US 20160110150A1

(19) United States
(12) Patent Application Publication
Sirpal et al.
(10) Pub. No.: US 2016/0110150 A1
(43) Pub. Date: Apr. 21, 2016

(54) **DISPLAYING THE DESKTOP UPON DEVICE OPEN**

(71) Applicant: **Z124**, George Town (KY)

(72) Inventors: **Sanjiv Sirpal**, Oakville (CA); **Paul E. Reeves**, Oakville (CA); **Alexander de Paz**, Burlington (CA); **Rodney W. Schrock**, San Diego, CA (US)

(21) Appl. No.: **14/885,139**

(22) Filed: **Oct. 16, 2015**

Related U.S. Application Data

(63) Continuation of application No. 13/248,138, filed on Sep. 29, 2011.

(60) Provisional application No. 61/539,884, filed on Sep. 27, 2011, provisional application No. 61/458,150, filed on Nov. 17, 2010, provisional application No. 61/389,000, filed on Oct. 1, 2010, provisional application No. 61/389,117, filed on Oct. 1, 2010, provisional application No. 61/389,087, filed on Oct. 1, 2010.

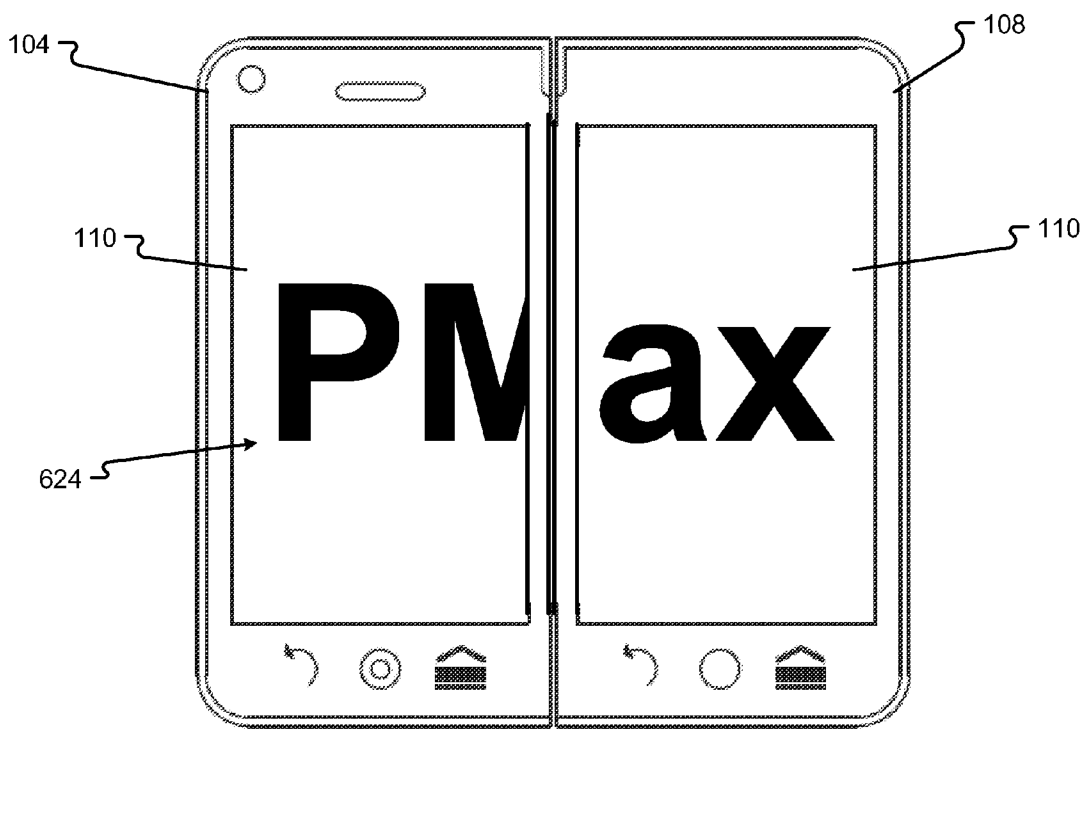
Publication Classification

(51) **Int. Cl.**
G06F 3/14 (2006.01)
G06F 3/0346 (2006.01)
G06F 3/0488 (2006.01)

(52) **U.S. Cl.**
CPC ***G06F 3/1446*** (2013.01); ***G06F 3/0488*** (2013.01); ***G06F 3/0346*** (2013.01)

(57) **ABSTRACT**

Systems and methods are provides for displaying a desktop for a multi-screen device in response to opening the device. The window stack can change based on the change in the orientation of the device. The system can receive an orientation change that transitions the device from a closed state to an open state. A previously created in the stack can expand over the area of the two or more displays comprising the device when opened. A desktop expands to fill the display area and be displayed on the second of the displays after the device is opened.



