129090, Moskva, ul.B.Spasskaja, 25, stroenie 3, OOO
"Juridicheskaja firma Gorodisskij i Partnery",
A.V.Mits(72) Inventor(s):
TAUNSEND Rid L. (US),
TU Sjao (US),
SKOTT Brajan D. (US),
TORSET Todd A. (US),
SAJKS Kennet V. (US),
PRADKhAN Samir S. (US),
TID Dzhennifer A. (US)(73) Proprietor(s):
MAJKROSOFT KORPOREJShN (US)C2
C6
C8
C4
C3
C4
C2
RUR U
2 5 3 4 4 8 6
C 2

(54) MULTI-TOUCH OBJECT INERTIA SIMULATION

(57) Abstract:

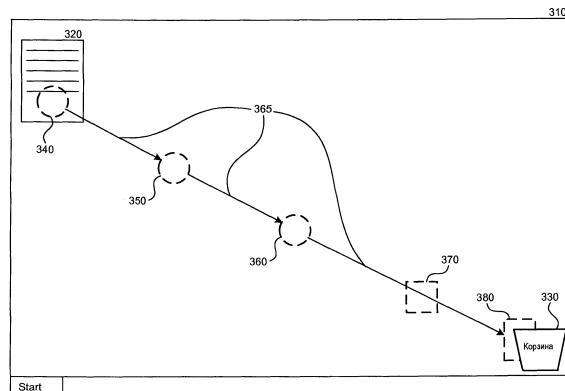
FIELD: physics, computer engineering.

SUBSTANCE: invention relates to controlling a graphical user application using multi-touch. The method comprises steps of: determining that a user has released an application object by deleting one or more contacts associated with multi-touch input from a touch input surface, wherein the application object is displayed on a display; invoking an inertia application programming interface (API) running on a computing device to process a simulation of movement of the application object for a current inertia processing period; receiving by the application an inertia event generated by the inertial API after said determination, wherein the inertia event describes a manipulation of the object based on simulated inertia; handling the received inertia event based on an effect of the manipulation in a context of the application by modifying the application object and displaying the application object on the display in

the appropriate manner.

EFFECT: providing more realistic movement of application objects controlled using multi-touch hardware.

20 cl, 7 dwg, 1 tbl



Фиг. 3

Уровень техники

Планшетный компьютер или компьютер со световым пером, является ноутбуком или мобильным компьютером в форме планшета, оснащенный сенсорным экраном или графической гибридной технологией планшет/экран, которая позволяет

- 5 пользователю работать на компьютере с помощью стилуса, цифрового пера или пальцев вместо клавиатуры или мыши. Планшетные компьютеры предлагают более естественные формы ввода, такие, как зарисовывание и запись от руки, которые являются гораздо более привычными формами ввода, чем клавиатура и мышь, особенно для людей, которые мало знакомы с компьютерами. Планшетные компьютеры также могут быть
- 10 более доступны, поскольку те люди, которые физически не в состоянии использовать клавиши, могут использовать дополнительные функции планшетного компьютера, чтобы иметь возможность взаимодействовать с электронным миром.

Технология multi-touch или multitouch (мультитач) обозначает множество способов взаимодействия, которые позволяют пользователям компьютеров управлять

- 15 графическими приложениями с использованием нескольких пальцев или устройств ввода (например, стилуса). Осуществления технологии multi-touch обычно включают сенсорное аппаратное обеспечение (например, экран, стол, стену и так далее) и программное обеспечение, которое распознает несколько одновременных точек касания. Технология multi-touch отличается от традиционных сенсорных экранов (например,
- 20 компьютерная сенсорная панель, АТМ, торговый киоск), которые могут распознавать только одну точку касания в один момент времени. Аппаратное обеспечение технологии multi-touch может определять касания с использованием тепла, давления пальцев, камер с высокой скоростью захвата, инфракрасного света, оптического захвата, настраиваемой электромагнитной индукции, ультразвуковых приемников, микрофонов
- 25 преобразователей, лазерных дальномеров, захвата тени и других механизмов. Существует много приложений для multi-touch интерфейсов, и разработчики и пользователи приложений предлагают еще больше. Некоторые использования являются индивидуалистическими (например, Microsoft Surface, Apple iPhone, HTC Diamond). Как новый способ ввода, способ multi-touch открывает возможности для новых парадигм
- 30 работы пользователей.

Приложение не может использовать аппаратное обеспечение multi-touch без интерфейса для программного обеспечения приложения для приема информации от аппаратного обеспечения multi-touch. К сожалению, каждое устройство с аппаратным обеспечением multi-touch включает в себя свой собственный интерфейс и авторы

- 35 приложения должны иметь определенные знания аппаратного обеспечения устройства для записи программного обеспечения, которое работает с устройством. Например, провайдер аппаратного обеспечения multi-touch может предоставить драйвер режима ядра (kernel-mode) и интерфейс приложения пользователяского режима (user-mode), через который приложения с программным обеспечением пользовательского режима
- 40 могут связываться с аппаратным обеспечением multi-touch для приема информации о касаниях. Авторы приложений записывают программное обеспечение, которое связывается с интерфейсом приложения пользователяского режима, но программное обеспечение автора приложения работает только с этим аппаратным обеспечением multi-touch. Пользователь компьютера с устройством с различным аппаратным
- 45 обеспечением multi-touch не может использовать программное обеспечение автора приложения, кроме случая, когда автор приложения производит различную версию программного обеспечения, которая работает корректно с устройством компьютера пользователя. Это обеспечивает очень ограниченный потенциал рынка для авторов

приложений, снижает стимулы для создания приложений, поддерживающих взаимодействия multi-touch, и сохраняет стоимость наиболее популярных устройств высокой, для которых наибольшее число приложений доступно.

Другой проблемой является трудность для приложений определить намерения

5 пользователя на основе сенсорного ввода, принимаемого от аппаратного обеспечения multi-touch. Сенсорный ввод может приниматься как список координат, где аппаратное обеспечение воспринимает сенсорный ввод в любой момент времени. Каждое приложение должно включать в себя программное обеспечение, чтобы интерпретировать координаты и определять намерение пользователя. Кроме того, намерение пользователя

10 может распространяться за пределы принятого фактического сенсорного ввода. Пользователь может ожидать поведения виртуальных объектов, как они ведут себя в физическом мире. Например, пользователь может ожидать, чтобы быть в состоянии "бросить" файл с одной стороны рабочего стола на другую, щелкая его/ее пальцем. Этот тип движения не поддерживается существующими приложениями multi-touch,

15 которые будут ожидать, чтобы пользователь перетащил его/ее пальцем с одной стороны экрана весь путь до другой. Даже если одно приложение обеспечивает поддержку этого типа движения, то другие приложения не могут этим воспользоваться, и таким образом авторам приложений придется повторить работу автора первого приложения, чтобы предложить ту же функциональность в своих приложениях.

20 **Сущность изобретения**

Инерционная система обеспечивает общую платформу и прикладной программный интерфейс (application-programming interface, API) для приложений для расширения ввода, принимаемого от различных устройств с аппаратным обеспечением multi-touch для моделирования реального поведения объектов. Манипуляции, принимаемые посредством 25 приложения, только описывают движения объекта на основе движения контактов с аппаратным обеспечением multi-touch. Однако чтобы двигаться естественно, объекты должны также проявлять физические характеристики, такие, как упругость и торможение. Когда пользователь поднимает все контакты от объекта, инерционная система обеспечивает дополнительные манипуляционные события с приложением так, 30 что приложение может обрабатывать события, как если бы пользователь еще двигал объект с помощью касания. Однако инерционная система фактически генерирует события на основе моделирования поведения объектов. Если пользователь перемещает объект в другой объект, инерционная система передает манипуляционные события на основе характеристик границ объектов. Таким образом, инерционная система 35 обеспечивает более реалистичное движение для объектов приложений, которыми манипулирует пользователь с использованием аппаратного обеспечения multi-touch, и API обеспечивает соответствующее ощущение для манипуляций в различных приложениях.

Эта Сущность изобретения представлена, чтобы представить набор концепций в 40 упрощенной форме, которые далее рассматриваются в Подробном описании. Эта Сущность изобретения не предназначена для идентификации ключевых функций или существенных функций заявленного предмета изобретения, она также не предназначена для ограничения области заявленного предмета изобретения.

Краткое описание чертежей

45 Фиг. 1 является блок-схемой, которая иллюстрирует компоненты инерционной системы, в одном варианте осуществления.

Фиг. 2 является схемой потока данных, которая иллюстрирует типичную операционную среду инерционной системы и поток данных между компонентами, в

одном варианте осуществления.

Фиг. 3 является схемой дисплея, которая иллюстрирует объект приложения, манипулируемый касанием пользователя, в одном варианте осуществления.

Фиг. 4 является блок-схемой последовательности операций, которая иллюстрирует обработку цикла ввода приложения multi-touch с использованием инерционной системы для обработки манипуляционных событий, в одном варианте осуществления.

Фиг. 5 является блок-схемой последовательности операций, которая иллюстрирует обработку инерционной системы, когда система принимает сенсорный ввод, в одном варианте осуществления.

Фиг. 6 является блок-схемой последовательности операций, которая иллюстрирует обработку приложения multi-touch с использованием инерционной системы для обработки инерционных событий, в одном варианте осуществления.

