(19) State Intellectual Property Office of the People's Republic of China

(12) Invention patent

(10) Authorization Announcement No. CN 104115107 B

(45) Authorization announcement date 2018.01.09

(21) Application No. 201280055598.3

(22) Application date: 2012.09.13

(65) Published document number of the same application

Application publication number CN 104115107 A

(43) Application publication date: 2014.10.22

(30) Priority data

61/534,005 2011.09.13 US

(85) PCT international application national phase entry date

2014.05.12

(86) Application data of PCT international applications

PCT/US2012/055249 2012.09.13

(87) Publication data of PCT international applications

WO2013/040270 EN 2013.03.21

(73) Patentee Newman Unlimited

Address California, USA

(72) Inventor: M.A. Newman

(74) Patent Agency: Yongxin Patent and Trademark Agency Co., Ltd.

72002

Agent: Zhang Xi Wang Ying

(51) Int.Cl.

G06F 3/0488(2013.01)

G06F 9/44(2006.01)

(56) Comparison documents

CN 101052939 A,2007.10.10,

US 2003/0095096 A1,2003.05.22,

US 2002/0191029 A1,2002.12.19,

CN 1129889 A,1996.08.28,

Examiner: Lu Xiaoguang

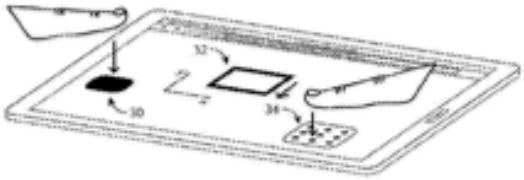
Claims: 3 pages Specification: 10 pages Drawings: 12 pages

(54) Invention title

Apparatus, method and computer-readable storage medium for manipulating user interface elements

(57) Abstract

Clutch features and boom features are disclosed herein that enable manipulation of user interface elements when using touch-sensitive components to build or otherwise design graphical displays such as websites, video games, magazine layouts, and the like. Upon touching a user interface element to be manipulated, the user interface element can be targeted for manipulation. In response to the clutch user interface element being engaged, the target user interface element can be activated for user manipulation (e.g., coloring, rotating, moving, etc.) while protecting non-target user elements from being manipulated. Booms are examples of manipulation functionality provided by some embodiments that can be configured to move a target user interface element by a precise amount (e.g., pixel by pixel).



CN 104115107 B

determining when the engaged touch event is occurring; and

When the engaging touch event is occurring, activating a manipulation function associated with the target user interface element,

wherein selecting the target user interface element for manipulation comprises: in response to determining that the selection trigger has occurred, Touch event, and the target element identifier associated with the target element is stored in a memory;

And wherein, in response to determining that the engagement touch event has begun to occur: retrieving the target element identifier from a memory; and associating the target user interface element with the manipulation function.

10. The apparatus of claim 9, wherein the processor is further configured to:

determining when the engaged touch event ceases to occur; and

The manipulation function associated with the target user interface is deactivated.

11. The apparatus of claim 10, wherein the processor is further configured to: activate a selection function that enables selection of a user interface element in the absence of the engaging touch event.

12. The apparatus of claim 9, wherein, in response to determining that the target user interface element has been selected for manipulation, the processor is further configured to cause display of a visual indicator identifying the target user interface as selected.

13. The apparatus of claim 9, wherein the processor is further configured to enable a user to manipulate the target user interface element in a manner that is permitted only while the engaging touch event is being detected.

14. The apparatus of claim 9, wherein the processor is further configured to enable a user to manipulate the target user interface element in a manner that is otherwise restricted unless the engaging touch event is detected.

15. The apparatus of claim 9, wherein the processor is further configured to support boom functionality.

16. The apparatus of claim 9, wherein the processor is further configured to facilitate display of an overlay comprising a plurality of selectable virtual buttons associated with one or more manipulation functions that are activated while the engaging touch event is occurring.

17. A computer-readable storage medium having stored therein computer-executable program code portions, the computer-executable program code portions comprising program code instructions for:

displaying a target user interface element on the touch-sensitive display;

Display clutch user interface element;

determining that a select touch event has occurred proximate to the target user interface element while an engage touch event has not occurred proximate to the clutch user interface element;

In response to determining that the select touch event has occurred, selecting the target user interface element for manipulation;

determining when the engaged touch event is occurring; and

When the engaging touch event is occurring, activating, by a processor, a manipulation function associated with the target user interface element,

Wherein, selecting the target user interface element for manipulation includes: in response to determining that the selection touch event has occurred, storing a target element identifier associated with the target element in a memory:

And wherein, in response to determining that the engagement touch event has begun to occur: retrieving the target element identifier from a memory; and associating the target user interface element with the manipulation function.

18. The computer-readable storage medium of claim 17, further comprising instructions for:

determining when the engaged touch event ceases to occur; and

The manipulation function associated with the target user interface is deactivated.

19. The computer-readable storage medium of claim 18, further comprising instructions for: activating

1. A method for manipulating a user interface element presented on a display screen, the method comprising:

displaying a target user interface element on the touch-sensitive display;

Display clutch user interface element;

determining that a select touch event has occurred proximate to the target user interface element while an engage touch event has not occurred proximate to the clutch user interface element;

In response to determining that the select touch event has occurred, selecting the target user interface element for manipulation;

determining when the engaged touch event is occurring; and

When the engaging touch event is occurring, activating, by a processor, a manipulation function associated with the target user interface element,

wherein selecting the target user interface element for manipulation comprises: in response to determining that the selection touch event has occurred, storing a target element identifier associated with the target element in a memory;

And wherein, in response to determining that the engagement touch event has begun to occur: retrieving the target element identifier from a memory; and associating the target user interface element with the manipulation function.

2. The method of claim 1, further comprising:

determining when the engaged touch event ceases to occur; and

The manipulation function associated with the target user interface is deactivated.

3. The method of claim 2 further comprises activating a selection function that enables selection of a user interface element in the absence of the engaging touch event.

4. The method of claim 1 further comprises, in response to determining that the target user interface element has been selected for manipulation, displaying a visual indicator identifying the target user interface as selected.

5. The method of claim 1, wherein activating the manipulation functionality enables a user to manipulate the target user interface element in a manner that is permissible only while the engaging touch event is being detected.

6. The method of claim 1, wherein activating the manipulation functionality enables a user to manipulate the target user interface element in a manner that is otherwise restricted unless the engaging touch event is detected.

7. The method of claim 1 , wherein activating the manipulation functionality enables an arm function for precise movement of the target user interface element on the display screen and displays an x and y pixel location associated with the location of the target user interface element.

8. The method of claim 1, further comprising displaying an overlay including a plurality of selectable virtual buttons associated with one or more manipulation functions that are activated while the engaging touch event is occurring.

9. An apparatus configured to manipulate a user interface element presented on a display screen, the apparatus comprising:

Touch-sensitive display configured for:

Display the target user interface element;

Display clutch user interface element;

detecting a select touch event; and

detecting an engagement touch event; and

A processor configured to:

determining that the select touch event has occurred proximate to the target user interface element while there is no engage touch event proximate to the clutch user interface element;

selecting the target user interface element for manipulation in response to determining that the select touch event has occurred;

A selection function of the user interface element can be selected in the absence of the engaging touch event.

20. The computer-readable storage medium of claim 17, further comprising instructions for displaying a visual indicator identifying the target user interface as selected in response to determining that the target user interface element has been selected for manipulation.

21. The computer-readable storage medium of claim 17, wherein the instructions for activating the manipulation functionality enable a user to manipulate the target user interface element in a manner that is permissible only while the engaging touch event is being detected.

22. The computer-readable storage medium of claim 17, wherein the instructions for activating the manipulation functionality enable a user to manipulate the target user interface element in a manner that is otherwise restricted unless the engaging touch event is detected.

23. The computer-readable storage medium of claim 17, wherein the instructions for activating the manipulation function enable an arm function for precise movement of the target user interface element on the display screen.

24. The computer-readable storage medium of claim 17, further comprising instructions for displaying an overlay comprising a plurality of selectable virtual buttons associated with one or more manipulation functions that are activated while the engaging touch event is occurring.

Apparatus, method and computer-readable storage medium for manipulating user interface elements

CROSS-REFERENCE TO RELATED APPLICATIONS

[0002] This application claims the benefit and priority of U.S. Provisional Patent Application No. 61/534,005, filed on September 13, 2011.

Technical field

[0003] Embodiments of the present invention generally relate to the graphic design of user interfaces.

Background technology

Software applications such as website design applications, video game design programs, presentation/drawing software, and smart phone application building tools are constantly evolving, making it possible to create, move, size, shape, color, and/or otherwise manipulate graphical objects using a mouse. As a result, actual coding to develop software is becoming less and less popular, as software can now be automatically compiled based on the graphical layout created.

The growth of graphics software creation seems to be related to the growth of the popularity of multi-touch devices such as tablet computers and smart phones. However, the ability to write software using these visualization tools mentioned above is generally limited by the lack of precision of multi-touch devices (compared to mice or other pointer devices) and the blurring of the display that occurs when a user touches the display. Therefore, graphics software creation is often more difficult to complete and is not always easily transferred to multi-touch devices, and graphics software creation is usually reserved for traditional computing devices (e.g., with a mouse and keyboard), while multi-touch devices are generally used for consumer purposes (e.g., surfing the Internet and reading media). Through the efforts, wisdom and innovations applied, many problems have been identified and solved by developing solutions included in embodiments of the present invention, examples of which are described in detail herein.

Contents of the invention

[0006] According to some exemplary embodiments, apparatus, methods, computer program products, and any other types of systems including modules for manipulating user interface elements presented on a display screen are provided herein. A touch-sensitive display screen can be provided, which is configured to: display a target user interface element (e.g., a user interface element to be selected for manipulation); display a clutch user interface element (e.g., a user interface element that engages the target user interface element for manipulation); detect a selection touch event (e.g., a user touches the display in proximity to the target user interface element); and detect an engagement touch event (e.g., a user touches the display in proximity to the clutch user interface element).