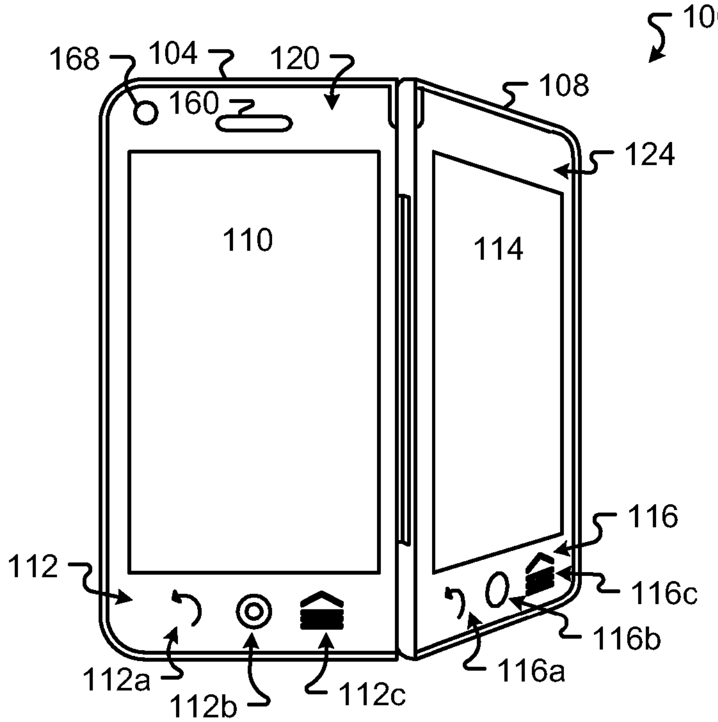

100
168
104
120
160
108
124
110
114
116
112
116c
116b
116a
112a
112b
112c

FIG. 1A

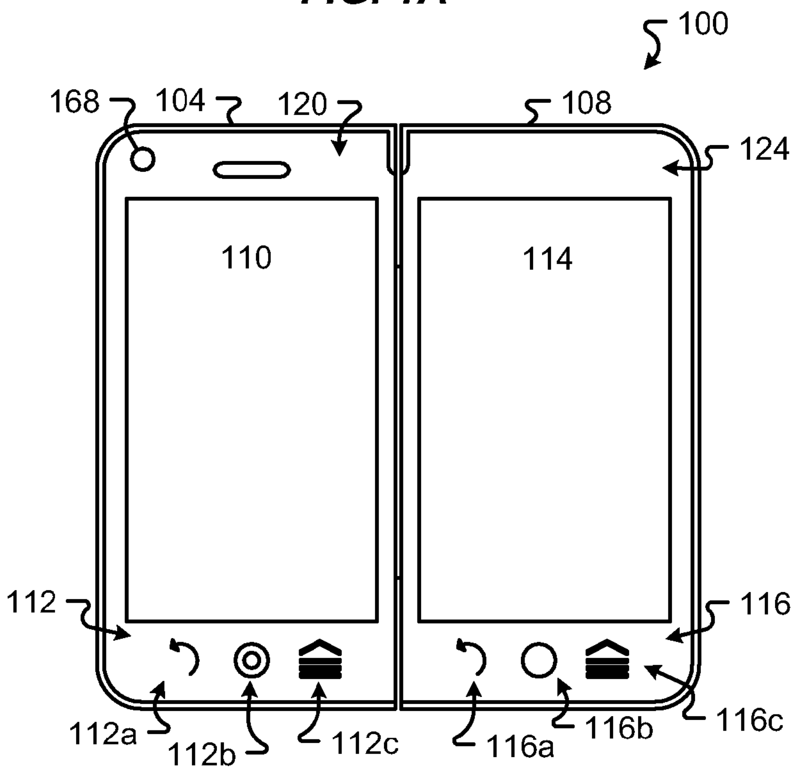

100
168
104
120
108
124
110
114
112
116
112a
112b
112c
116a
116b
116c

FIG. 1B

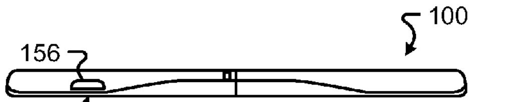


FIG. 1F

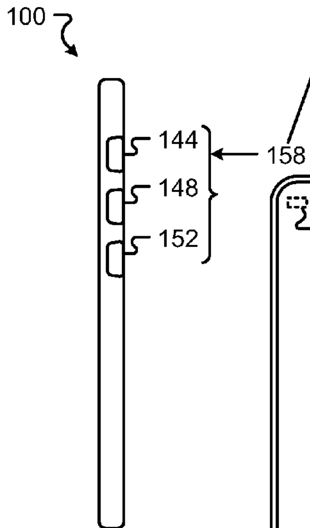


FIG. 1E

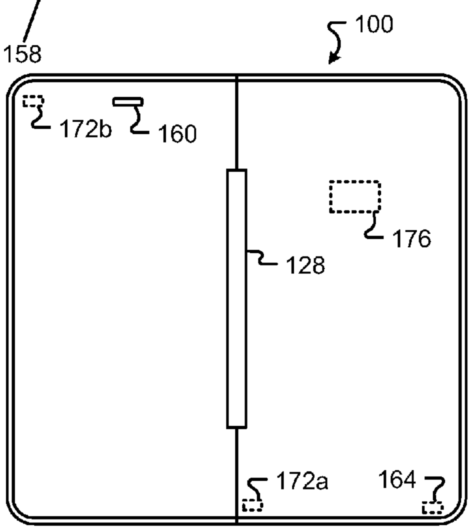


FIG. 1C

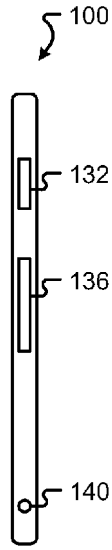


FIG. 1D

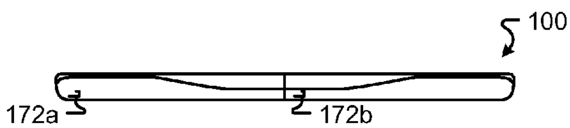


FIG. 1G

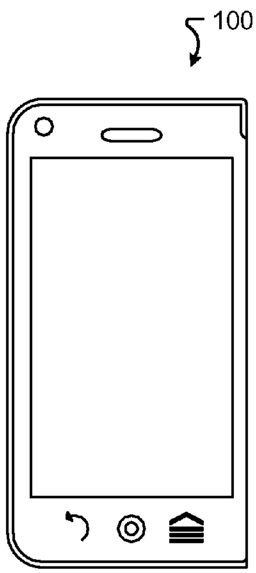


FIG. 1H

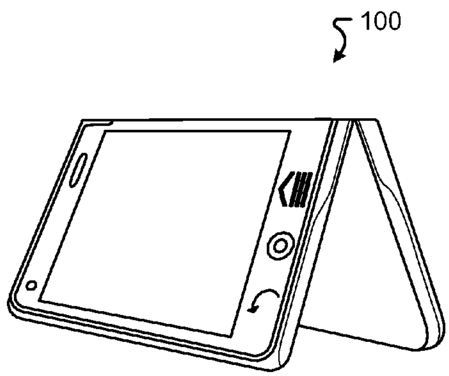


FIG. 1I

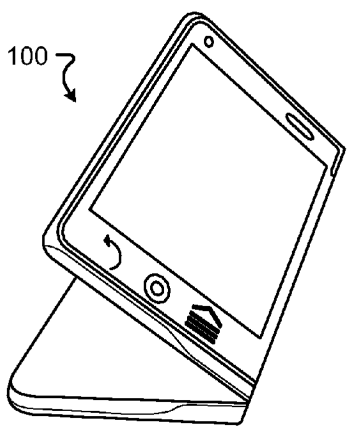