Фиг. 7 является блок-схемой последовательности операций, которая иллюстрирует обработку компоненты моделирования инерционной системы обработки, в одном варианте осуществления.

Подробное описание

Инерционная система обеспечивает общую платформу и API для приложений для расширения ввода, принимаемого от различных устройств с аппаратным обеспечением multi-touch, для моделирования реального поведения объектов. Например, реальные 20 объекты, как правило, не прекращают движение, когда пользователь прекращает толкать их, а проявляют некоторую инерцию и продолжают двигаться, пока трение не замедляет их до остановки. В некоторых вариантах осуществления, сенсорный ввод сначала проходит процесс интерпретации движения одного или более контактов как манипуляций. Манипуляции отображаются более явно в намерения пользователя, чем 25 отображаются индивидуальные сенсорные вводы, и добавляют поддержку для основной трансформации объектов с использованием нескольких сенсорных контактов.

Приложение может использовать манипуляции для поддержки вращения, изменения 30 размера и преобразование нескольких объектов (например, фотографий) в то же время. Манипуляции могут быть описаны как двухмерные (2D) аффинные преобразования, которые содержат информацию вращения, масштабирования (например, увеличения) и преобразования (например, панорамирования).

Каждое прикосновение аппаратного обеспечения multi-touch называется контактом. Например, когда пользователь устанавливает его/ее палец на аппаратное обеспечение multi-touch, перемещает его/ее палец вокруг, и поднимает его/ее палец, эта 35 последовательность событий является отдельным контактом. Например, если пользователь перемещает два контакта ближе друг к другу или дальше друг от друга, система может определить, что пользователь осуществляет масштабирование (например, масштабирования в или из) объекта. В качестве другого примера, если пользователь перемещает несколько контактов в круговом движении, тогда система может 40 интерпретировать движение как вращение объекта. Каждое приложение может определять объекты, которые относятся по-разному, так что приложение решает присоединить событие системы (называемой манипуляционным процессором) к каждому объекту, чтобы пользователь мог манипулировать с использованием сенсорного ввода в приложении. Например, приложения просмотра фотографий могут присоединять 45 манипуляционный процессор к каждой отображаемой фотографии, так что пользователь может перемещать фотографии вокруг, масштабировать фотографии, вращать фотографии и т.д.

Манипуляции, обрабатываемые посредством приложения, только описывают

движения объекта на основе движения контактов. Однако чтобы двигаться естественно, объекты должны также проявлять физические характеристики, такие, как упругость и замедление. Когда пользователь поднимает все контакты от объекта, инерционная система обеспечивает дополнительные манипуляционные события для приложения,

- 5 так что приложение может обрабатывать события, как если бы пользователь еще перемещал объект с помощью прикосновения. Однако инерционная система фактически генерирует события на основе моделирования поведения объектов. Например, если пользователь поднял контакты, пока у объекта была скорость в определенном направлении, тогда инерционная система продолжает отправку событий, которые
- 10 указывают, что объект перемещается в этом направлении, замедляя во времени, как объект замедляется. Если пользователь перемещает объект в другой объект, такой, как край экрана, инерционная система передает манипуляционные события на основе характеристик границ объектов. Например, если автор приложения определяет два объекта как упругие, тогда два объекта могут отскакивать друг от друга, когда
- 15 пользователь перемещает объекты друг в друга. Таким образом, инерционная система обеспечивает более реалистичное движение для объектов приложений, которыми манипулирует пользователь с использованием аппаратного обеспечения multi-touch, и API обеспечивает согласованное чувство для манипуляции в различных приложениях.

Фиг. 1 является блок-схемой, которая иллюстрирует компоненты инерционной 20 системы, в одном варианте осуществления. Инерционная система 100 включает интерфейс 110 аппаратного обеспечения, один или более манипуляционных процессоров 120, компоненту 130 преобразования ввода, компоненту 140 моделирования и интерфейс 150 приложения. Каждая из этих компонент описывается более подробно здесь.

Интерфейс 110 аппаратного обеспечения связывается с аппаратным обеспечением 25 для приема контактов прикосновений и перемещений. Интерфейс 110 аппаратного обеспечения может включать в себя несколько субкомпонент, которые могут работать вместе, чтобы обеспечить информацию сенсорного ввода. Например, операционная система может обеспечить общую модель управления для производителей аппаратного обеспечения multi-touch, чтобы обеспечить сенсорную информацию для их конкретного 30 аппаратурного обеспечения. Операционная система может преобразовывать сенсорную информацию, принимаемую через эту модель, в оконные сообщения (например, WM_TOUCH, рассматриваемые здесь) и передавать эти сообщения приложению. Таким образом, интерфейс 110 аппаратного обеспечения может включать в себя координацию аппаратного обеспечения, драйвер аппаратного обеспечения и слой операционной 35 системы. Результатом являются последовательности сообщений инерционной системе, которые идентифицируют конкретный контакт (например, прикосновение пальца) и координаты контакта с течением времени. Например, операционная система может обеспечить сообщение, когда новый контакт устанавливается на аппаратном обеспечении multi-touch, сообщение каждый раз, когда контакт перемещается, и 40 сообщение, когда контакт убирается от аппаратного обеспечения multi-touch.

Один или более манипуляционных процессоров 120 используют компоненту 130 преобразования ввода для интерпретации перемещения каждого контакта, связанного с конкретным объектом приложения. Манипуляционный процессор 120 может определить, что пользователь использует несколько контактов для выполнения 45 отдельного действия. Например, пользователь может прикоснуться к фотографии всеми пятью пальцами одной руки и повернуть его/ее руку, чтобы указать намерение вращать фотографию. Манипуляционный процессор 120 принимает пять отдельных контактов (один для каждого пальца) и изменения координат каждого контакта, как пользователь

вращает его/ее руку. Манипуляционный процессор 120 определяет, что каждый контакт захватывает тот же объект и выполняет то же вращение. Система информирует приложение, что пользователь повернул объект, но приложение может игнорировать, использовал ли пользователь два, пять или любое конкретное число пальцев или других контактов для выполнения вращения. Это значительно упрощает редактирование приложения, поскольку автор приложения может обрабатывать те типы манипуляций, которые имеют отношение к приложению и оставить инерционной системе интерпретацию значения каждого сенсорного ввода нижнего уровня, принимаемого от аппаратного обеспечения multi-touch.

Манипуляционный процессор 120 использует компоненту 130 преобразования ввода для вынесения решений о значении принимаемых перемещений различных контактов, как самостоятельно, так и согласованно. Например, если пользователь манипулирует фотографией двумя пальцами, которые создают два соответствующих контакта ввода, тогда манипуляционный процессор 120 использует компоненту 130 преобразования ввода для определения значения относительных перемещений между этими двумя контактами. Если два контакта двигаются врозь, тогда компонента 130 преобразования ввода может определить, что пользователь осуществляет масштабирование объекта для изменения размера объекта. Если два контакта вращаются, тогда компонента 130 преобразования ввода может определить, что пользователь вращает объект. Если два контакта оба перемещаются в конкретном направлении, тогда компонента 130 преобразования ввода может определить, что пользователь панорамирует объект в новое место. Хотя каждый вид перемещения обсуждается отдельно, отметим, что пользователь может осуществлять все три типа перемещений в то же время, и процессор преобразования ввода может сообщать общее преобразование приложению. Например, пользователь может вращать, масштабировать и панорамировать объект, все в одном движении.

Компонента 140 моделирования имитирует продолжение перемещения объекта приложения после того, как пользователь останавливает касание объекта, на основе параметров инициализации и ограничений, определенных для объекта. Приложение может инициализировать компоненту 140 моделирования с конечным состоянием манипуляционного процессора 120, связанного с объектом. Приложение может также определять различные характеристики объекта, такие, как поведение границ объекта. Компонента 140 моделирования использует технологии на основе физики для моделирования поведения объекта за интервал после того, как пользователь освобождает объект. Например, компонента 140 моделирования может продолжать запускать уведомления для приложения в той же форме, как манипуляционные события, принятые посредством приложения, пока пользователь двигал объект. Приложение может затем сконцентрироваться на реагировании на перемещения объекта вместо концентрации на действиях (пользователя или физических), вызвавших движение объекта.

Специалисты в данной области распознают много известных технологий для имитации эквивалентного физического поведения виртуальных объектов в программном обеспечении.

Интерфейс 150 приложения связывается с приложением для приема информации и обеспечения преобразования манипуляций для приложения. Интерфейс 150 приложения принимает информацию инициализации от приложения. Информация инициализации может определять, какие типы преобразований объект приложения поддерживает для конкретного объекта и связанного манипуляционного процессора, а также данные инициализации для компоненты 140 моделирования, когда пользователь больше не

двигает объект. Например, некоторые объекты приложения могут поддерживать масштабирование, но не вращение. Информация инициализации может также определять опорную точку объекта. Инерционная система обеспечивает преобразования манипуляций для приложения через интерфейс приложения. Например, когда

5 инерционная система принимает сенсорный ввод нижнего уровня, который система интерпретирует как распознаваемое преобразование (например, вращение), система запускает событие для уведомления приложения о манипуляции. Приложение обрабатывает преобразование манипуляции для изменения объекта на основе преобразования. Например, если пользователь повернул объект, тогда приложение

10 может хранить новое положение объекта, чтобы использовать в следующий раз, когда приложение отображает объект. В качестве другого примера, если объект продолжает вращаться, после того, как пользователь отпустил его, на основе вычислений компоненты 140 моделирования, тогда приложение может хранить новое положение объекта.