[0007] In some embodiments, a processor is also included, which is configured to: determine that the selection touch event has occurred near the target user interface element while there is no engagement touch event near the clutch user interface element; select the target user interface element for manipulation in response to determining that the selection touch event has occurred; determine when the engagement touch event is occurring; and when the engagement touch event is occurring, activate the manipulation function associated with the target user interface element.

[0008] The processor can be further configured to: determine when the engaging touch event stops occurring; and deactivate the manipulation function associated with the target user interface. Additionally or alternatively, the processor can be further configured to activate a selection function that enables selection of a user interface element while there is no engaging touch event. In some embodiments

In the example, the selection function can be activated even when an engaging touch event is detected.

[0009] A non-transitory storage medium and/or other type of memory can also be included and configured to store a target element identifier associated with the target element in response to determining that a selection touch event has occurred. The target element identifier can be unique to the target element. The target element identifier can be randomly generated, assigned when each user interface element is created, and/or generated in any other suitable manner.

In response to determining that an engaging touch event has begun to occur, the processor according to some embodiments can be configured to: retrieve a target element identifier from a memory; and associate the target user interface element with a manipulation function. In some embodiments, in response to determining that the target user interface element has been selected for manipulation, the processor is further configured to display a visual indicator identifying the target user interface as selected.

The processor can also or alternatively be configured to enable the user to manipulate the target user interface element in a manner that is only permitted when an engagement touch event is detected. As another example, the processor can be further configured to enable the user to manipulate the target user interface element in a manner that is otherwise restricted unless an engagement touch event is detected.

[0012] The processor is further configured to enable boom functionality that enables a user of a multi-touch device to precisely (e.g., pixel by pixel) move user interface elements. One or more overlays with various tabs to other menus can also be included. In addition to having one or more selectable virtual buttons associated with one or more manipulation functions that are activated while an engaging touch event is occurring, the overlays and/or other types of menus can also display information representative of the user interface being constructed, such as the location (pixel x-y-z coordinates) of a target or other user interface element and/or a cursor.

Description of the drawings

[0013] Having thus generally described the invention, reference will now be made to the accompanying drawings, which are not necessarily drawn to scale, and in which:

[0014] FIG. 1 shows a schematic block diagram of a system configured to operate in accordance with some exemplary embodiments of the present invention;

FIG. 2 shows a schematic block diagram of an apparatus configured to operate in a client and/or server according to some exemplary embodiments of the present invention:

FIG. 3 FIG. 4 show a flow chart illustrating the steps in the method according to some exemplary embodiments of the present invention. and

[0017] Figures 5-7 show example displays that may be presented by an apparatus according to some example embodiments of the present invention:

[0018] FIG. 8 shows an exemplary linear function representing how a target display element may be moved on a display screen according to some exemplary embodiments of the present invention;

9 and 10 illustrate exemplary non-linear functions representing how a target display element may be moved on a display screen according to some exemplary embodiments of the present invention; and

11-13 example displays that can be presented by devices according to some exemplary embodiments of the present invention. show

Detailed implementation

Now the present invention will be described more fully below with reference to the accompanying drawings, in which preferred embodiments of the present invention are shown.

However, the present invention may be implemented in a variety of different forms and should not be construed as being limited to the embodiments described herein.

For example, directions and orientations referred to herein include up, down, right, and left; however, it should be understood that any reference to directions and orientations is merely an example, and any particular direction or orientation may depend on the particular object to which the direction or orientation reference is to be made, and/or the orientation of the particular object. Similar references refer to similar elements throughout.

Graphic design applications such as those used for websites, video games and other graphical interfaces are becoming increasingly popular. As a result, many application developers are more likely to build graphical interfaces using such graphic design applications, as opposed to, for example, writing software code.

[0023] Tablet computers and other touch-type devices are also becoming increasingly popular. However, due to the inherent characteristics associated with touch-type devices (e.g., having to touch the display being constructed and only select discrete items, which is different from performing a broad scanning gesture), it is relatively difficult to design and build graphical interfaces using touch-type devices (compared to traditional computing devices). In solving some of these problems, FIG. 1 shows a system 10 that can benefit from an exemplary embodiment of the present invention (terms such as "example", "exemplary", etc. used herein refer to "used as an example, instance or demonstration"). As shown, the system includes one or more clients, such as a workstation 12, and may further include one or more servers, such as a server 14. In general, the workstation 12 and server 14 of the exemplary embodiment may include (e.g., include and/or be specifically implemented as) one or more portable or fixed electronic devices, such as one or more of a portable media player, a smart phone, a portable digital assistant (PDA), a tablet computer, a laptop computer, a desktop computer, a server computer, etc. In this regard, although some examples of embodiments of the present invention are disclosed herein with reference to a tablet computer with a multi-touch display, some embodiments discussed herein may also be implemented by other types of devices without departing from the essence of the present invention.

[0024] The workstation 12 and the server 14 may be configured to communicate with each other directly and/or indirectly in any of a variety of different ways, including, for example, any of a variety of wired or wireless communications or networking technologies. Examples of these technologies include, but are not limited to, universal serial bus (USB), radio frequency (RF), Bluetooth (BT), infrared (IrDA), any of a variety of different cellular (wireless) communication technologies such as any of a variety of 2G, 2.5G, 3G or 4G communication technologies, local area network (LAN), wireless LAN (WLAN) technology, and/or any other suitable technology. According to various technologies in these technologies, the client and server may be coupled to one or more networks 16 and configured to communicate across one or more networks 16. The network 16 may include any of a variety of different combinations of one or more different types of networks (including data and/or voice networks). For example, the network 16 may include one or more data networks, such as a LAN, a metropolitan area network (MAN), and/or a wide area network (WAN) (e.g., the Internet), and include one or more voice networks, such as a public switched telephone network (PSTN). Although not shown, the network 16 may include one or more devices for relaying data, information, etc. between the workstation 12 and the server 14, such as one or more routers, switches, etc.

[0025] Referring now to FIG. 2, a block diagram of circuits is shown, some or all of which may be included in an apparatus that may be configured to operate as a workstation 12 and/or server 14 or otherwise perform one or more functions of a workstation 12 and/or server 14. Although shown as separate apparatuses in FIG. 1, in some embodiments, the apparatus may support functions associated with one or both of the workstations 12 and/or servers 14 that are logically separate but co-located within the same housing. The apparatus of the exemplary embodiment of the present invention includes various modules for performing one or more functions in accordance with the exemplary embodiment of the present invention, including those modules particularly illustrated and described herein. However, it should be understood that either or both of the apparatuses may include optional modules for performing one or more similar functions without departing from the spirit and scope of the present invention.

[0026] As shown in FIG. 2 , the apparatus may include a processor 18 connected to a memory 20. According to some exemplary embodiments, the circuit may include various components, such as the processor 18, the memory 20, the communication module 24 and/or the input/output module 28.

In some embodiments, such as when the circuit is included in a tablet device and/or other device using a touch-sensitive interface component. As referred to herein, a "module" includes hardware, software, and/or firmware configured to perform one or more specific functions. In this regard, the device of the circuit as described herein may be specifically implemented as, for example, a circuit, a hardware element (e.g., a suitably programmed processor, a combinational logic circuit, and/or the like), a computer program product including computer-readable program instructions stored in a non-transitory computer-readable medium (e.g., memory 20) that can be executed by a suitably configured processing device (e.g., processor 18), or some combination thereof.

Processor 18 can be embodied as various devices, for example, including one or more microprocessors with digital signal processors, one or more processors without digital signal processors, one or more coprocessors, one or more multi-core processors, one or more controllers, processing circuits, one or more computers, other various processing elements including integrated circuits such as ASIC (Application Specific Integrated Circuit) or FPGA (Field Programmable Gate Array), or some combination thereof. Therefore, although a single processor is illustrated in FIG. 2, in some embodiments, processor 18 includes multiple processors. Multiple processors can be embodied on a single computing device, or can be distributed on multiple computing devices uniformly configured to act as circuits. Multiple processors can be operatively communicated with each other, and can be uniformly configured to perform one or more functions of the circuit as described herein. In an exemplary embodiment, processor 18 is configured to execute instructions stored in memory 20 or otherwise accessible by processor 18. When executed by processor 18, these instructions can cause the circuit to perform one or more functions of the circuit as described herein.

Whether configured by hardware, firmware/software methods or providing a combination thereof, the processor 18 may include an entity capable of performing operations according to embodiments of the present invention while being configured accordingly. Thus, for example, when the processor 18 is specifically implemented as an ASIC, FPGA, etc., the processor 18 may include specially configured hardware for performing one or more operations described herein. Alternatively, as another example, when the processor 18 is specifically implemented as an executor of instructions, such as may be stored in the memory 20, the instructions may specifically configure the processor 18 to perform one or more algorithms and operations described herein, such as those discussed in conjunction with FIG. 3 and FIG. 4.

[0029] The memory 20 may include, for example, a volatile memory, a non-volatile memory, or some combination thereof. Although illustrated as a single memory in FIG. 2 , the memory 20 may include multiple memory components. Multiple memory components may be implemented on a single computing device or distributed on multiple computing devices. In various embodiments, the memory 20 may include, for example, a hard disk, a random access memory, a cache memory, a flash memory, a compact disk read-only memory (CD-ROM), a digital versatile disk read-only memory (DVD-ROM), an optical disk, a circuit configured to store information, or some combination thereof. The memory 20 may be configured to store information, data, applications, instructions, or the like for enabling the circuit to implement various functions according to exemplary embodiments of the present invention. For example, in at least some embodiments, the memory 20 is configured to buffer input data processed by the processor 18. Additionally or alternatively, in at least some embodiments, the memory 20 is configured to store program instructions executed by the processor 18. The memory 20 may store information in the form of static and/or dynamic information. The stored information may be stored and/or used by the circuit in the process of performing its function.

[0030] The communication module 24 may be embodied as any device or apparatus embodied in circuitry, hardware, a computer program product including computer-readable program instructions stored on a computer-readable medium (e.g., memory 20) and executed by a processing device (e.g., processor 18), or a combination thereof configured to receive data from another device circuit and/or send data to another device circuit. In some embodiments, the communication module 24 (similar to other components discussed herein) can be at least partially embodied as a processor 18 or otherwise controlled by a processor 18. In this regard, the communication module 24 may communicate with the processor 18, for example, via a bus. The communication module 24 may include, for example, an antenna, a transmitter, a receiver, a transceiver, a network interface card, and/or supporting hardware and/or firmware/software for enabling communication with another computing device.