FIG. 1J

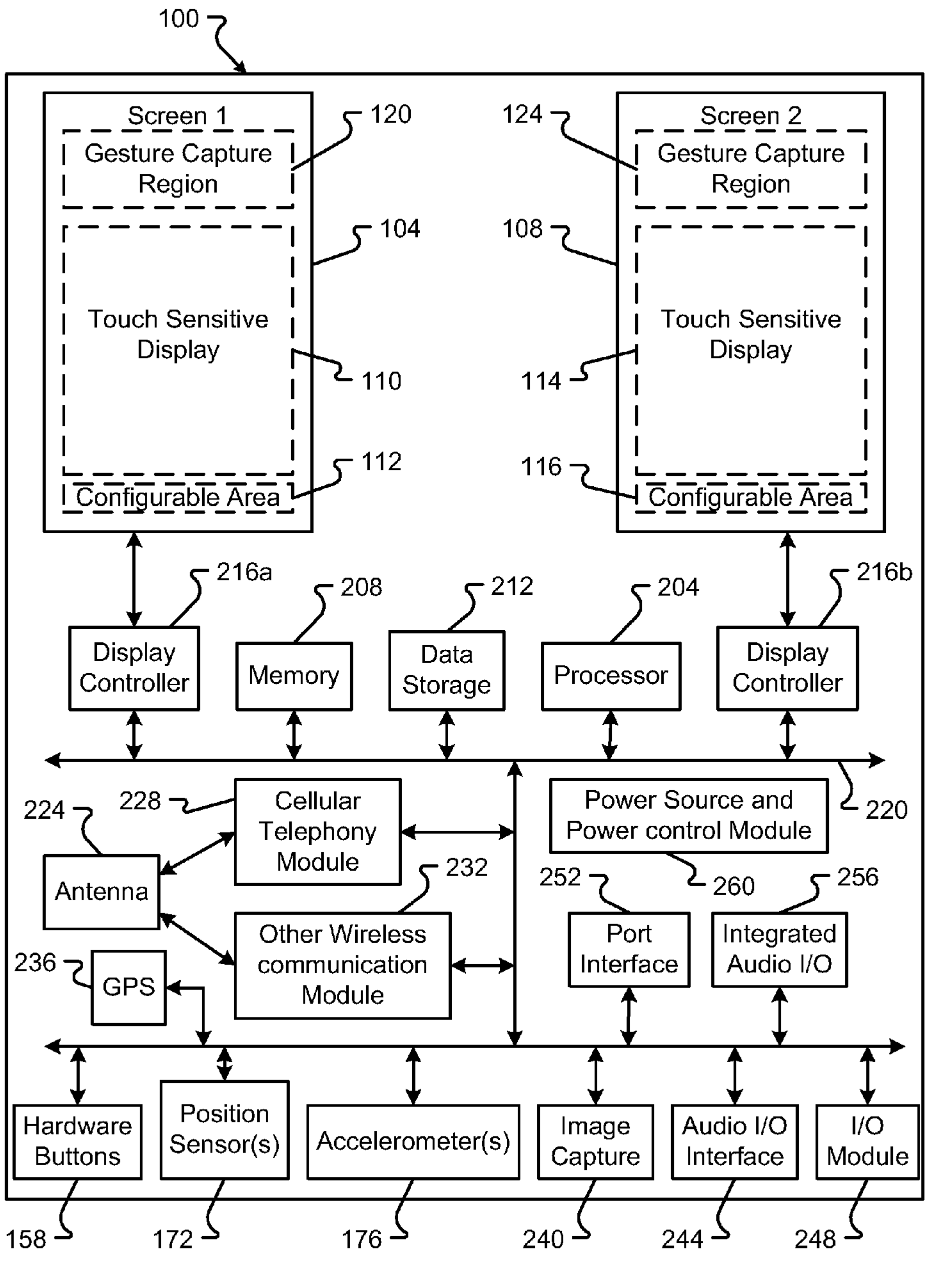

100
Screen 1
Gesture Capture Region
120
Touch Sensitive Display
104
110
112
Configurable Area
Screen 2
124
Gesture Capture Region
108
Touch Sensitive Display
114
116
Configurable Area
216a
Display Controller
208
Memory
212
Data Storage
204
Processor
216b
Display Controller
220
224
228
Cellular Telephony Module
Antenna
232
Other Wireless communication Module
236
GPS
Power Source and Power control Module
252
260
256
Port Interface
Integrated Audio I/O
Hardware Buttons
Position Sensor(s)
Accelerometer(s)
Image Capture
Audio I/O Interface
I/O Module
158
172
176
240
244
248