15 Вычислительное устройство, на котором осуществляется система, может включать в себя центральный процессор, память, устройства ввода (например, клавиатуру и указательные устройства), устройства вывода (например, дисплей) и устройства хранения (например, дисководы). Память и устройства хранения являются машиночитаемыми носителями, которые могут быть закодированы с помощью исполняемых компьютером

20 инструкций, которые осуществляет система, что означает машиночитаемый носитель, который содержит инструкции. Кроме того, структуры данных и структуры сообщений могут храниться или передаваться через среду передачи данных, такую, как сигнал на линии связи. Различные линии связи могут использоваться, такие, как сеть Internet, локальная сеть, глобальная сеть, подключение точка-точка удаленного доступа, сотовая

25 телефонная сеть и так далее.

Варианты осуществления системы могут осуществляться в различных операционных средах, которые включают в себя персональные компьютеры, серверы, портативные устройства или ноутбуки, многопроцессорные системы, системы на основе микропроцессоров, программируемую бытовую электронику, цифровые камеры, сетевые

30 персональные компьютеры, миникомпьютеры, универсальные вычислительные машины, распределенные вычислительные среды, которые включают в себя любые из упомянутых выше систем или устройств, и так далее. Компьютерные системы могут быть сотовыми телефонами, персональными цифровыми помощниками, смартфонами, персональными компьютерами, программируемой бытовой электроникой, цифровыми камерами и так

35 далее.

Система может быть описана в общем контексте исполняемых компьютером инструкций, таких, как программные модули, выполняемые одним или более компьютерами или другими устройствами. В целом, программные модули включают процедуры, программы, объекты, компоненты, структуры данных и так далее, которые

40 выполняют конкретные задачи или осуществляют конкретные абстрактные типы данных. Обычно, если требуется, то функциональность программных модулей может быть совмещена или распределена в различных вариантах осуществления.

Фиг. 2 является схемой потока данных, которая иллюстрирует типичную операционную среду инерционной системы и поток данных между компонентами, в

45 одном варианте осуществления. Аппаратное обеспечение устройства multi-touch порождает данные 210 ввода через интерфейс аппаратного обеспечения. Например, аппаратное обеспечение может передавать данные 210 ввода операционной системе через драйвер программного обеспечения, предоставляемый производителем

аппаратного обеспечения. Интерфейс аппаратного обеспечения обеспечивает входные события 220 для приложения 230. Например, приложение может информировать операционную систему, что приложение 230 поддерживает пользовательский ввод multi-touch и регистр для приема сообщений, относящихся к пользовательскому вводу multi-touch. Приложение 230 принимает информацию сенсорного ввода нижнего уровня как изменения 240 ввода и направляет изменения 240 ввода манипуляционной системе 250. Например, изменения 240 ввода могут описывать каждое перемещение одного или более контактов прикосновений с помощью аппаратного обеспечения с использованием набора координат, которые указывают текущую позицию каждого контакта и другие характеристики перемещения. Манипуляционная система 250 интерпретирует изменения 240 ввода и уведомляет приложение 230 об одном или более манипуляционных событиях 260, которые указывают манипуляции верхнего уровня, которые пользователь выполняет над отображаемым объектом. Например, если движение контактов указывает, что пользователь намерен вращать объект, то манипуляционные события 260 указывают угол вращения.

Когда пользователь осуществляет перемещение объекта (например, когда приложение принимает уведомление, что каждый контакт, касающийся объекта, был перемещен от сенсорного аппаратного обеспечения), приложение 230 передает информацию 270 инициализации инерционной системе 280. Инерционная система 280 определяет следующую позицию объекта и обеспечивает инерционные события 290, подобные манипуляционным событиям 260, которые манипуляционная система 250 предоставила, когда пользователь двигал объект. Приложение 230 также обеспечивает управляющий таймер, чтобы периодически вызывать инерционную систему 280, чтобы обеспечить следующую позицию объекта через инерционные события 290. Приложение 230 обрабатывает инерционные события способом подобно манипуляционным событиям.

Хотя схема иллюстрирует, что приложение сначала принимает сенсорный ввод и передает сенсорный ввод манипуляционной системе и инерционной системе, в некоторых вариантах осуществления, эти системы принимают сенсорный ввод напрямую от интерфейса аппаратного обеспечения, интерпретируют сенсорный ввод и предоставляют интерпретированные манипуляционные события приложению. Подобным образом, приложение может не знать, что отдельная инерционная система 280 предоставляет инерционные события после того, как пользователь прекращает перемещать объект с помощью прикосновения, но даже может принимать события от одного интерфейса в течение времени, когда пользователь перемещает объект и после того, как объект перемещается на основе инерции. Это представляет альтернативную архитектуру, которая обеспечивает подобную результирующую функциональность, но дает приложению меньше контроля над обработкой входных данных. Например, приложение может не быть в состоянии определить отдельные объекты приложения, к которым система присоединяет отдельные манипуляционные процессоры. RTS plug-in, рассматриваемый здесь, является примером этой альтернативной архитектуры для системы.

Фиг. 3 является схемой дисплея, которая иллюстрирует объект приложения, манипулируемый касанием пользователя, в одном варианте осуществления. Приложение может одновременно отображать и принимать сенсорный ввод для многих таких объектов. Например, командное приложение операционной системы может отображать один или более документальных объектов, хранящихся на рабочем столе компьютера пользователя. На дисплее 310, документальный объект 320 представляет документ на рабочем столе пользователя, который пользователь хочет перетащить в корзину 330

с использованием касания. Пользователь выполняет мерцающее движение документального объекта 320, которое приводит к обработке системой первой манипуляционной позиции 340, второй манипуляционной позиции 350 и третьей манипуляционной позиции 360. Приложение принимает первую манипуляционную позицию 340, когда пользователь первоначально касается документального объекта 320 одним или более пальцами (т.е. контактами). Приложение принимает вторую манипуляционную позицию 350, когда пользователь двигает его/ее пальцы через экран. Приложение принимает третью манипуляционную позицию, когда пользователь поднимает его/ее пальцы от экрана. Стрелки 365 представляют векторы перемещения документального объекта 320.

Без инерции, документальный объект 320 остановился бы на третьей манипуляционной позиции 360, которая вероятно не соответствует намерению пользователя. Инерционная система обеспечивает дополнительные манипуляционные позиции приложению, как если бы пользователь еще касался и перемещал документальный объект 320, на основе скорости документального объекта 320, когда пользователь отпускает документальный объект 320. Приложение принимает первую манипуляционную позицию 370 на основе инерции, когда приложение инициирует инерционную систему, и вызывает функцию обработки инерционной системы для первого момента времени. Приложение принимает вторую манипуляционную позицию 380 на основе инерции, по мере того, как приложение продолжает вызывать функцию обработки инерционной системы. Поскольку итоговая манипуляционная позиция 380 документального объекта 320 оканчивается на корзине 330, приложение обрабатывает контакт между этими двумя объектами (например, посредством размещения документального объекта 320 в корзину 330). В иллюстрированном примере, даже хотя инерционная система замедляет движение документального объекта 320, документальный объект 320 еще может перемещаться порядочное расстояние через дисплей 310 на основе высокой начальной скорости перемещения пользователя документального объекта 320 в начале.

Фиг. 4 является блок-схемой последовательности операций, которая иллюстрирует цикл обработки ввода приложения multi-touch с использованием инерционной системы для обработки манипуляционных событий, в одном варианте осуществления. В блоке 410, приложение принимает сенсорный ввод нижнего уровня. Например, операционная система или экземпляр инерционной системы принимает информацию сенсорного контакта от аппаратного обеспечения multi-touch и направляет информацию сенсорного контакта приложению. В блоке 420, приложение определяет объект, к которому применяется ввод. Например, приложение может проверить координаты принимаемого ввода посредством сравнения этих координат с координатами каждого объекта приложения, отображаемого приложением. Если сенсорный ввод находится в пределах границ отображаемого объекта приложения, тогда приложение определяет, что сенсорный ввод применяется к этому объекту. В блоке 430, приложение передает принимаемый сенсорный ввод и информацию об определенном объекте приложения манипуляционному API для вызова инерционной системы (смотрите Фиг. 5). Например, приложение может создать числовой идентификатор для каждого объекта приложения и передавать числовой идентификатор инерционной системе каждый раз, когда сенсорный ввод соответствует этому объекту.