The communication module 24 may be configured to receive and/or send any data that may be stored by the memory 20 using any protocol that may be used to communicate between computing devices. Thus, some or all of the memory 20 may be located remotely from the processor 18, such as on a network drive, thereby enabling cloud-based functionality. The communication module 24 may additionally or alternatively communicate with the memory 20, the input/output module 28, and/or any other component of the circuit via, for example, a bus.

Input/output module 28 can communicate with processor 18, to receive the indication of user input and/or provide audible, visual, mechanical or other output to the user. Discuss the visual output of some examples that can be provided to the user by circuit with reference to Figure 5-7 and Figure 11-13.Therefore, input/output module 28 can include, for example, support for keyboard, mouse, joystick, touch sensitive component, microphone, loudspeaker, RFID reader, barcode reader, biometric scanner and/or other input/output mechanism.Input/output module 28 can include display 26 and/or, for example, communicate with memory 20, communication module 24 and/or any other component via bus.Although more than one input/output module and/or other components can be included in the circuit, only one is shown in Figure 2 to avoid the figure being too complicated (similar to other components discussed herein).

[0032] As described above, the memory 20 may store instructions in the form of one or more software applications, modules, etc., such as software 22, which configure the processor 18 to perform steps associated with the operation of the device according to an embodiment of the present invention. Two examples of such functionality are provided herein, which involve manipulating one or more user interface elements, referred to herein without limitation as "clutch" and "boom". For example, the software 22 may configure the processor 18 to function as a dedicated machine that provides specialized functionality such as the clutch module 18A and the boom module 18B, etc., as discussed below. In some embodiments and unlike other functionality of the device discussed herein, the clutch module 18A and/or the boom module 18B can be configured to be agnostic to all operating systems, with universal application compatibility. Examples of other software applications and/or functionality that may be included are one or more web browsers, productivity applications, game applications, utility appliances, etc., as discussed below.

The memory 20 may also store content sent from the device and/or content received by the device. As described herein, the software applications may each include software operated by the device. However, it should be understood that any one or more of the software applications described herein may be implemented alternatively by firmware, hardware, or any combination of software, firmware and/or hardware without departing from the spirit and scope of the present invention.

[0034] In addition to the memory 20, the processor 18 may also be connected to at least one interface or other device for displaying, sending and/or receiving data, content, etc., such as in accordance with USB, RF, BT, IrDA, WLAN, LAN, MAN, WAN (e.g., Internet), PSTN technology, etc. In this regard, the interface may include at least one communication interface 24 or other device for sending and/or receiving data, content, etc. In addition to the communication interface, the interface may also include at least one user interface, which may include one or more headphones and/or speakers, a display 26, and/or a user input interface 28. In turn, the user input interface may include any of a plurality of components and/or devices that allow the device to receive data from a user, such as a microphone, a keypad, a touch-sensitive surface (integrated with a display or separate from the display), a joystick, or other input device. In some embodiments, the clutch module 18A and/or the boom module 18B can be optimized and/or otherwise configured to enable more efficient and accurate user interaction and/or control of a graphical object via a multi-touch input interface.

[0035] As described above, the device can store software applications executed by the processor 18 to act as a special-purpose computer and perform specialized steps associated with the operation of the device in accordance with an embodiment of the present invention, and two examples of such specialized functions, namely clutch and boom, are provided for configuring the processor 18 as a special-purpose machine to create one or more user interface elements. Figures 3 and 4 are flow charts showing the operation of the front end (e.g., user interface interaction) and back end (e.g., device) of the clutch module 18A and boom 18B in accordance with an exemplary embodiment of the present invention. Figures 5-11 are example displays that may be presented by a device implementing the clutch module 18A and boom module 18B, including the presentation of user interface elements for each module. As shown in the figures and as described herein, the user interface elements presented during the operation of the clutch module 18A may be referred to as clutch.

Similarly, the user interface elements presented during operation of the boom module 18B may be referred to as boom user interface (UI) elements, boom elements, or simply booms.

[0036] As shown, the clutch is a dedicated mechanism for engaging and disengaging program functions within another software application (e.g., a web browser) of a computing environment, and is particularly useful for touch interface environments. For example, in some embodiments, the clutch can configure the device to provide a mechanism for engaging and disengaging program functions within a browser and/or other application of a touch interface environment. By functioning within an application (application) rather than an operating system, the clutch can be agnostic to all operating systems and have universal application compatibility. Referring to Figures 3 and 5, by engaging the clutch via a suitable clutch UI element 30, a user can adopt, control, and manipulate a variety of functions within an application, and in step S302, the user (and/or any other suitable service provider) assigns the function to the clutch.

For example, as shown in Figure 5, the UI element 32 of the application operated on the device can be selected to engage with the clutch by a single tap and/or by the user keeping a finger on the display screen near the clutch UI element 30 in step S304, as shown in Figures 6, 7 and 13. The UI element 32 can be a design element of any type that can be manipulated and/or displayed by a touch-sensitive display. Although only a single UI element is shown in Figures 5-7, 12 and 13, the clutch can allow the user to switch between the different elements of the application for manipulation or otherwise indicate and select (e.g., highlight) the different elements of the application for manipulation. Multiple functions (such as those related to, for example, moving, coloring, bolding, grouping, otherwise editing/formatting, etc.) can be used to manipulate the UI element 32, while working in the application where the clutch is engaged and the user element has been selected.

[0038] In step S306, an element identifier ("ID") of UI element 32 can be stored in memory, for example, in response to the device detecting a touch event near the UI element 32 being displayed. As mentioned herein, a "touch event" "near" a UI element includes any suitable touch (e.g., by a user's finger, a stylus, etc.) on a touch-sensitive display, such that the device's operating system and/or other system configurations cause the device to determine that the user has "selected" a virtual button presented by the display screen. In this regard, in response to UI element 32 being selected, a visual indication of UI element 32 being selected can be presented. For example, an indicator 602 shown in FIG. 6 can be provided after the user and/or system selects UI element 32.

[0039] In step S308, the user may touch a screen area and/or at least one on-screen button that is designated and/or associated with the clutch functionality discussed herein, such as the clutch UI element 30. The clutch module 18A may respond to the system determining that the clutch UI element 30 has been selected by locating the element ID in a memory (e.g., memory 20).

In step S310, function can be assigned to selected UI element 32. For example, while clutch UI element 30 is selected, mobile function, coloring function, and/or any other manipulable function can be assigned to UI element 32. In some embodiments, various manipulable functions can be associated with one or more menus that can be presented as palette, color band and/or any other suitable format, and some examples of these formats are shown in Figures 11-13. The function assigned to UI element 32 can be selected by the user (for example, by touching the manipulable function included in the displayed palette overlay), and/or automatically selected by the system, and can be determined before or after the element ID of UI element 32 is stored in the memory, and/or before/after the clutch UI element 30 is selected by the user.

In step S312, the selected UI element 32 can then display the assigned function (e.g., change color, move, rotate, etc.). In some embodiments, the manipulable function is activated in response to further user interaction. For example, the user can touch the clutch UI element 30 while dragging another finger on the screen, thereby moving the UI element 32 on the screen.

[0042] When the user releases the clutch (e.g., stops touching the portion of the touch screen proximate to the clutch UI element 30), in step S314, the device can be configured to disengage the manipulation function associated with the UI element 32.

If the finger is dragged, the UI element 32 will not move on the screen as it would when the clutch UI element 30 is selected. In some embodiments, the UI element 32 may not be manipulated in any way when the clutch UI element 30 is disengaged, and in some embodiments, the UI element 32 may be manipulated in a manner other than how it is manipulated when the clutch UI element 30 is engaged. In this regard, the clutch UI element 30 can act as an activation switch for a manipulable function and/or act as a toggle switch between available manipulable functions.

[0043] In some embodiments, the application operated while the clutch is in operation may also affect the manipulable functions enabled and/or disabled by engaging/disengaging the clutch UI element 30. For example, a web browser may have a first set of manipulable functions associated with the clutch UI element 30, while a video game or application design application may have a second set of manipulable functions associated with the clutch UI element 30.

3, in S316, the assigned function can be terminated and removed from the selected UI element 32, so that the UI element 30 is out of manipulation or otherwise has limited manipulable functionality that will result if the device detects a touch event (e.g., a user touches various locations on the touch-sensitive screen).

[0045] By operating in an application rather than within the operating system, the boom can add supplemental functionality to the native operating system of the device, and/or add other functionality by creating independent auxiliary user interface elements 34 that can facilitate precise articulation (e.g., pixel by pixel) of application elements within a touch screen environment. When operating in a two-dimensional environment, the boom can allow articulation (movement) of application UI elements 32 in the x-axis and/or y-axis; and when operating in a three-dimensional environment, the boom function can further allow articulation of UI elements in the z-axis.

[0046] In operation, referring to Figures 4 and 5, in step S402, the user may select a target and touch the screen to select an application UI element, such as UI element 32, to engage with and be manipulated by the boom UI element 34. Similar to the discussion above regarding the clutch, in step S404 the element ID of the UI element 32 is stored in memory as the target element identifier.

[0047] In step S406, the boom UI element 34 may be selected by applying any other type of touch event by the user touching and/or approaching the display of one or more components of the boom UI element 34. The processor may then be configured to locate the element ID in a memory and assign the articulation function to the selected UI element 32. In step S408, the user may engage the boom UI element 34 (e.g., touch the direction indicator) to articulate the selected UI element 32 in the x-axis, y-axis, and/or z-axis.

As shown in Figures 6 and 7, the clutch UI element 30 must also be engaged in order to enable the boom function. When the device detects an engagement touch event at the clutch UI 30 and a portion of the boom UI element 34 is also close to the second touch event, the selected UI element 32 can move downward and/or upward on the y-axis in a manner related to the portion of the selected boom UI element 34 (as shown in Figures 6 and 7, respectively). It should be understood that the selected UI element 32 is not limited to moving on the y-axis as shown in Figures 6 and 7, and it can also move on the x-axis and/or z-axis.