FIG. 2

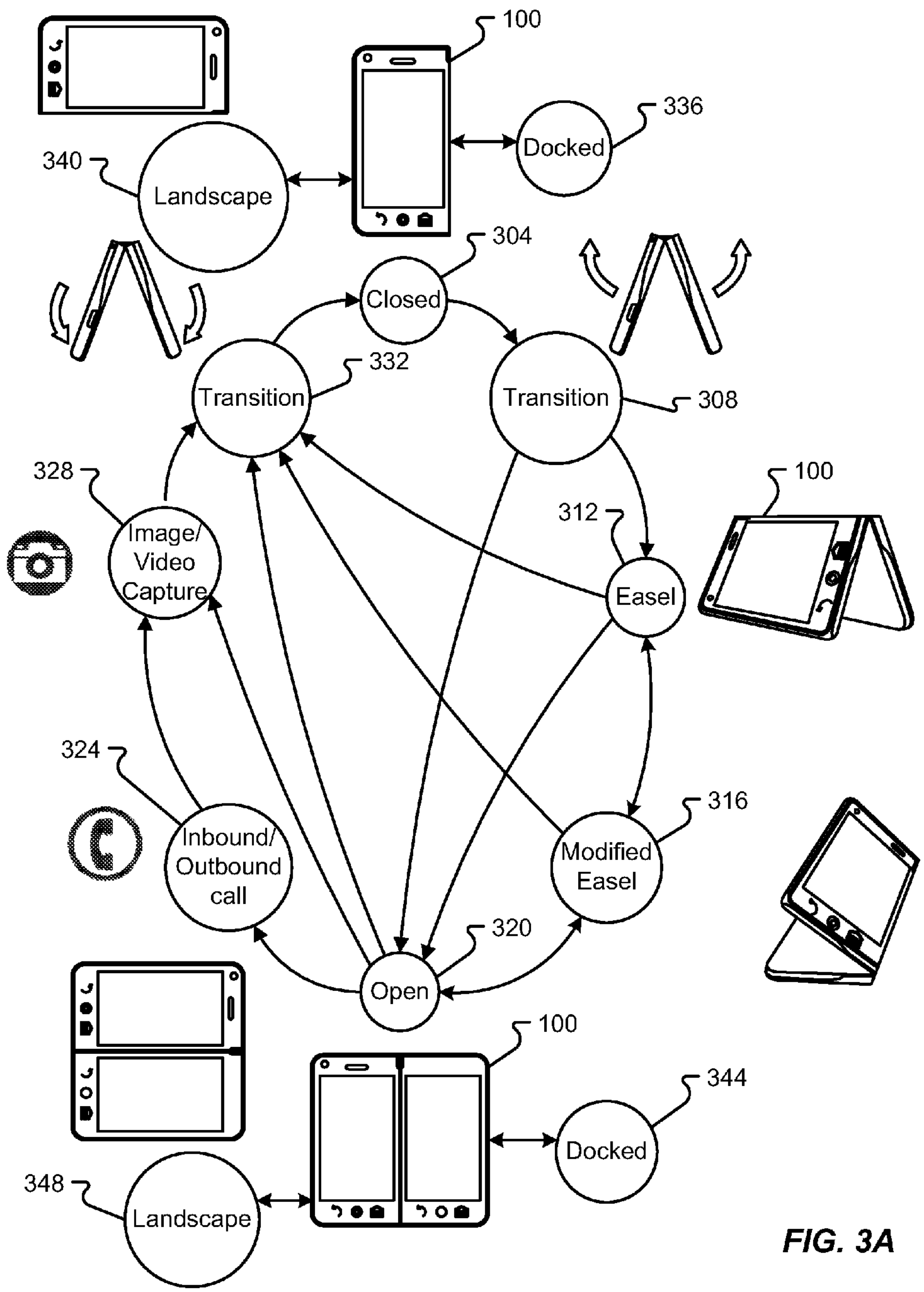

100
336
Docked
340
Landscape
304
Closed
332
Transition
Transition
308
100
312
328
Image/
Video
Capture
Easel
324
316
Inbound/
Outbound
call
Modified
Easel
320
Open
100
344
Docked
348
Landscape

FIG. 3A

376 352 356 360

		PORTRAIT						LANDSCAPE					
		OPEN	CLOSED	EASEL	MODIFIED EASEL	PHONE	IMAGE / VIDEO	OPEN	CLOSED	EASEL	MODIFIED EASEL	PHONE	IMAGE / VIDEO
PORTRAIT	OPEN	X	HT	HT	HT	P	I	AT	HAT	HAT	HAT	P	I
	CLOSED	HT	X	HAT	HAT	P	I	HAT	AT	HAT	HAT	P	I
	EASEL	HT	HT	X	X	P	I	HAT	HAT	HAT	HAT	P	I
	PHONE	HT	X	HT	HT	X	I	HAT	HAT	HAT	HAT	X	I
	IMAGE / VIDEO	HT	HT	HT	X	P	X	HAT	HAT	HAT	HAT	X	HAT
LANDSCAPE	OPEN	AT	HAT	HAT	HAT	P	I	X	HT	HAT	HAT	P	I
	CLOSED	HAT	AT	HAT	HAT	P	I	HT	X	HAT	HAT	P	I
	EASEL	HAT	HAT	HAT	HAT	P	I	HT	HT	X	HAT	P	I
	MODIFIED EASEL	HAT	HAT	HAT	HAT	P	I	HT	HT	HAT	X	P	I
	IMAGE / VIDEO	HAT	HAT	HAT	HAT	HAT	AT	HT	HT	HT	HAT	P	X
	DOCKED	DOCKING SIGNAL											

364 368 372

Key:
H - Hall Effect Sensor(s)
a - accelerometer(s)
T- Timer
P – communications Trigger
I – Image / Video capture Request