В блоке 440, приложение принимает манипуляционное событие от инерционной системы, которое описывает одну или более манипуляций определенного объекта приложения. Например, приложение может принимать событие, описывающее 2D

аффинное преобразование объекта приложения. Необходимо отметить, что блок 440 иллюстрируется последовательно после блока 430 для простоты иллюстрации. На практике, приложение может принимать много событий сенсорного ввода, перед тем, как инерционная система уведомляет приложение манипуляционным событием. Не 5 обязательно отображать одно-к-одному события сенсорного ввода в манипуляционные события. Поскольку манипуляционные события представляют интерпретацию верхнего уровня сенсорного ввода нижнего уровня, множество данных сенсорного ввода может составить одно манипуляционное событие. В блоке 450, приложение обрабатывает 10 принимаемое манипуляционное событие. Например, если принимаемое манипуляционное событие является вращением, тогда приложение может вращать объект приложения на экране и сохранять новые позиции объектов приложения для использования, когда объект приложения отображается снова. Инерционная система освобождает приложение от выполнения этапов, специфичных для конкретного устройства с аппаратным 15 обеспечением multi-touch, или даже от сведений о том, какое устройство с аппаратным обеспечением предоставляет multi-touch ввод. Кроме того, инерционная система освобождает приложение от обработки перемещений отдельных контактов и позволяет приложению фокусироваться на обработке преобразований на уровне объектов приложения.

В блоке 460, приложение ждет следующего сенсорного ввода. Например, приложение 20 может вызвать предоставляемое операционной системой сообщение API, такое, как GetMessage в Microsoft Windows, которое ждет доставки следующего сообщения в очередь сообщений приложения. В блоке 470 принятия решений, если приложение принимает следующий сенсорный ввод, тогда приложение возвращается к блоку 410 для обработки 25 ввода, иначе приложение возвращается к блоку 460 для продолжения ожидания следующего ввода. Когда приложение закрывается, приложение выходит из цикла ввода (не показано).

Фиг. 5 является блок-схемой последовательности операций, которая иллюстрирует обработку инерционной системы, когда система принимает сенсорный ввод, в одном варианте осуществления. В блоке 505, система принимает сенсорный ввод наряду с 30 информацией, определяющей объект приложения, с которым связан сенсорный ввод. Например, сенсорный ввод может включать координаты или информацию о других позициях одного или более сенсорных контактов, и информация об объекте приложения может включать в себя идентификатор, который приложение назначило конкретному отображаемому объекту, которому соответствует сенсорный ввод аппаратного 35 обеспечения multi-touch. В блоке 510, система определяет манипуляционный процессор, связанный с объектом приложения. В блоке 520 принятия решений, если система не имеет процессор управления, заранее связанный с объектом приложения, тогда система переходит к блоку 530, иначе система переходит к блоку 540. В блоке 530, система 40 создает манипуляционный процессор и связывает его с объектом приложения, затем переходит к блоку 540.

В блоке 540 принятия решений, если принимаемый сенсорный ввод указывает, что приложение приняло новый контакт (например, событие прикосновения), тогда система переходит к блоку 550, иначе система переходит к блоку 560. Например, пользователь может сделать первый контакт пальцем на объекте на экране или установить другой 45 палец (т.е. контакт) на объект, к которому он ранее прикоснулся. В блоке 550, система добавляет новый контакт в список контактов, связанных с манипуляционным процессором, и затем переходит к блоку 560. В блоке 560 принятия решений, если принимаемый сенсорный ввод указывает, что принимаемое приложением уведомление

о том, что сенсорный контакт был удален (например, событие, когда прикосновение убирается), тогда система переходит к блоку 570, иначе система переходит к блоку 580. Например, пользователь может поднять один или более пальцев от объекта, к которому он ранее прикоснулся. В блоке 570, система удаляет контакт из списка контактов,

5 связанных с манипуляционным процессором, и затем переходит к блоку 580. В блоке 580, система обрабатывает сенсорный ввод, чтобы определить какие-либо манипуляционные события, представленные сенсорным вводом. Например, перемещение прикосновения может указывать вращение или манипуляцию преобразования, в то время как удаление сенсорного контакта может указывать на завершение манипуляции.

10 В блоке 590, система запускает манипуляционное событие, чтобы передать информацию о преобразовании, описывающую манипуляцию приложению. Например, система может предоставить степень углового вращения объекта для приложения. После блока 590, эти этапы завершаются.

Фиг. 6 является блок-схемой последовательности операций, которая иллюстрирует 15 обработку приложения multi-touch с использованием инерционной системы для обработки инерционных событий, в одном варианте осуществления. В блоке 610, приложение определяет, что пользователь освободил объект. Например, после обработки манипуляционного события в блоке 450 Фиг. 4, приложение может принимать индикацию того, что манипуляция завершилась или что пользователь поднял все

20 контакты, которые касались объекта приложения. В блоке 620, приложение инициирует инерционную систему. Например, приложение может передавать ссылку манипуляционному процессору об обработке движения объекта и другую информацию инициализации. В блоке 630, приложение устанавливает таймер, который будет запускать интервал обработки инерции инерционной системы. Например, приложение может

25 установить таймер, который будет срабатывать каждые 100 миллисекунд для обработки следующего приращения перемещения объекта. В блоке 640, приложение вызывает функцию обработки инерционной системы (смотрите Фиг. 7). Например, инерционная система может обеспечить функцию "Process (Обработка)", которую приложение вызывает, чтобы информировать инерционную систему, что настало время для

30 выполнения моделирования за интервал с последнего интервала моделирования.

В блоке 650, приложение принимает одно или более инерционные события, которые описывают манипуляции объекта (например, вращение, преобразование, и/или масштабирование) на основе моделированной инерции. Например, если объект

35 перемещался в конкретном направлении, приложение может принимать инерционное событие, которое описывает манипуляцию преобразования в этом направлении. В качестве другого примера, если объект расширялся, когда пользователь отпустил его, приложение может принимать инерционное событие, которое описывает манипуляцию масштабирования. Необходимо отметить, что блок 650 иллюстрируется последовательно

40 после блока 640 для простоты иллюстрации. На практике, приложение может вызвать функцию обработки инерции несколько раз перед тем, как инерционная система уведомляет приложение инерционным событием. Не обязательно отображать один-к-одному вызовы функции обработки и инерционные события. С другой стороны, инерционная система может уведомить приложение о множестве инерционных событий после одного вызова функции обработки.

45 В блоке 660, приложение обрабатывает принимаемое инерционное событие на основе значения (например, эффекта) манипуляции в контексте конкретного приложения. Например, если принимаемое инерционное событие является вращением, тогда приложение может вращать объект приложения на экране и сохранять новые позиции

объектов приложения для использования, когда приложение отображает объект приложения снова. В блоке 670 принятия решений, если инерционные события завершены, тогда эти этапы завершаются, иначе система переходит к блоку 680. Инерционная система может информировать приложение, что конкретное

5 моделированная манипуляция завершилась, как возвращаемое значение от функции обработки или через уведомления, предоставляемые приложению (например, через интерфейс событий объектной модели программных компонентов (Component Object Model, COM)). В блоке 680, приложение ждет следующего запуска таймера, затем возвращается к блоку 640 для вызова функции обработки инерционной системы.

10 Фиг. 7 является блок-схемой последовательности операций, которая иллюстрирует обработку компоненты моделирования инерционной системы обработки, в одном варианте осуществления. В блоке 710, компонента принимает начальные параметры моделирования. Например, приложение или манипуляционный процессор может обеспечить итоговое состояние объекта приложения, когда пользователь остановил 15 касание объекта. В блоке 720, компонента инициирует моделирующую машину, которая выполняет вычисления на основе физики, чтобы определить поведение объекта на основе параметров. Например, моделирующая машина может обеспечить реалистичное замедление или упругое поведение для объектов приложений, приведенных в движение посредством сенсорного ввода пользователя. В блоке 730, компонента принимает вызов 20 обработки от приложения. Приложение или другая компонента запускает процесс моделирования путем многократного вызова функции обработки в регулярные интервалы, чтобы двигать моделирование вперед. Компонента моделирования может также внутренне генерировать таймер.

В блоке 740, компонента имитирует перемещения объекта на основе начальных 25 параметров, какой-либо предыдущей обработки и времени, которое прошло с момента последнего вызова обработки. Вызов обработки может также предоставлять метку, которая указывает время, в течение которого приложение желает использовать моделирование. Это позволяет приложению моделировать поведение приложения в других режимах, чем в режиме реального времени (например, для тестирования, 30 приложения или отладки). В блоке 750 принятия решений, если движение завершилось, тогда компонента переходит к блоку 760, иначе компонента переходит к блоку 770. Компонента может определить, что движение завершилось на основе таких факторов, как движется ли еще объект или перемещение объекта снизилось ниже определенного порога. В блоке 760, компонента устанавливает флаг завершения на следующем 35 инерционном событии. В блоке 770, компонента запускает инерционное событие для передачи информации преобразования, описывающей текущее перемещение (например, как манипуляция), приложению. Например, система может обеспечить степень углового вращения объекта для приложения. После блока 770, эти этапы завершаются.

В некоторых вариантах осуществления, инерционная система принимает ограничения 40 объекта от приложения. Например, приложение может определять упругость объекта, коэффициент трения (чтобы определить, как объект замедляется), граничные характеристики объекта и т.д. Например, автор приложения может определить жесткие объекты, которые пользователь может перемещать, и пружинистый край окна приложения, так что объекты, перемещаемые на край окна, пружинят от края окна, 45 когда пользователь отпускает их.

В некоторых вариантах осуществления, инерционная система принимает информацию о начальном состоянии объекта от манипуляционной системы, которая отслеживала движение объекта, когда пользователь манипулирует объектом с помощью сенсорного

ввода. Например, манипуляционная система может отслеживать текущую позицию каждого объекта, историю перемещения объекта, линейную и угловую скорость объекта и т.д. Автор приложения может предоставлять выходные данные манипуляции инерционной системе для инициализации инерционной системы, так что инерционная 5 система может плавно продолжить последнее перемещение объекта и замедлить его на основе соответствующей физики и характеристик объекта.