[0049] The selected element targeted for movement may also be accelerated linearly and/or nonlinearly, depending on, for example, the configuration of the device, the axis/axes on which the UI element is moving, and/or for any other reason. For example, there may be multiple acceleration functions that may be applied by the system. When a higher speed change is achieved, the target UI element 32 can travel on the screen as needed, such as with a nonlinear, acceleration-based function. FIG. 8 shows an exemplary linear function x+1, while FIG. 9 and FIG. 10 show exemplary nonlinear functions, x0.5 and x^3, respectively. These functions may be applied to how the target UI element moves on the screen when the boom is used. For example, the boom function may be a linear and/or nonlinear feature. If it is nonlinear, the longer the user touches the boom button, the faster the target object moves on the screen. In these instances, time-sensitive functions can be achieved by using a timer, for example, a timer is started once the clutch is engaged. Precise hardware optimized timing

A controller can be created by retrieving the time (e.g., in milliseconds) at which the "clutch" was depressed/engaged from the processor and storing that value into a global variable in memory, and retrieving the current time and storing it into a local variable within an interval. Each time an interval is activated, the start time can be subtracted from the current time to get the total amount of time that has passed since the "clutch" has been engaged. Similar and/or identical functionality can be implemented when the boom is activated (clutch engaged or not). [0050] The movement function (e.g., "linear" vs. "accelerate") can also or additionally depend on the touch area between the finger and the touch screen to correlate the force applied to the boom UI element 34 with the acceleration of the selected UI element 32. For example, a larger touch area between the finger and the touch screen due to a larger force applied to the boom UI element 34 may cause the processor to move the selected UI element 32 faster than when a smaller force (measured by the surface area of the finger causing the touch event) is applied to the boom UI element 34. Once the element reaches its final position, the user may release the boom UI element 34 in step S410 and the directional functionality may be removed from the selected element in step S412.

Figures 11-13 show additional user interface displays that can be presented according to some embodiments of the present invention. In order to create a graphical user interface, a user may need to adopt, control and/or manipulate the multiple functions of a single user interface element in a plurality of user interface elements. Although a single user interface element has been discussed above so far, when designing a web page, a video game, a magazine layout, a serialized comic and/or any other type of graphic display, a plurality of UI elements may exist, such as UI element 32 and UI element 32B. However, only one UI element needs to be manipulated. For example, a user may want to define UI element 32 as the target of fill color. Other example manipulations include realizing gradient colors, realizing that the layer on the appearance is fuzzy or distinct, making the selected area darken or brighten, or moving the selected area or selected layer around the display screen.

For example, as shown in Figure 11, the user may expect to use online design application 1102 to build a web page. In the blank background, the user can create one or more UI elements. Then, the user may expect to fill the UI element 32 with color. The user may first select a color from the color palette displayed in the overlay diagram on the screen by touching near color option 1104. By selecting color option 1104 and then touching UI element 32, if clutch UI element 30 is engaged, UI element 32 is identified as a target user interface element. When the selected UI element 32 is the target, as described above, the element identifier of UI element 32 can be stored in the memory. Then, the user can select clutch UI element 30, thereby isolating and engaging UI element 32 for manipulation. When the clutch UI element 30 is engaged while touching UI element 32, the device can cause UI element 32 to be colored with the color associated with option 1104. If the user subsequently selects UI element 32B without disengaging clutch UI element 30, UI element 32B will remain intact and not be manipulated because the engagement of clutch UI element 30 isolates the manipulation functionality from the target UI element, thereby avoiding undesired manipulation when constructing or otherwise designing a graphical display using a multi-touch display device.

[0053] The user may also desire to enlarge the selected, target UI element 32 after it has been grayed out to cover (and hide) UI element 32B, as shown in FIG12. As another example, as shown in FIG13, UI element 32B is not included, and while clutch UI element 30 is engaged, the user can use a finger to freely move UI element 32 around within web design application 1102 without using boom UI element 34. For example, UI element 32 can track the movement of the user's finger while clutch UI element 30 is engaged, and UI element 32 is targeted for manipulation. In some embodiments, as shown in FIG12, the precise pixel position of the target UI element can be provided. For example, x-coordinate 1106 and y-coordinate 1108 can be provided to enable the user to know the coordinate reference point when building a graphical user interface. When using boom to obtain the precise position of the target UI element, x-coordinate 1106 and y-coordinate 1108 can also be used, although sometimes there are difficulties associated with the precision tolerance of the multi-touch display component.

[0054] Without the clutch function to engage and disengage specific UI elements, there is a risk that some or all of the UI elements will have event listener assigned functions for specific touch events such as "touch move", in which case the user may accidentally move the UI element. Once the clutch has been engaged, the clutch provides a mechanism for individually assigning functions to event listeners to the elements. Although some embodiments are discussed herein in terms of rules related to clutch engagement,

However, in other embodiments, the clutch function can be configured on the device such that the clutch function is agnostic to the elements and the rules for engagement of the clutch function.

[0055] According to some embodiments of the present invention discussed herein, although the examples and displays shown above relate to clutch UI elements 30 and/or other UI elements included in the same display as the display for building a graphical user interface, in some embodiments (not shown), one or more other user input components may be used, such as an external tracking pad (relative to or in addition to UI element 30) acting as a clutch starter. Similarly, one or more other features discussed herein may be implemented in any suitable manner without departing from the spirit of the present invention. In addition, as described above, FIG. 3 and FIG. 4 are flowcharts reflecting methods, systems, and computer programs according to exemplary embodiments of the present invention. It will be understood that each block or step in the flowchart and the combination of blocks in the flowchart may be implemented in various ways, such as hardware, firmware, and/or software including one or more computer program instructions. It will be understood that any such computer program instructions may be loaded onto a computer or other programmable device to make a machine, so that the instructions executed on the computer or other programmable device (e.g., hardware) produce means for implementing the functions specified in the blocks or steps of the flowchart. These computer program instructions may also be stored in a computer readable memory that may direct a computer or other programmable device to operate in a particular manner so that the instructions stored in the computer readable memory produce an article of manufacture that includes instruction modules that implement the functions specified in the blocks or steps of the flowchart. The computer program instructions may also be loaded onto a computer or other programmable device so that a series of operating steps are performed on the computer or other programmable device to generate a computer-implemented process, whereby the instructions executed on the computer or other programmable device provide steps for implementing the functions specified in the blocks or steps of the flowchart.

Therefore, the block or step of the flow chart supports the combination of the mode for realizing the specified function, the combination of the step for executing the specified function and the program instruction means for executing the specified function. It will also be understood that one or more blocks or steps of the flow chart and the combination of blocks or steps in the flow chart can be realized by a hardware-based special-purpose computer system or a combination of special-purpose hardware and computer instructions that executes the specified function or step.

[0057] After benefiting from the teachings provided in the foregoing description and the associated drawings, many variations and other embodiments of the invention described herein will be readily apparent to those skilled in the art to which the invention pertains. For example, although many of the above examples Mainly related to using a multi-touch device to create content (e.g., designing a web page, building a video game, laying out a presentation, etc.) by manipulating visual, on-screen objects, but similar functionality can be used to interact with content using, for example, a multi-touch device and/or otherwise consume content using, for example, a multi-touch device. For example, a clutch UI element 30 and/or a boom UI element 34 can be used to play a video game. For example, in a first-person shooter game, the clutch UI element 30 can be used as a trigger button, and the boom UI element 34 is used as a character and/or field of view motion control. Timing functions (e.g., how long a user touches a certain area of the screen), pressure-sensitive functions (e.g., how hard a user presses based on the touched screen surface area), and/or acceleration functions (e.g., based on how long and/or how hard a touch event occurs as discussed in conjunction with Figures 9 and 10 to apply acceleration) can also be applied to a video game environment. For example, the longer and/or harder a user touches a trigger button (e.g., a clutch UI element 30), the faster the character's firearm will fire. As another example, the clutch UI element 30 can act as a throttle in a racing and/or other type of driving game, and the longer and/or harder the user touches the clutch UI element 30, the faster the vehicle will accelerate, and the boom UI element 34 can be used to steer the vehicle (e.g., an airplane, a boat, a car, a motorcycle, etc.). As yet another example, the clutch UI element 30 and the boom UI element 34 can be used in a city simulation game, a puzzle game, and/or any other type of game to navigate (e.g., using the boom) and select for manipulation (e.g., using the clutch) a plethora of displayed selectable objects. Furthermore, due to the agnostic configuration of the embodiments discussed herein, the clutch and boom functionality can enable a user to utilize native applications (e.g., a web browser, other applications capable of viewing web-based content) to control the vehicle.

For example, a video game may be provided over the network 16 by the server 14, and the workstation 12 may be configured to play the video game solely through the multi-touch device without downloading any game-specific software or utilizing any additional user interface components (e.g., a joystick, a Bluetooth mouse, etc.).

Therefore, it should be understood that the present invention is not limited to the disclosed specific embodiments, and modifications and other embodiments should also be included in the scope of the appended claims. Although specific terms are used herein, they are only used for general and descriptive meanings, rather than for the purpose of restriction.