FIG. 3B

Tap

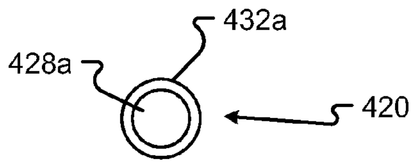


FIG. 4A

Long Press

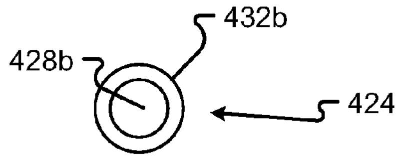


FIG. 4B

Drag

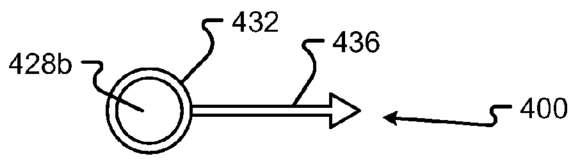


FIG. 4C

Flick

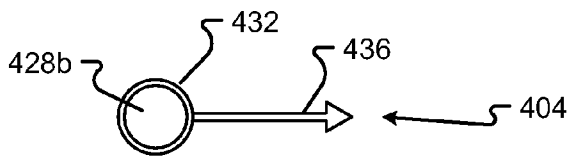


FIG. 4D

Pinch

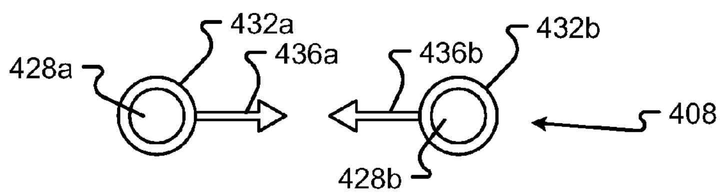


FIG. 4E

Spread

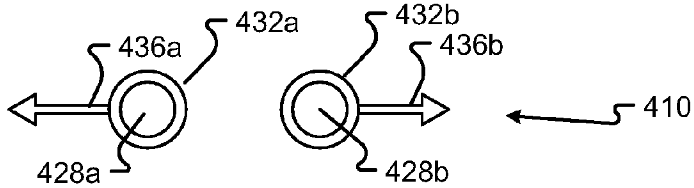


FIG. 4F

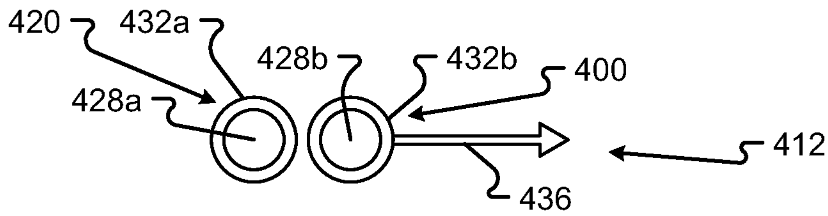

420
432a
428b
432b
400
428a
412
436

FIG. 4G

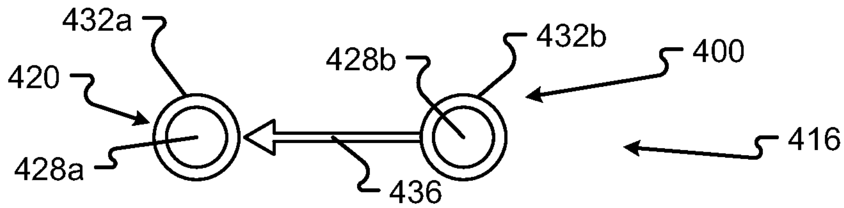

432a
432b
400
428b
420
416
428a
436

FIG. 4H

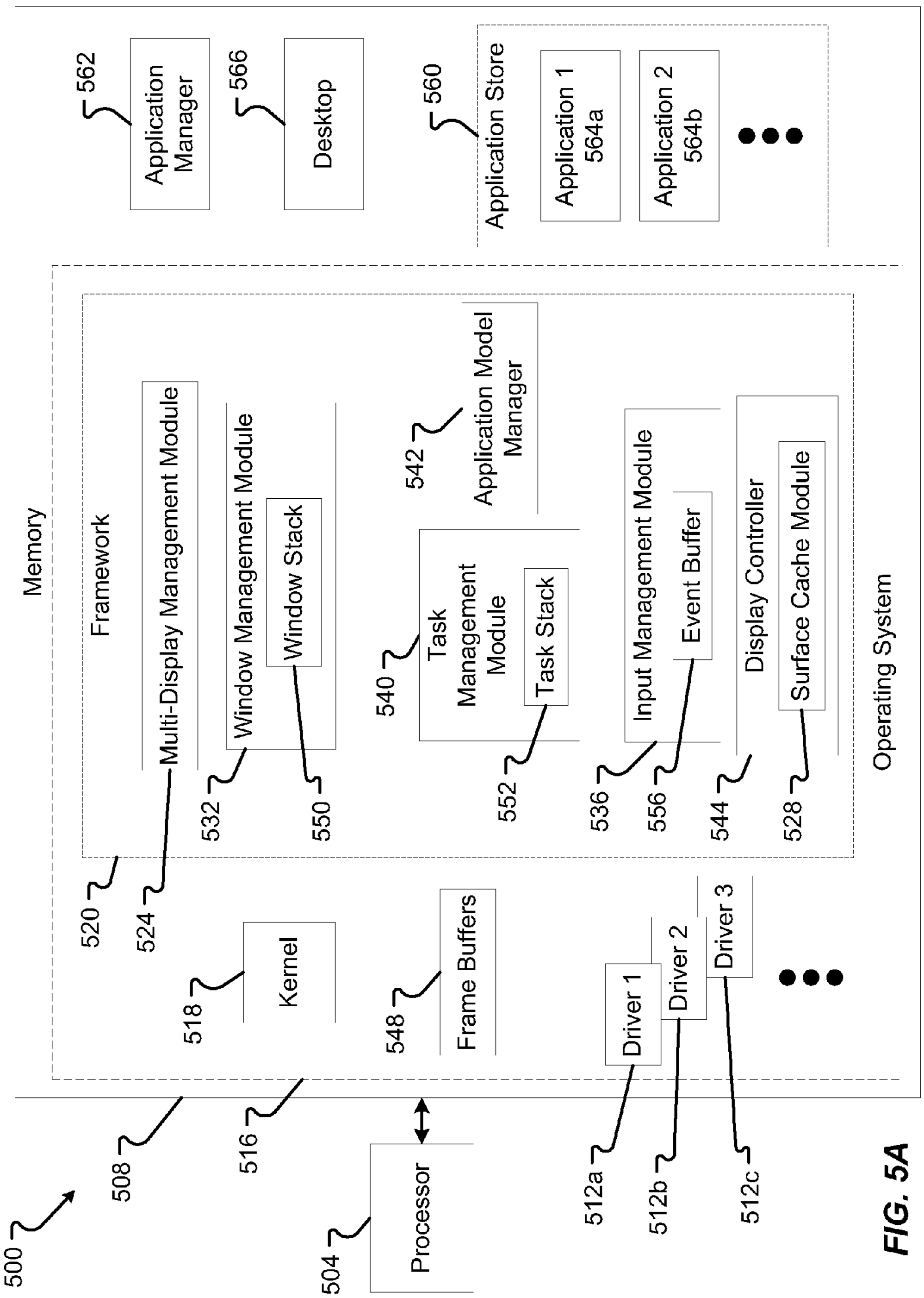

500
508
516
504
Processor
512a
512b
512c
Memory
518
Kernel
548
Frame Buffers
Driver 1
Driver 2
Driver 3
520
524
Framework
Multi-Display Management Module
532
Window Management Module
Window Stack
550
540
Task Management Module
Task Stack
552
542
Application Model Manager
536
Input Management Module
Event Buffer
556
544
Display Controller
Surface Cache Module
528
Operating System
562
Application Manager
566
Desktop
560
Application Store
Application 1 564a
Application 2 564b