В некоторых вариантах осуществления, инерционная система принимает ограничения на движение объектов от приложения. Например, автор приложения может определять верхнюю границу расстояния, на которое объект может перемещаться, как только

10 пользователь отпустит объект. В качестве другого примера, приложение может определять верхнюю границу длительности перемещения объекта, как только пользователь отпустит объект. Эти и другие ограничения позволяют автору приложения регулировать инерционную систему с учетом типов объектов, манипулируемых посредством приложения, и расширять опыт работы пользователя с приложением.

15 В некоторых вариантах осуществления, инерционная система не предоставляет дополнительное перемещение для объектов с перемещением ниже заранее заданного порога. Порог может конфигурироваться посредством приложения. Например, инерционная система может иметь линейную или угловую скорость для конкретного объекта, ниже которой система не будет продолжать перемещение объекта после того, 20 как пользователь отпустит объект. Если объект не перемещается очень быстро, когда пользователь отпускает его, то пользователь может ожидать, что объект будет оставаться на месте и не продолжит перемещаться. Порог позволяет приложению или автору инерционной системы определить уровень перемещения после манипуляции, что обеспечивает хороший опыт работы пользователю.

25 В некоторых вариантах осуществления, инерционная система принимает инструкции для моделирования перемещения постепенно от приложения. Например, инерционная система может обеспечить функцию "Process (Обработка)" или "DoWork (Выполнение работы)", которую приложение вызывает, чтобы поручить инерционной системе выполнение части всего моделирования. Инерционная система может ожидать, когда 30 приложение установит таймер или иначе периодически вызывать функцию, чтобы вызывать инерционную систему для моделирования перемещения во времени в соответствии с обычным временем. Приложение может повлиять на характеристики манипуляционных событий, предоставляемых инерционной системой, путем изменения того, как часто приложение вызывает функцию. В других вариантах осуществления, 35 инерционная система использует внутренний таймер, чтобы обеспечить события управления на регулярной основе пока каждый объект не остановит перемещение (например, из-за замедления или других моделируемых сил).

В некоторых вариантах осуществления, инерционная система является частью операционной системы на основе сообщений, и система принимает сообщения, связанные 40 с сенсорным вводом, который операционная система принимает от аппаратного обеспечения. Например, с использованием парадигмы, похожей на WM_MOUSEMOVE для сообщений мыши, будущие версии Microsoft Windows могут предоставлять сообщение WM_TOUCH, которое содержит информацию сенсорного перемещения нижнего уровня, принимаемую от аппаратного обеспечения multi-touch. Операционная система может 45 также предоставлять мелкоструктурные сообщения, такие, как WM_TOUCHDOWN (когда новый контакт осуществляется с помощью аппаратного обеспечения multi-touch), WM_TOUCHMOVE (когда существующий контакт перемещается) и WM_TOUCHUP (когда контакт поднимается от аппаратного обеспечения multi-touch). Приложение,

которое принимает сообщение, связанное с WM_TOUCH, может ссылаться на инерционную систему и передавать сообщение инерционной системе для интерпретации и обработки. Приложение затем принимает события верхнего уровня, которые представляют интерпретацию манипуляций инерционной системы, направляемых 5 пользователем, на основе принимаемой информации сенсорного перемещения нижнего уровня.

В некоторых вариантах осуществления, инерционная система принимает информацию сенсорного перемещения нижнего уровня от специализированного аппаратного обеспечения, такого, как стилус в режиме реального времени. Например, средство

10 разработки программного обеспечения (Software Development Kit, SDK) для планшетного компьютера Microsoft обеспечивает компоненту стилуса в реальном времени (real-time stylus, RTS), которую авторы приложений могут расширять добавочными блоками (hooks). Добавочные блоки RTS принимают входные данные от RTS аппаратного обеспечения и могут выполнять обработку принимаемых входных данных. Инерционная 15 система может обеспечить добавочный блок, который приложение может добавлять в компоненту RTS для автоматической обработки RTS и других входных данных для манипуляции объектами приложений как рассматривается здесь. Добавочный блок RTS обеспечивает другой способ для инерционной системы для приема входных данных, но инерционная система интерпретирует входные данные и запускает события для 20 приложения, описывая манипуляции, подразумеваемые входными данными, как описано ранее. Пользователь может использовать сочетание стилуса и сенсорного ввода. Например, пользователь может рисовать объект стилусом и затем вращать объект с использованием его/ее пальцев.

В некоторых вариантах осуществления, инерционная система является частью общего 25 контроля, на который приложение может ссылаться, чтобы обеспечить общий пользовательский интерфейс. Microsoft Windows обеспечивает общие элементы управления для отображения списков, деревьев, кнопок и т.д. Подобным образом, инерционная система может обеспечить управление на основе multi-touch для манипуляции объектами приложений способами, рассматриваемыми здесь. Например, 30 система может обеспечить распределенное управление, которое позволяет пользователю отображать один или более объектов и манипулировать объектами. Распределенное управление обеспечивает обработку сенсорного ввода нижнего уровня и связывает входные данные с конкретным объектом приложения, и приложение принимает события от управления для обработки манипуляции объектов приложений. Например, если 35 управление указывает, что пользователь изменил размеры объекта, тогда приложение может хранить объекты нового размера.

В некоторых вариантах осуществления, инерционная система выполняет 40 рассматриваемую здесь обработку в трех измерениях. Хотя двумерное аппаратное обеспечение multi-touch рассматривается здесь, специалисты в данной области распознают, что обработку системы, рассматриваемой здесь, можно также эквивалентно применять к трехмерным (3D) манипуляциям, если доступно аппаратное обеспечение, чтобы обеспечить координатное перемещение в трех измерениях. Например, аппаратное обеспечение, которое определяет давление или использует камеры, чтобы определить 45 3D перемещения пальцев пользователя, может предоставлять координаты перемещения в третьем измерении инерционной системе, и инерционная система может тогда производить 3D преобразования, которые описывают манипуляции (например, вращение, масштабирование и преобразование) объектов во многих 3D направлениях.

Следующая таблица определяет один API, который инерционная система обеспечивает

для приложений для предоставления перемещения на основе инерции для объектов приложений, следя перемещению объектов на основе касаний пользователя.

	Свойства:	
5	BoundaryBottom (Граница снизу)	Ограничивает, насколько далеко в направлении нижней части экрана может двигаться целевой объект.
	BoundaryLeft (Граница слева)	Ограничивает, насколько далеко в направлении влево экрана может двигаться целевой объект.
	BoundaryRight (Граница справа)	Ограничивает, насколько далеко в направлении вправо экрана может двигаться целевой объект.
10	BoundaryTop (Граница сверху)	Ограничивает, насколько далеко в направлении верхней части экрана может двигаться целевой объект.
	DesiredAngularDeceleration (Требуемое угловое замедление)	Определяет требуемую скорость, при которой целевой объект остановится вращаться в радианах на миллисекунду.
	DesiredDeceleration (Требуемое замедление)	Определяет требуемую скорость, при которой операции преобразования будут замедляться.
15	DesiredDisplacement (Требуемое смещение)	Определяет требуемое расстояние, на которое будет перемещаться объект.
	DesiredExpansion (Требуемое расширение)	Определяет требуемое изменение среднего радиуса объекта.
	DesiredExpansionDeceleration (Требуемое расширение)	Определяет требуемое изменение среднего радиуса объекта.
	ElasticMarginBottom (Эластичный край снизу)	Определяет область снизу для подпрыгивания целевого объекта.
20	ElasticMarginLeft (Эластичный край слева)	Определяет крайнюю слева область для подпрыгивания целевого объекта.
	ElasticMarginRight (Эластичный край справа)	Определяет крайнюю справа область для подпрыгивания целевого объекта.
	InitialAngularVelocity (Начальная угловая скорость)	Определяет вращение цели, когда начинается движение.
25	InitialOriginX (Начало по горизонтали)	Получает или ставит свойство, указывающее горизонтальную позицию для целевого объекта. Это свойство определяет начальную горизонтальную позицию для цели с инерцией.
	InitialOriginY (Начало по вертикали)	Получает или ставит свойство, указывающее вертикальную позицию для целевого объекта. Это свойство определяет начальную вертикальную позицию для цели с инерцией.
30	InitialRadius (Начальный радиус)	Определяет расстояние от края цели до ее центра перед тем, как объект был изменен.
	InitialTimestamp (Начальная временная метка)	Определяет начальную временную метку для целевого объекта с инерцией.
	InitialVelocityX (Начальная скорость по горизонтали)	Определяет начальное движение целевого объекта по горизонтальной оси.
	InitialVelocityY (Начальная скорость по вертикали)	Определяет начальное движение целевого объекта по вертикальной оси.
35	Способы:	
	HRESULT Reset();	Инициализирует процессор с начальной временной меткой.
40	HRESULT Process([out] BOOL* completed);	Выполняет вычисления для данной отметки и может породить событие Delta (Дельта) или Completed (Завершение) в зависимости от того, завершена ли экстраполяция или нет. Если экстраполяция завершена на предыдущей отметке, то способ не работает (noop).
	HRESULT ProcessTime([in] DWORD timestamp, [out] BOOL* completed);	Выполняет вычисления для данной отметки и может породить событие Delta (Дельта) или Completed (Завершение) в зависимости от того, завершена ли экстраполяция или нет. Если экстраполяция завершена на предыдущей отметке, то способ не работает (noop).
45	HRESULT Complete();	Порождает событие Completed (Завершение).
	HRESULT CompleteTime([in] DWORD timestamp);	Обрабатывает данную отметку и порождает событие Completed (Завершение).