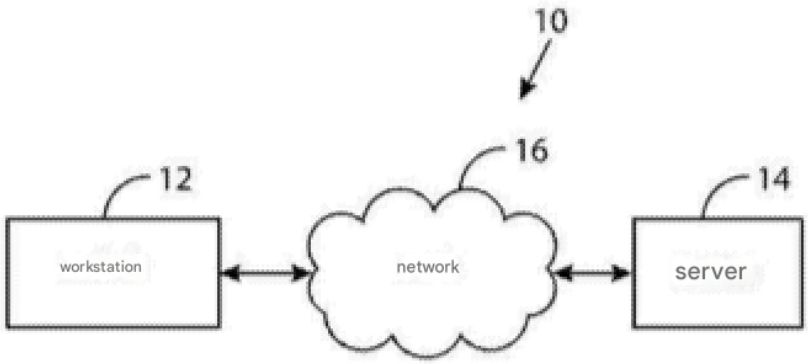


Figure 1

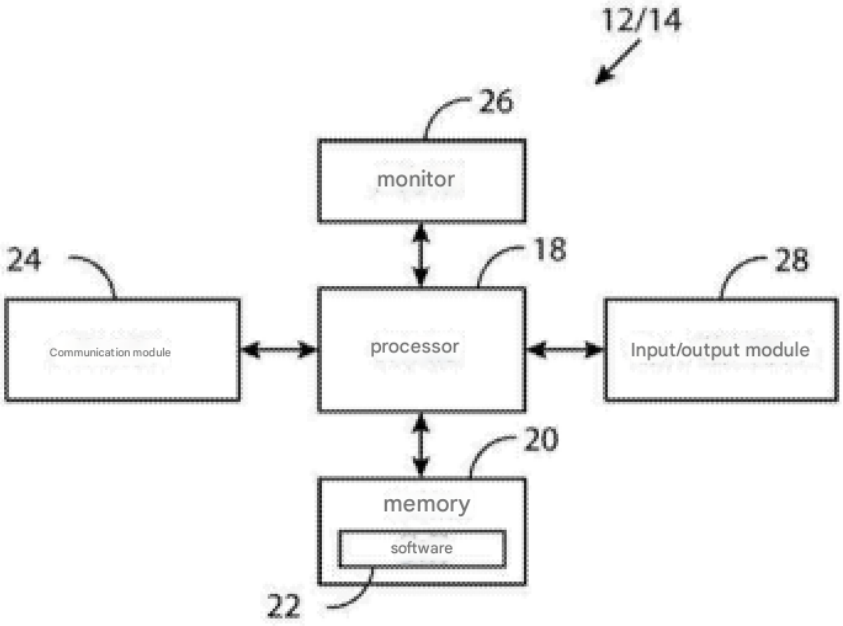


Figure 2

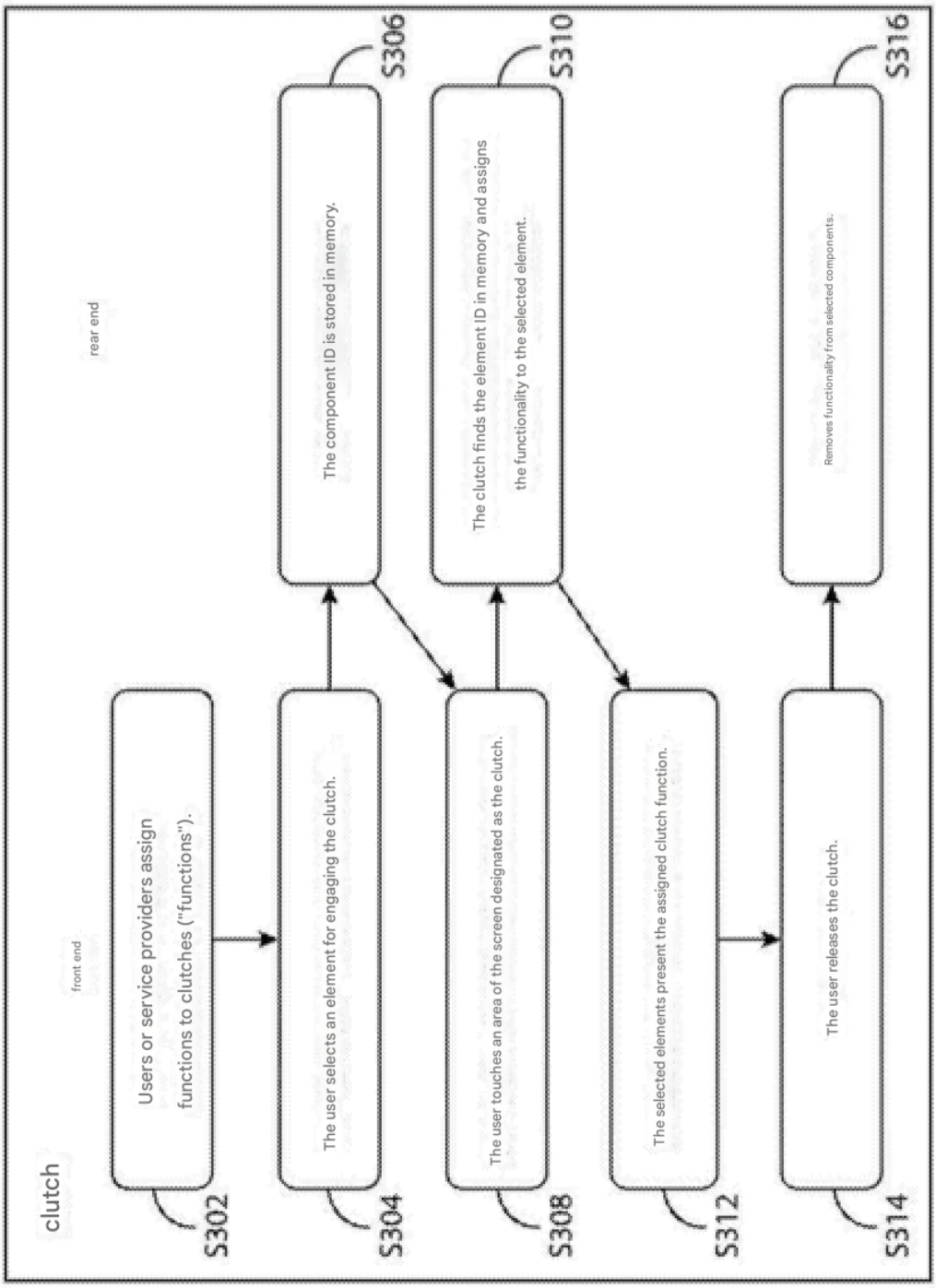

clutch
front end
rear end
Users or service providers assign functions to clutches ("functions").
S302
The user selects an element for engaging the clutch.
S304
The component ID is stored in memory.
S306
The user touches an area of the screen designated as the clutch.
S308
The clutch finds the element ID in memory and assigns the functionality to the selected element.
S310
The selected elements present the assigned clutch function.
S312
The user releases the clutch.
S314
Removes functionality from selected components.
S316

Figure 3

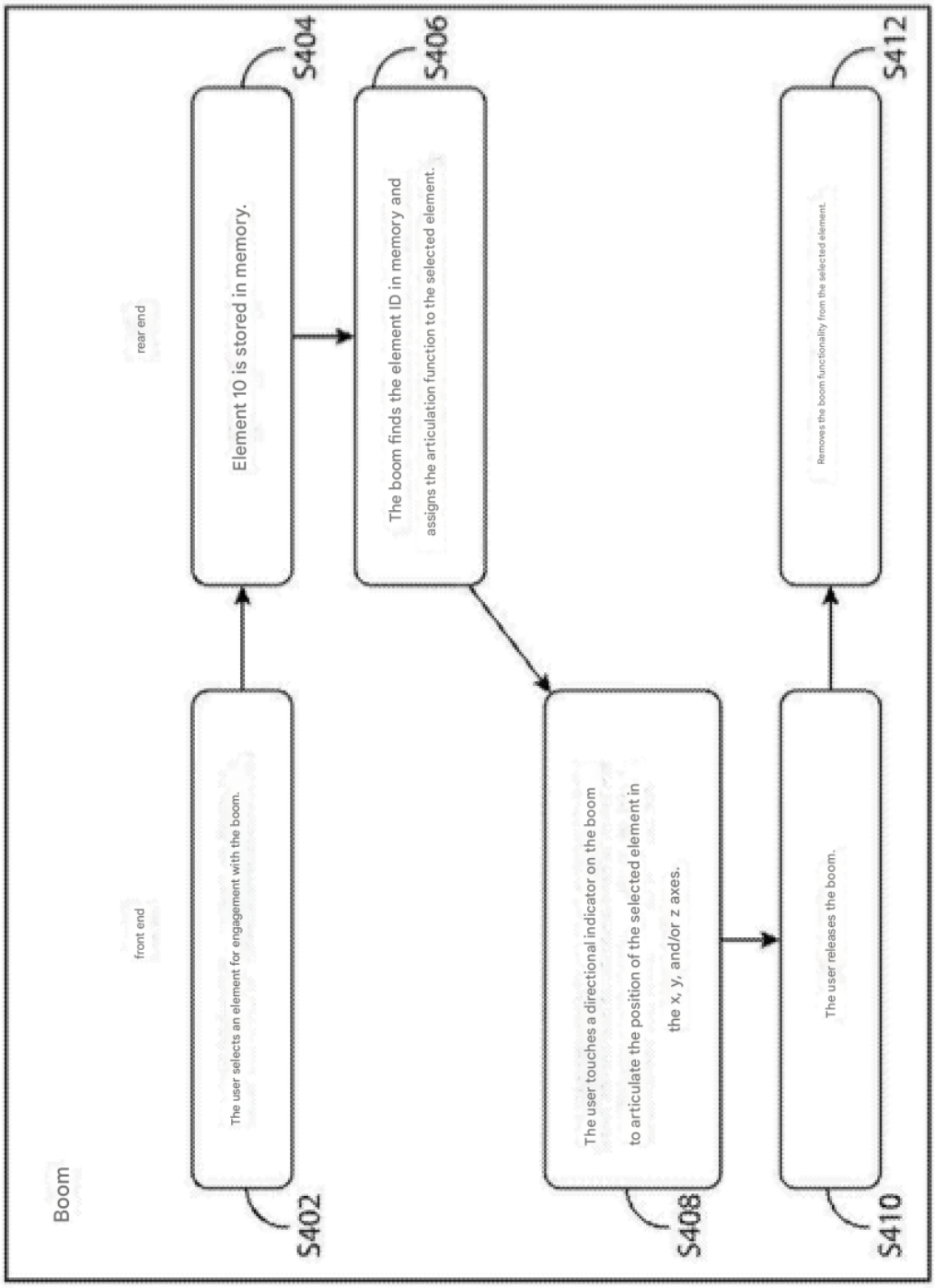

Boom
front end
rear end
The user selects an element for engagement with the boom.
S402
Element 10 is stored in memory.
S404
The boom finds the element ID in memory and assigns the articulation function to the selected element.
S406
The user touches a directional indicator on the boom to articulate the position of the selected element in the x, y, and/or z axes.
S408
The user releases the boom.
S410
Removes the boom functionality from the selected element.
S412