FIG. 5A

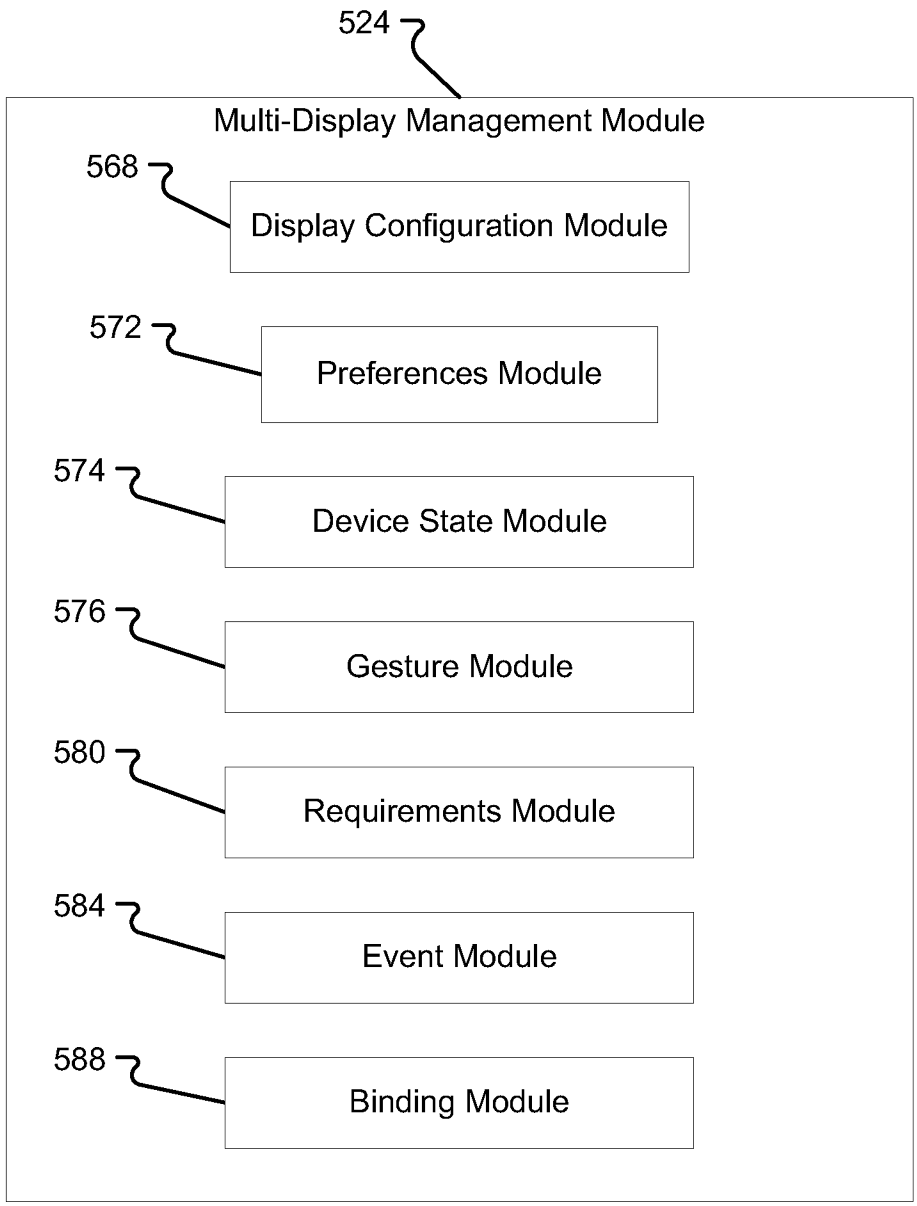

524
Multi-Display Management Module
568
Display Configuration Module
572
Preferences Module
574
Device State Module
576
Gesture Module
580
Requirements Module
584
Event Module
588
Binding Module