	События:	
	HRESULT ManipulationStarted([in] FLOAT x, [in] FLOAT y);	Обработка события, когда управление началось.
5	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,);	Обработка событий, которые происходят, когда управляемый объект изменяется.
10	HRESULT ManipulationCompleted([in] FLOAT x, [in] FLOAT y, [in] FLOAT cumulativeTranslationX, [in] FLOAT cumulativeTranslationY, [in] FLOAT cumulativeScale, [in] FLOAT cumulativeExpansion, [in] FLOAT cumulativeRotation,);	Обработка события, когда управление завершилось.
15		

В таблице выше, инерционная система может обеспечить перечисленные события на том же интерфейсе, на котором приложение ранее принимало события на основе 20 пользовательского перемещения.

Из вышеупомянутого будет очевидно, что конкретные варианты осуществления инерционной системы рассматривались здесь с целями иллюстрации, но что различные модификации могут быть сделаны без отклонения от идеи и области действия 25 изобретения. Например, хотя система рассматривалась в контексте манипуляций multi-touch, система обеспечивает моделирование инерции, которая может использоваться в других контекстах, таких, как игры и другие области, где обычно используется моделирование. Соответственно, изобретение не ограничивается кроме как прилагаемой формулой изобретения.

30 Формула изобретения

1. Способ, осуществляемый на компьютерном устройстве, содержащем поверхность сенсорного ввода и дисплей, при этом способ предназначен для обеспечения реалистичного перемещения на дисплее объектов, воздействие на которые выполняется с использованием ввода множественным касанием через поверхность сенсорного ввода, 35 причем способ содержит этапы, на которых:

40 определяют, что пользователь высвободил объект приложения, соответствующий приложению, исполняющемуся на компьютерном устройстве, путем удаления одного или более контактов, связанных с вводом множественным касанием, с поверхности сенсорного ввода, при этом объект приложения отображается на дисплее;

45 после данного определения вызывают инерционный интерфейс прикладного программирования (API), исполняющийся на компьютерном устройстве, для обработки моделирования перемещения объекта приложения для текущего периода обработки инерции, причем инерционный API обеспечивает независимую от приложений платформу, активируемую различными приложениями на компьютерном устройстве, при этом инерционным API моделируются инерционные перемещения объектов, независимо от типа объекта приложения;

принимают посредством приложения инерционное событие, сгенерированное инерционным API после упомянутого определения, причем инерционное событие

описывает воздействие на объект на основе смоделированной инерции; и

обрабатывают принимаемое инерционное событие на основе эффекта упомянутого воздействия в контексте приложения путем изменения объекта приложения и отображения объекта приложения на дисплее соответственным образом.

5. 2. Способ по п.1, дополнительно содержащий, после определения того, что пользователь высвободил объект, этап, на котором инициализируют инерционный API путем передачи ссылки на процессор воздействия, который обрабатывал перемещение данного объекта перед тем, как пользователь высвободил объект.

10. 3. Способ по п.1, дополнительно содержащий, после определения того, что пользователь высвободил объект, этап, на котором инициализируют инерционный API путем передачи одного или более параметров, описывающих состояние объекта, когда пользователь высвободил объект.

4. Способ по п.1, в котором при приеме инерционного события принимают информацию, описывающую двумерное аффинное преобразование объекта приложения.

15. 5. Способ по п.1, в котором воздействие, описываемое инерционным событием, включает в себя по меньшей мере одно из воздействия для поворота, воздействия для переноса и воздействия для масштабирования.

6. Способ по п.1, в котором принимаемое инерционное событие является поворотом, при этом обработка принимаемого инерционного события содержит поворот объекта 20 приложения на дисплее.

7. Способ по п.1, дополнительно содержащий этап, на котором устанавливают таймер, который определяет длительность периода обработки инерции, при этом приложение вызывает инерционный API при каждом запуске таймера.

25. 8. Способ по п.7, дополнительно содержащий этап, на котором принимают от инерционного API указание о том, что моделирование завершено, и таймер истекает, когда это указание принимается.

9. Способ по п.1, в котором при приеме инерционного события принимают уведомление через интерфейс событий СОМ (модели компонентных объектов).

10. 10. Машиночитаемый носитель информации, на котором закодированы инструкции 30 для управления компьютерным устройством, содержащим дисплей и поверхность сенсорного ввода, для моделирования перемещения на дисплее объекта приложения, ранее перемещаемого посредством ввода множественным касанием через поверхность сенсорного ввода, причем объект приложения является частью приложения, исполняющегося на компьютерном устройстве, посредством способа, содержащего 35 этапы, на которых:

принимают один или более начальных параметров моделирования, которые предоставляют состояние объекта приложения, соответствующее высвобождению пользователем объекта приложения путем прекращения контакта с поверхностью сенсорного ввода;

40. инициализируют машину моделирования, которая выполняет вычисления для моделирования инерции объекта приложения на основе начальных параметров моделирования;

принимают указание текущего периода моделирования;

45. моделируют посредством машины моделирования инерционное перемещение объекта приложения на основе упомянутых начальных параметров, какого-либо предыдущего моделирования инерции, выполненного машиной моделирования, и времени, которое прошло с какого-либо предыдущего периода моделирования; и

выдают инерционное событие для передачи информации преобразования,

описывающей текущее перемещение объекта приложения, в приложение, при этом объект приложения отображается на дисплее в соответствии с информацией преобразования.

11. Машиночитаемый носитель по п.10, при этом машина моделирования моделирует

5 реалистичный процесс замедления для объекта приложения после приведения его в движение пользователем посредством касания этого объекта.

12. Машиночитаемый носитель по п.10, при этом машина моделирования имитирует реалистичное упругое поведение для объекта приложения, когда объект приложения перекрывается с другим элементом на дисплее.

10 13. Машиночитаемый носитель по п.10, в котором при приеме указания текущего периода моделирования принимают от приложения вызов функции обработки моделирования.

14. Машиночитаемый носитель по п.10, в котором способ дополнительно содержит 15 этап, на котором определяют, завершилось ли моделирование, и, если моделирование завершилось, информируют приложение о том, что моделирование завершилось.

15. Компьютерное устройство для обработки сенсорного ввода с аппаратного обеспечения мультисенсорного ввода, содержащее:

модуль, который принимает низкоуровневые вводы для различных приложений от мультисенсорного устройства ввода, переводит низкоуровневые вводы в воздействия, 20 принимаемые этими различными приложениями, при этом, по меньшей мере, некоторые из воздействий основываются на соответственных совокупностях из низкоуровневых вводов, причем воздействия содержат двумерные аффинные преобразования;

приложение, содержащее графический объект, отображаемый на дисплее, причем с графическим объектом взаимодействует пользователь путем контакта с 25 мультисенсорным устройством ввода, так что соответствующие низкоуровневые вводы, направленные на графический объект, принимаются упомянутым модулем и приложение принимает соответствующие воздействия, генерируемые данным модулем, которые используются приложением для отображения графического объекта, при этом, когда пользователь прекращает контакт с мультисенсорным устройством ввода, упомянутый

30 модуль продолжает генерировать воздействия для графического объекта путем моделирования замедляющегося инерционного перемещения данного объекта.

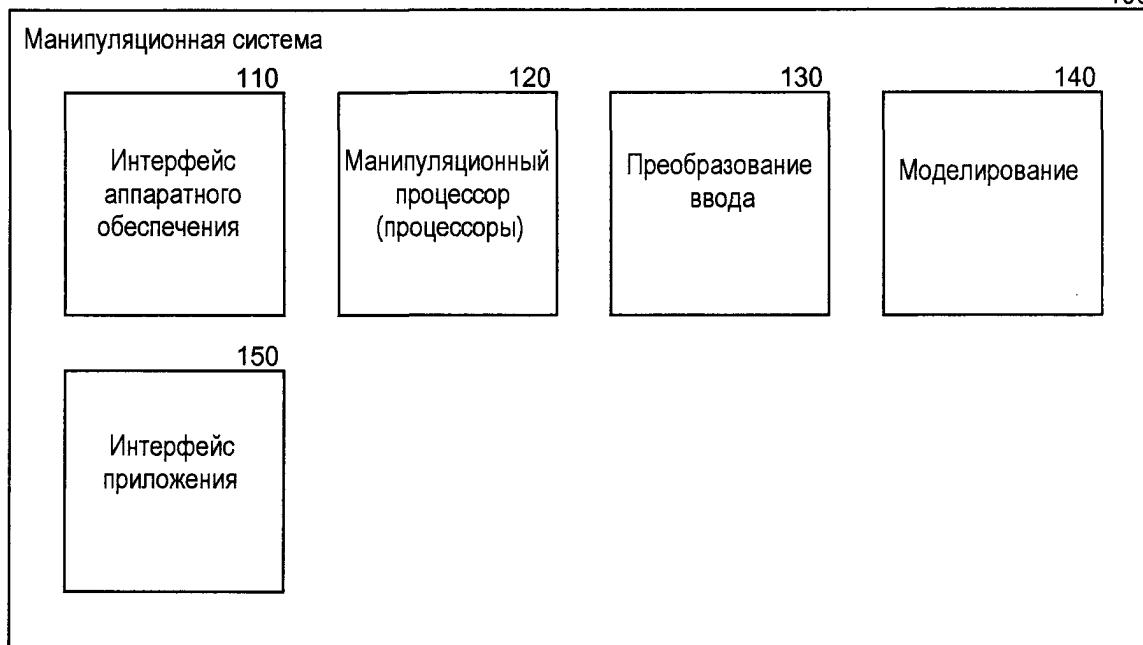
16. Компьютерное устройство по п.15, в котором упомянутый модуль моделирует замедляющееся инерционное перемещение в соответствии с состоянием объекта перед тем, как пользователь прекратил контакт с мультисенсорным устройством ввода.