Figure 4

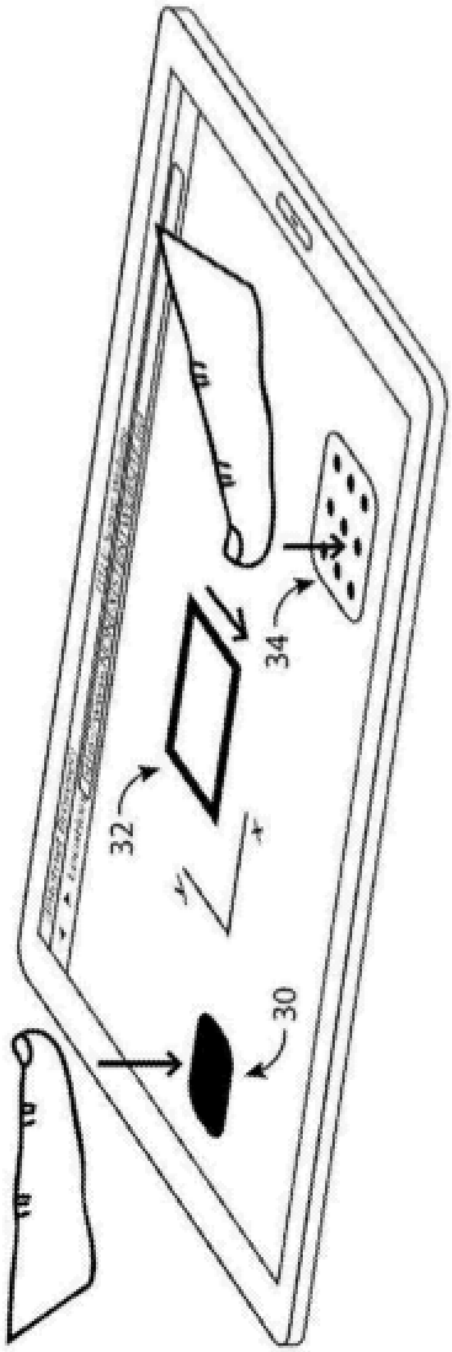


Figure 5

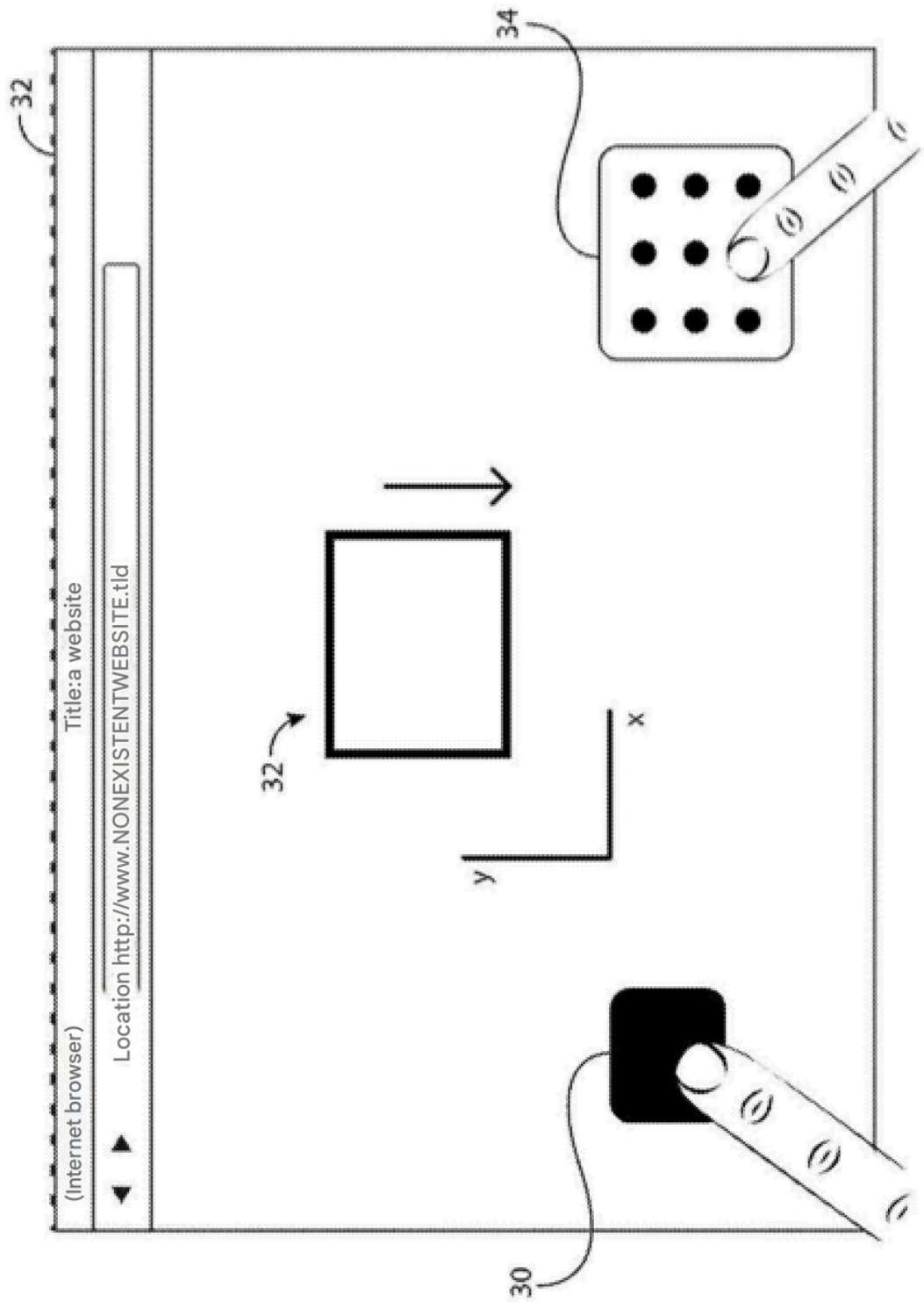

32
34
(Internet browser)
Title:a website
Location http://www.NONEXISTENTWEBSITE.tld
32
x
y
30

Figure 6

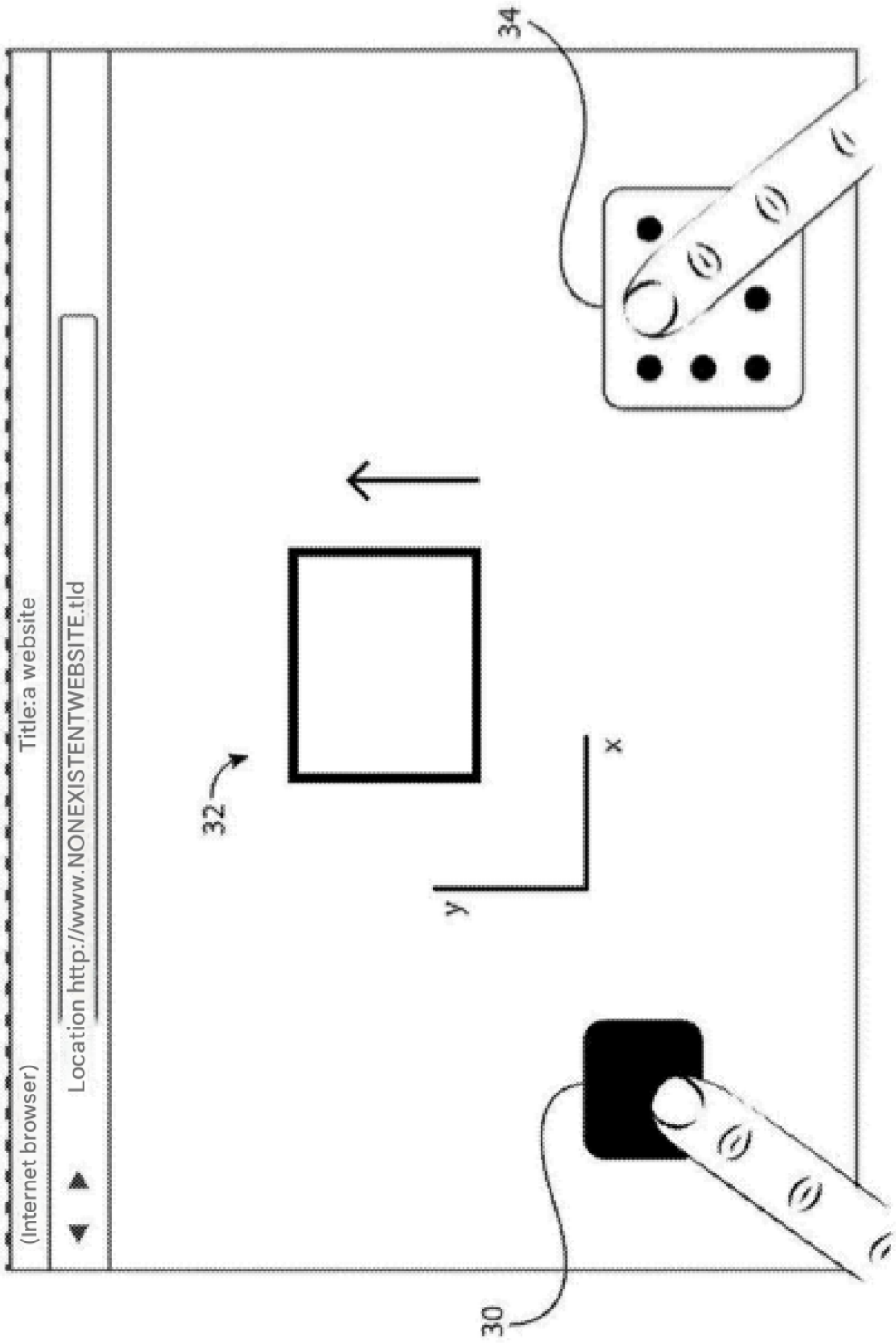

34
(Internet browser)
Title:a website
Location http://www.NONEXISTENTWEBSITE.tld
32
x
y
30