FIG. 5B

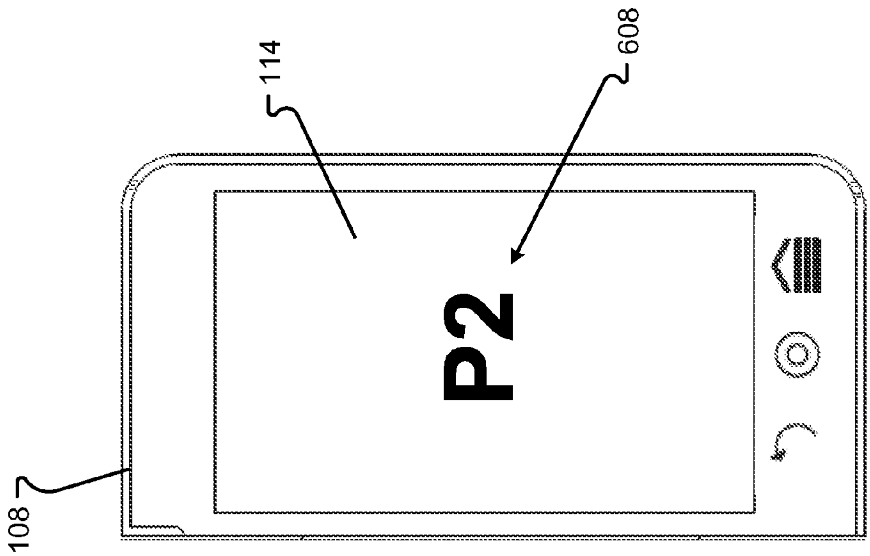


FIG. 6B

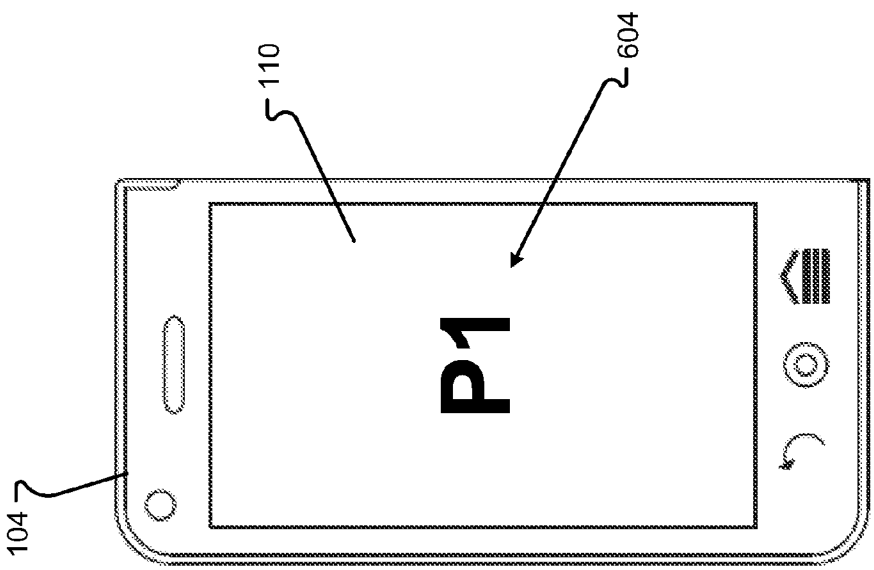


FIG. 6A

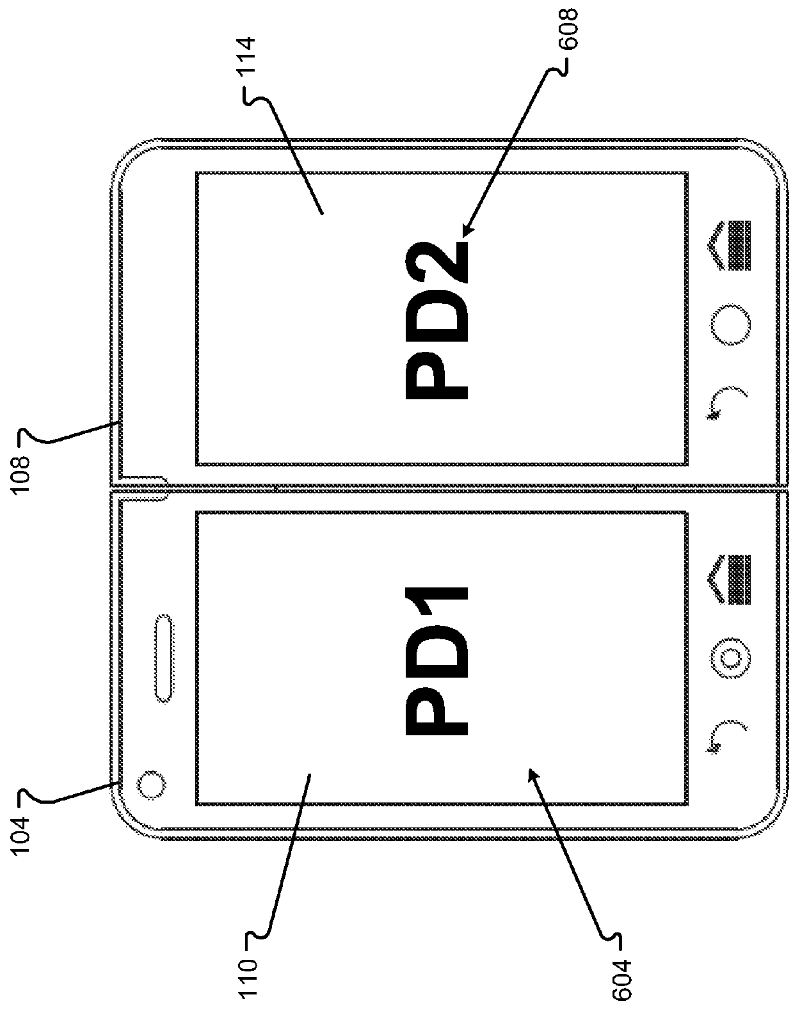


FIG. 6C

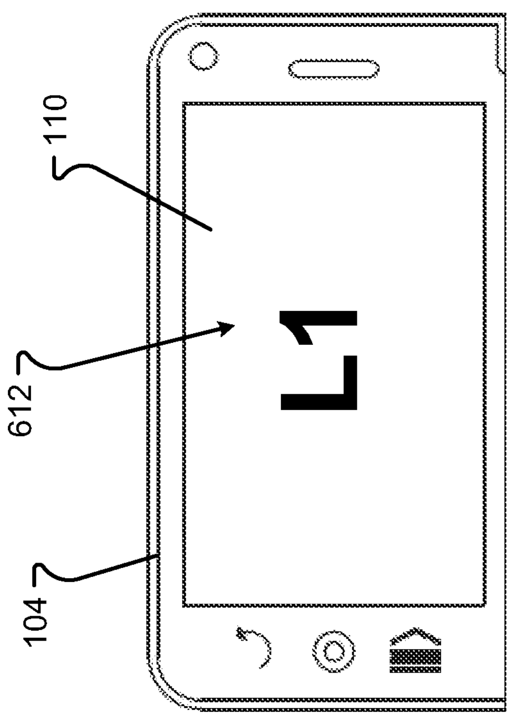