35 17. Компьютерное устройство по п.16, в котором упомянутое состояние содержит скорость и направление объекта приложения.

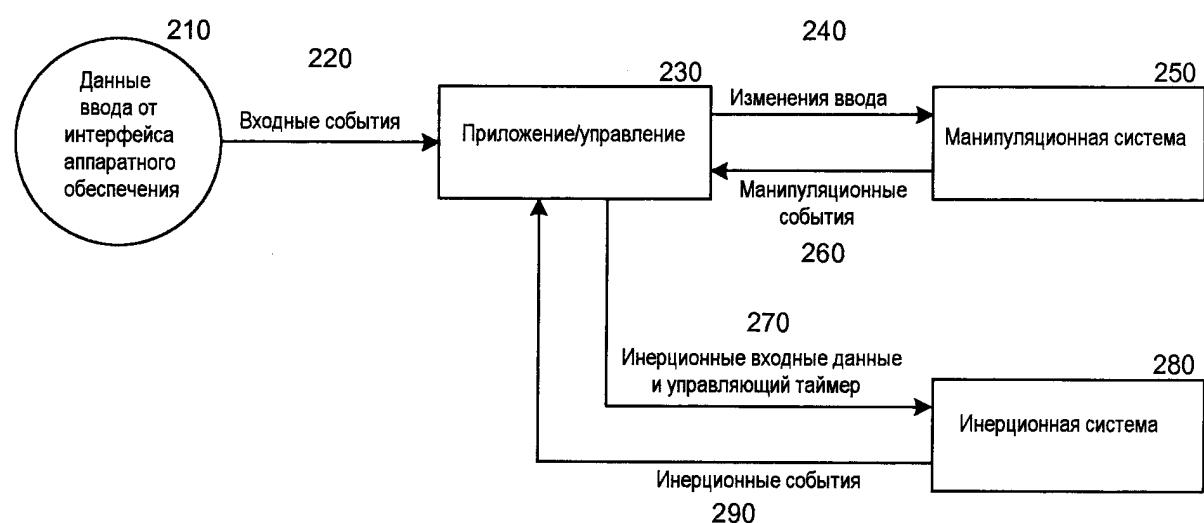
18. Компьютерное устройство по п.15, в котором низкоуровневые вводы содержат позиции, в которых пользователь осуществляет контакт с мультисенсорным устройством ввода.

40 19. Компьютерное устройство по п.15, в котором упомянутый модуль обеспечивает прием приложением уведомления о том, что графический объект остановился.

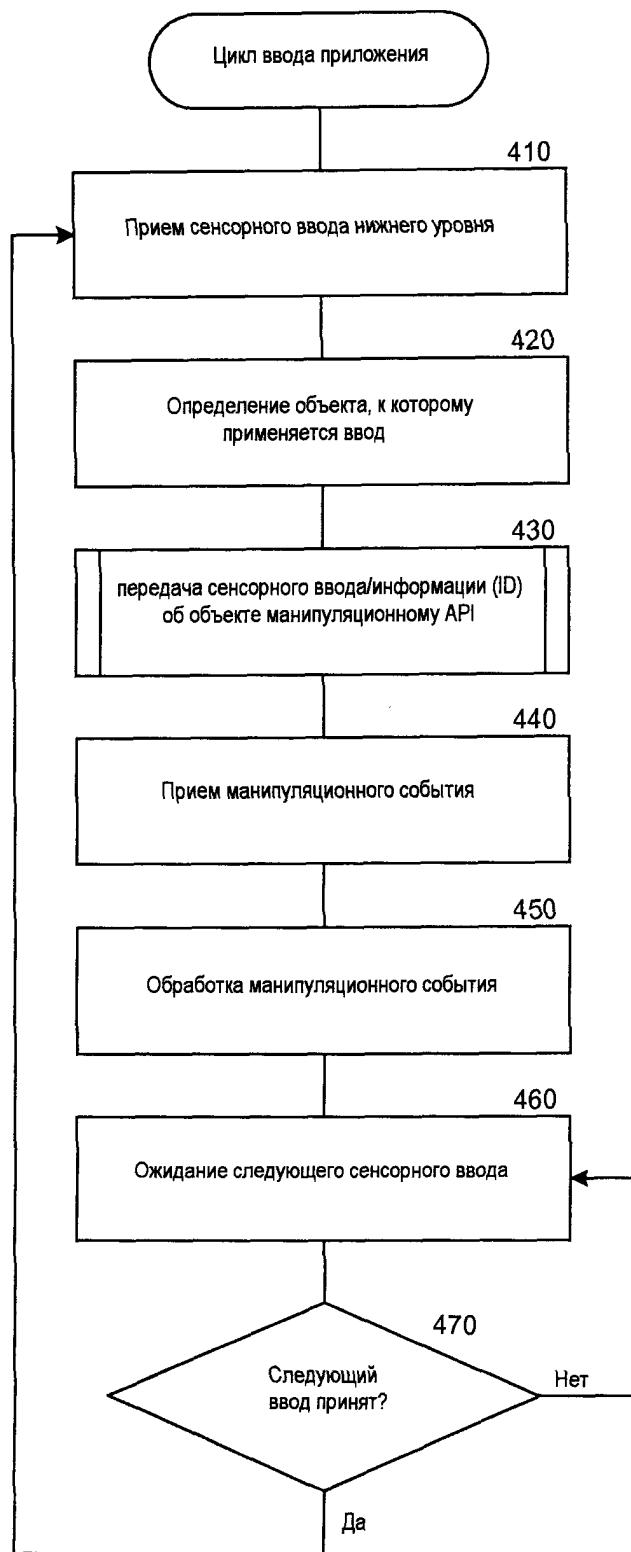
20. Компьютерное устройство по п.15, в котором воздействия до и после того, как пользователь прекратит контакт с мультисенсорным устройством ввода, имеют один и тот же формат и один и тот же канал связи с приложением.