Figure 7

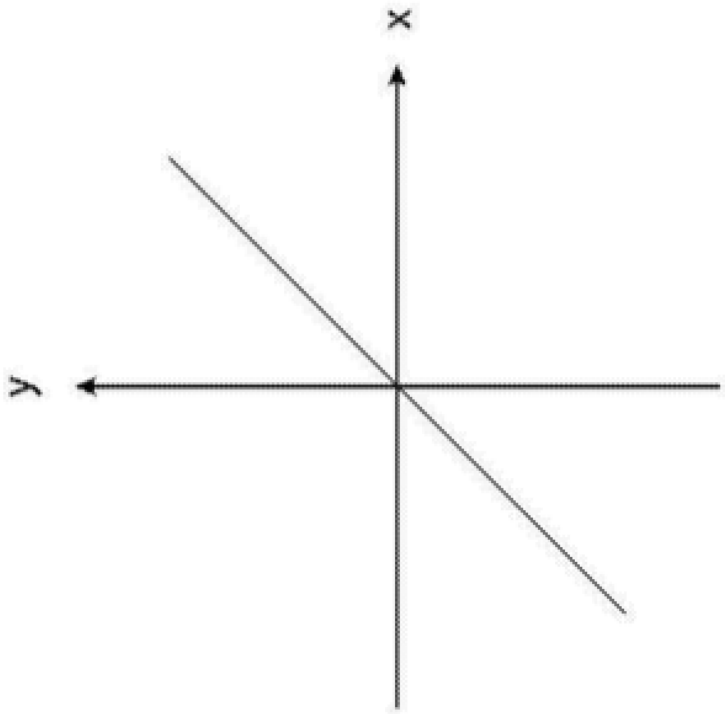


Figure 8

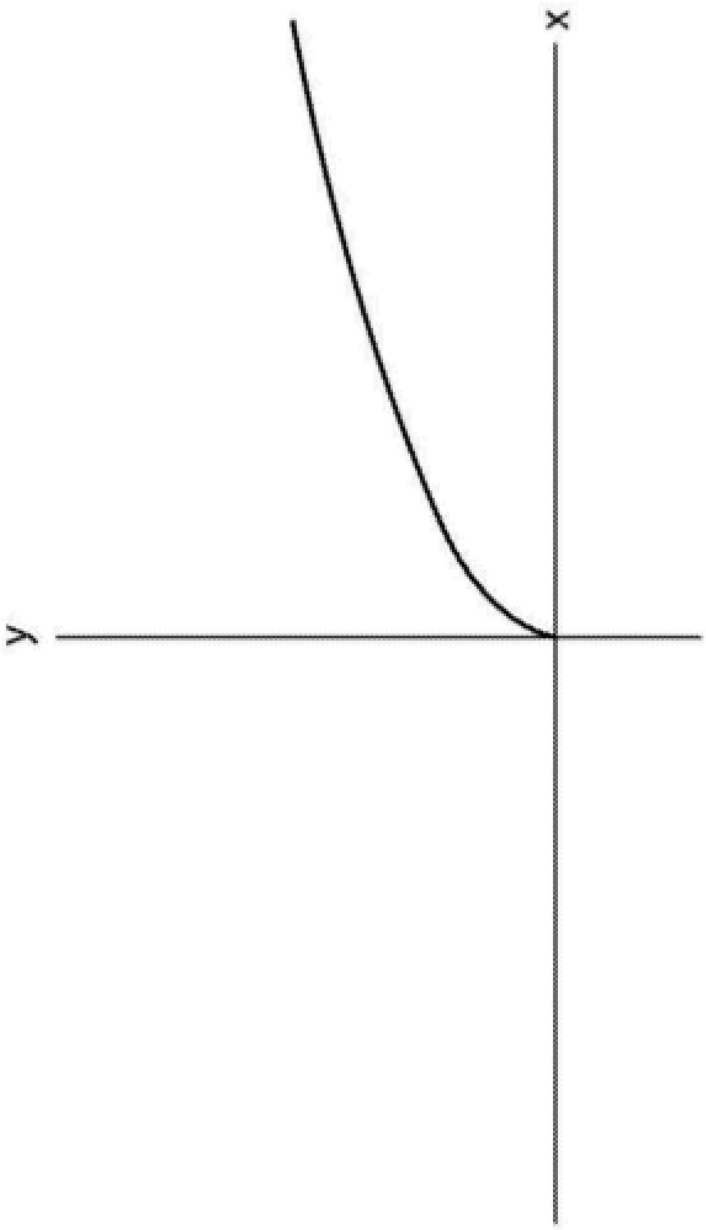


Figure 9

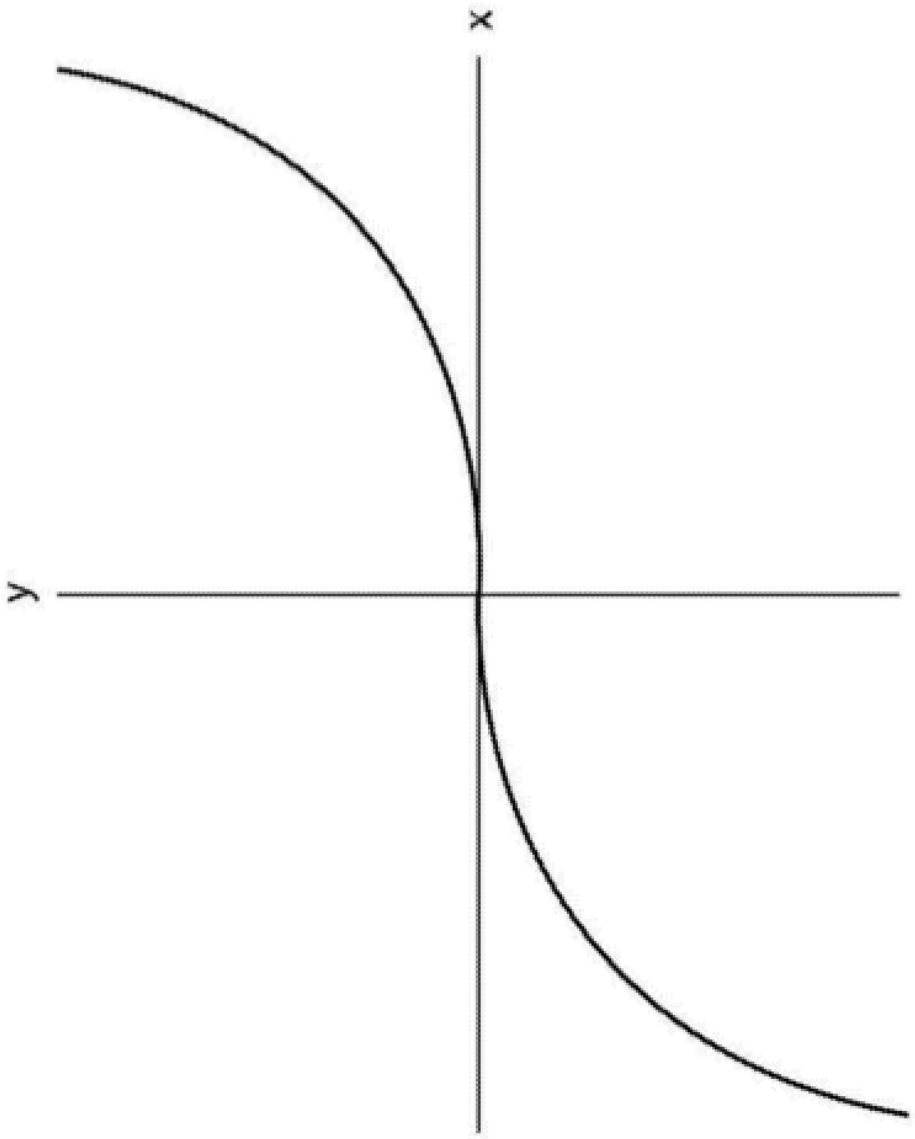


Figure 10

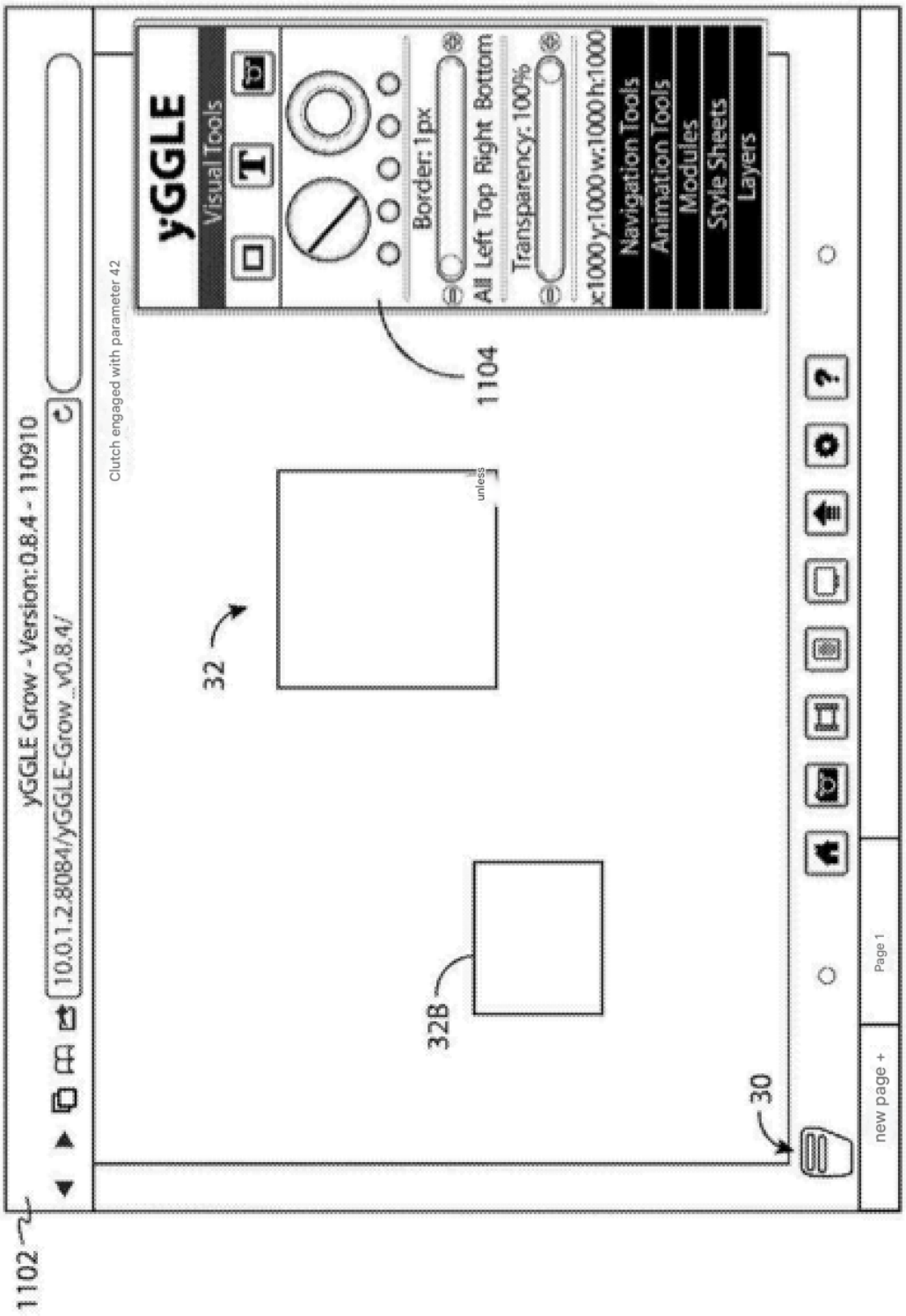

1102
yGGLE Grow - Version:0.8.4 - 110910
10.0.1.2:8084/yGGLE-Grow_v0.8.4/
Clutch engaged with parameter 42
yGGLE
Visual Tools
Border: 1px
All Left Top Right Bottom
Transparency: 100%
x:1000 y:1000 w:1000 h:1000
Navigation Tools
Animation Tools
Modules
Style Sheets
Layers
1104
32
unless
32B
30
new page +
Page 1

Figure 11