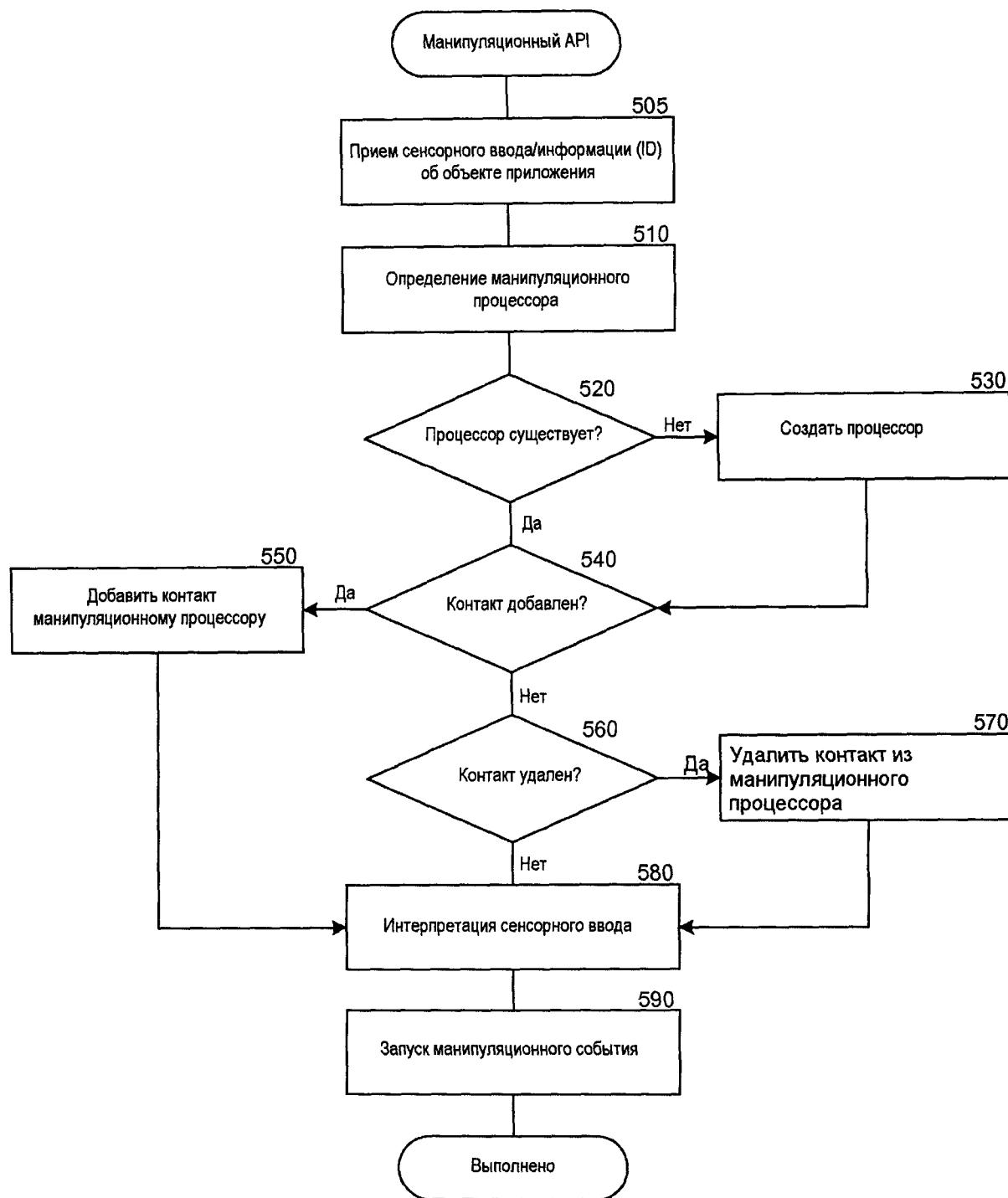
Фиг. 1



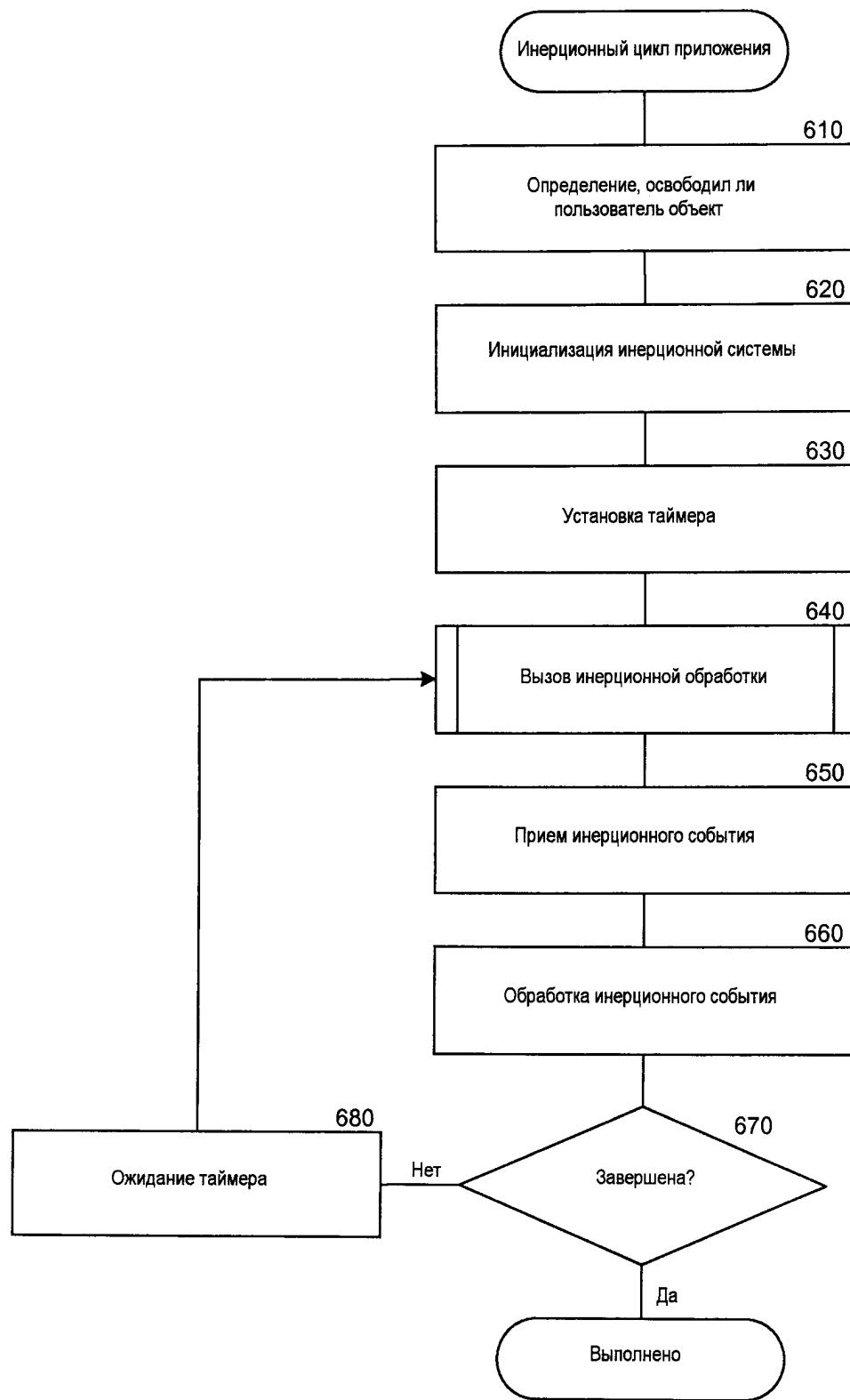
Фиг. 2



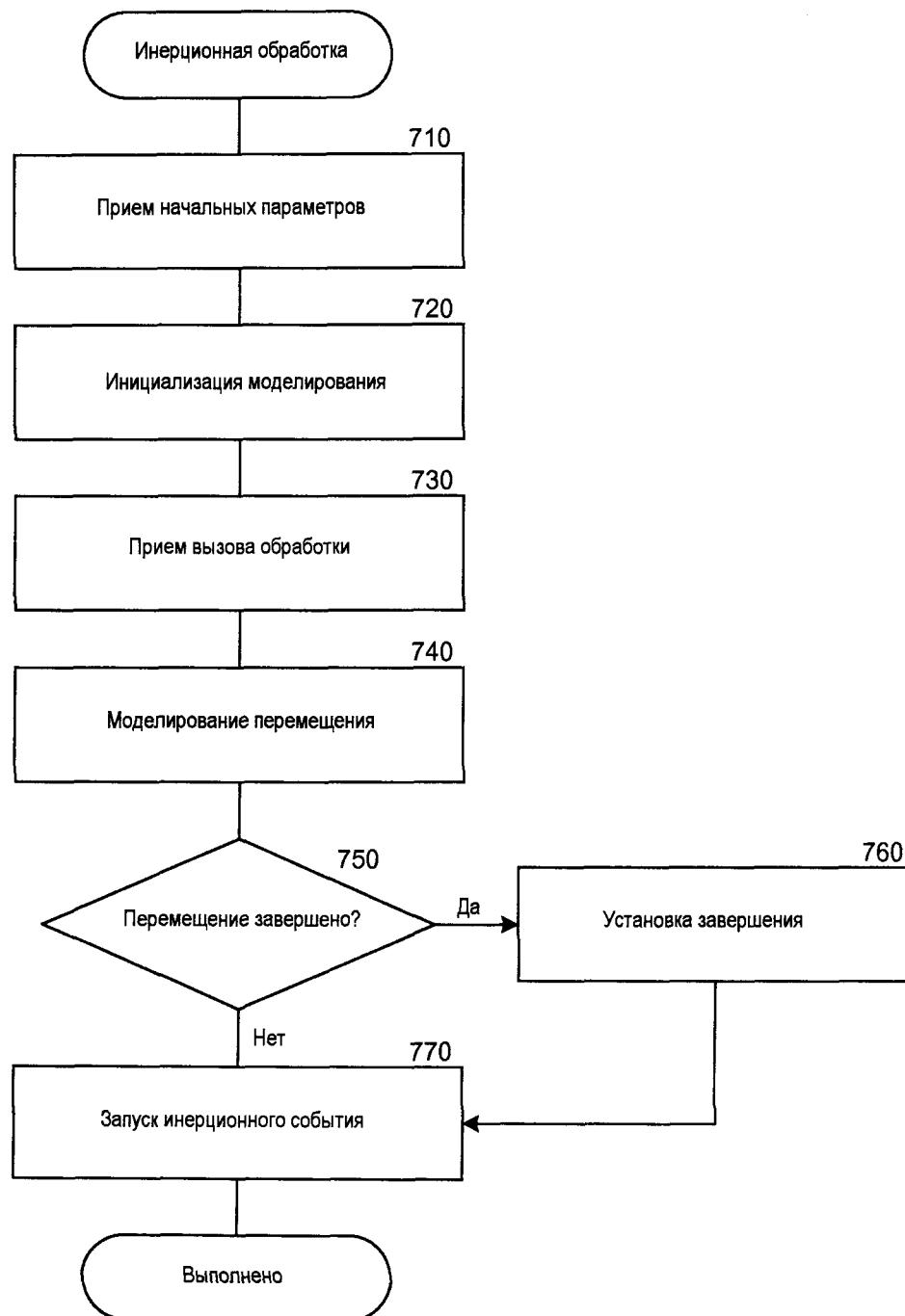
Фиг. 4



Фиг. 5



Фиг. 6



Фиг. 7