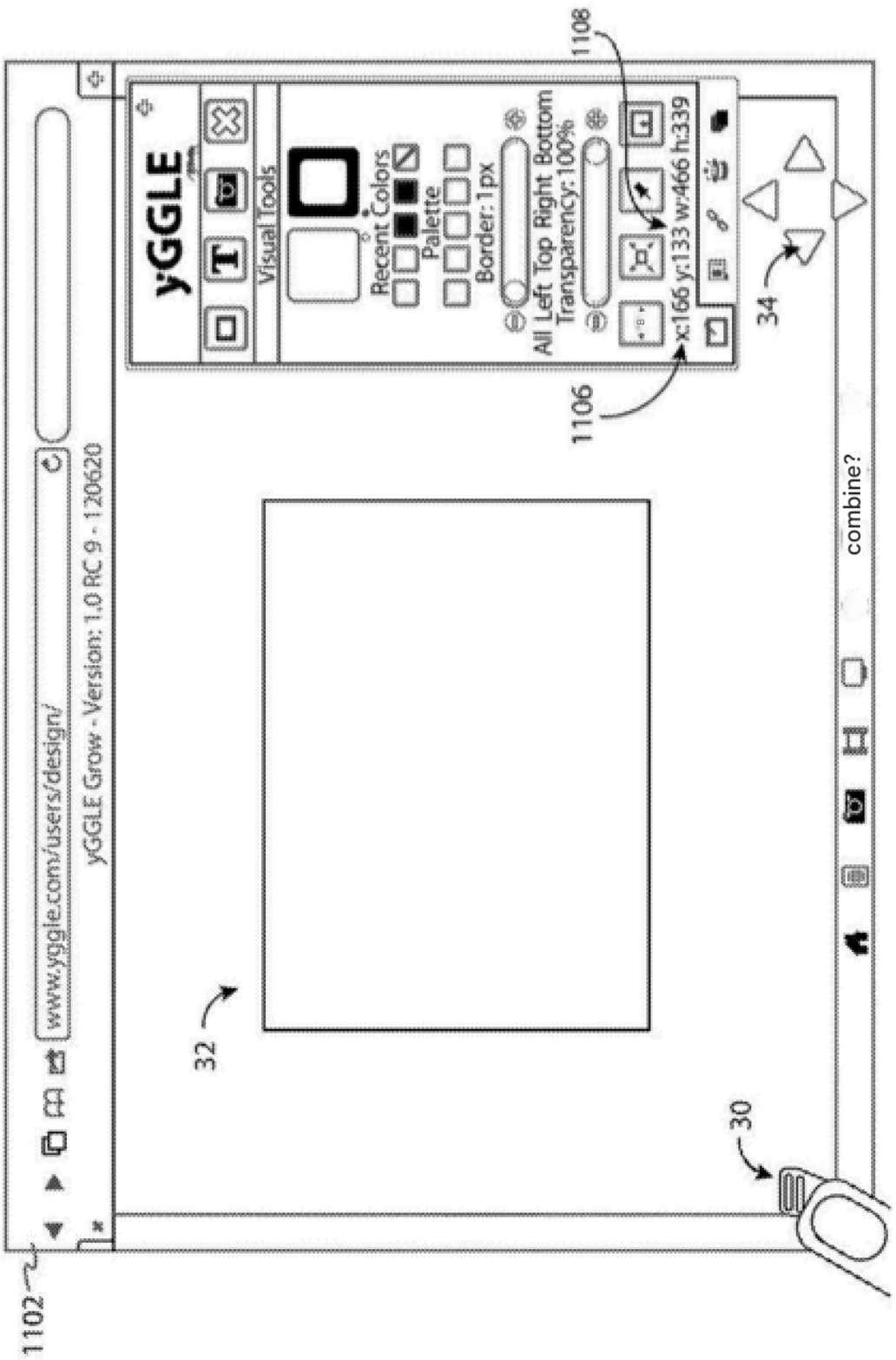

1102
www.yggle.com/users/design/
yGGLE Grow - Version: 1.0 RC 9 - 120620
yGGLE
Visual Tools
Recent Colors
Palette
Border: 1px
All Left Top Right Bottom
Transparency: 100%
x:166 y:133 w:466 h:339
1108
1106
34
32
30
combine?

Figure 12

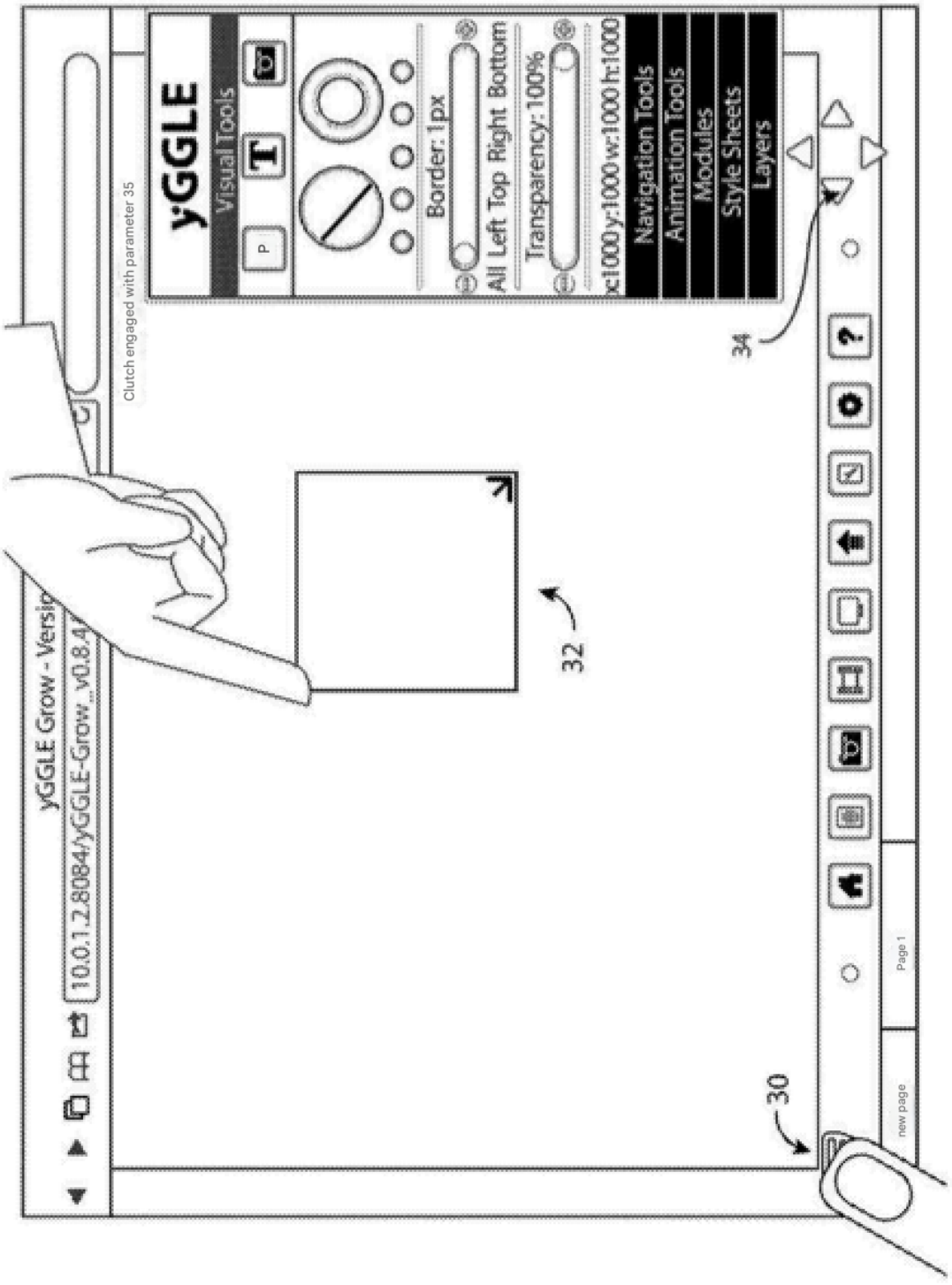

yGGLE
Visual Tools
Border: 1px
All Left Top Right Bottom
Transparency: 100%
Navigation Tools
Animation Tools
Modules
Style Sheets
Layers
Clutch engaged with parameter 35
yGGLE Grow - Versio
10.0.1.2:8084/yGGLE-Grow_v0.8.4/
new page
Page 1
30
32
34

Figure 13