FIG. 6D

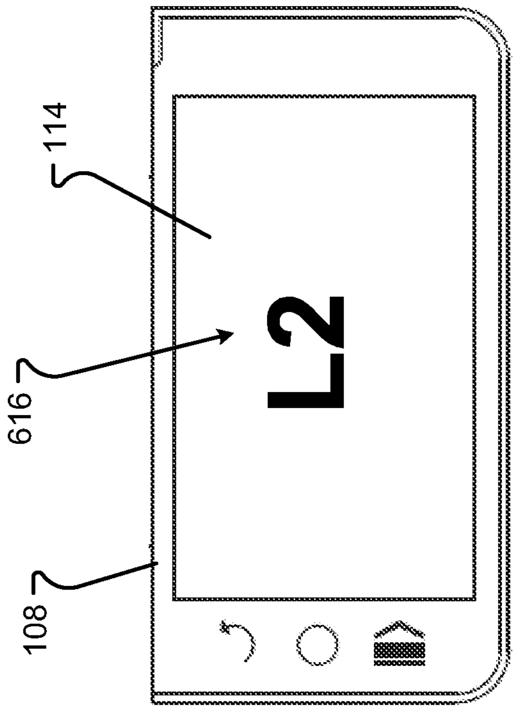


FIG. 6E

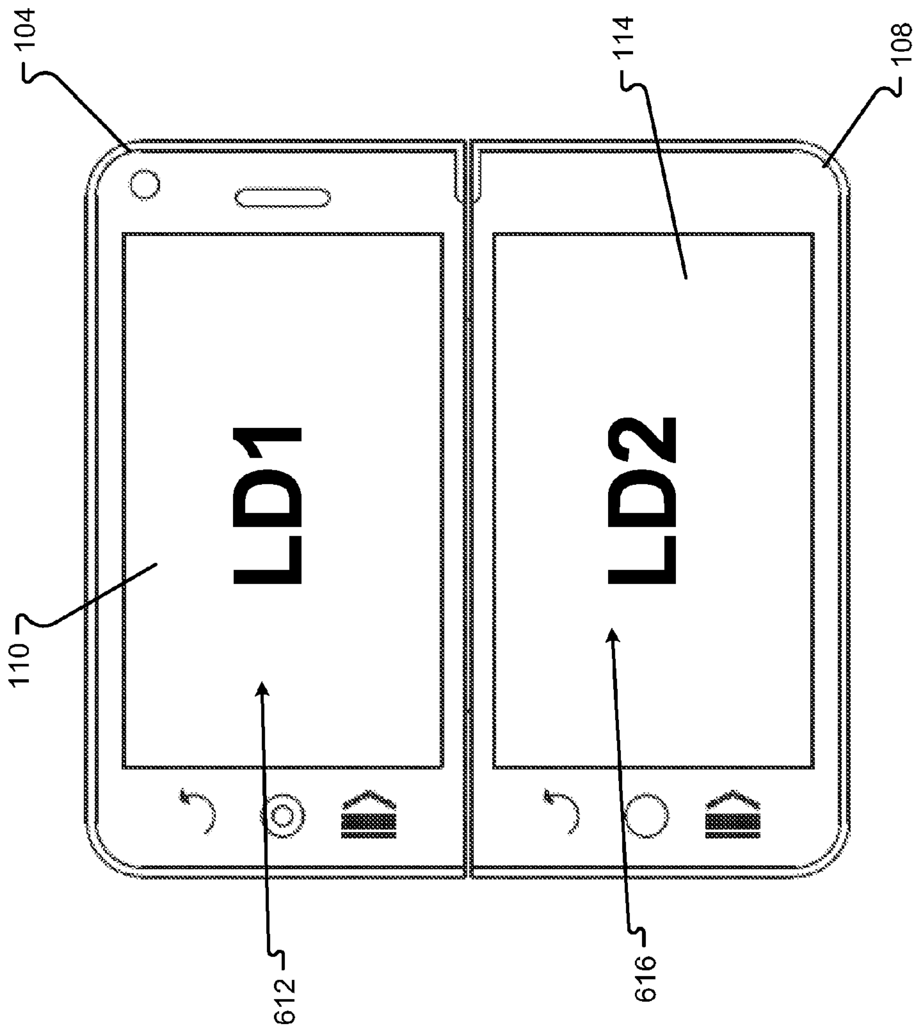


FIG. 6F

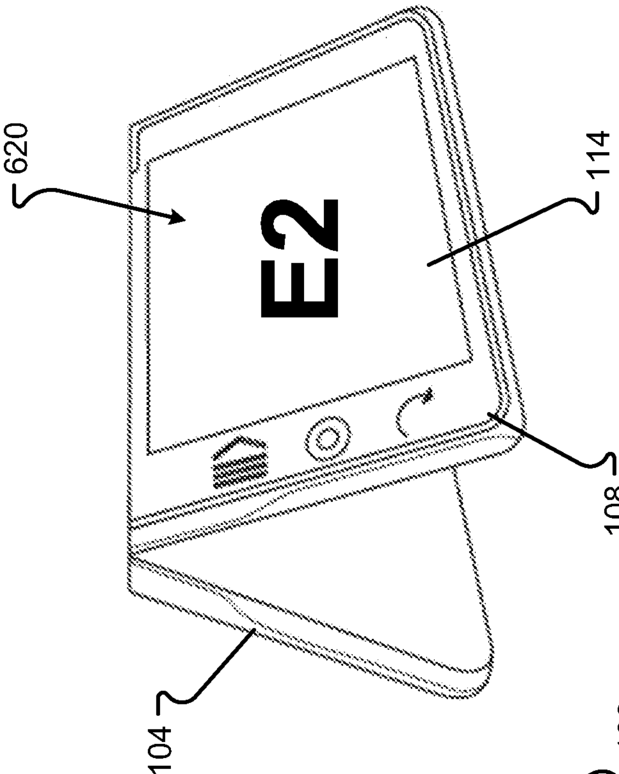


FIG. 6H

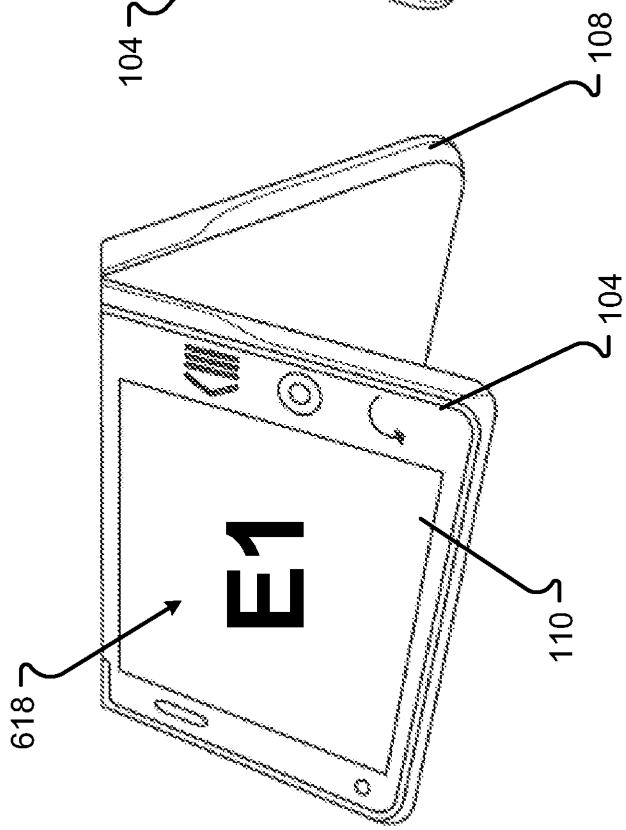


FIG. 6G

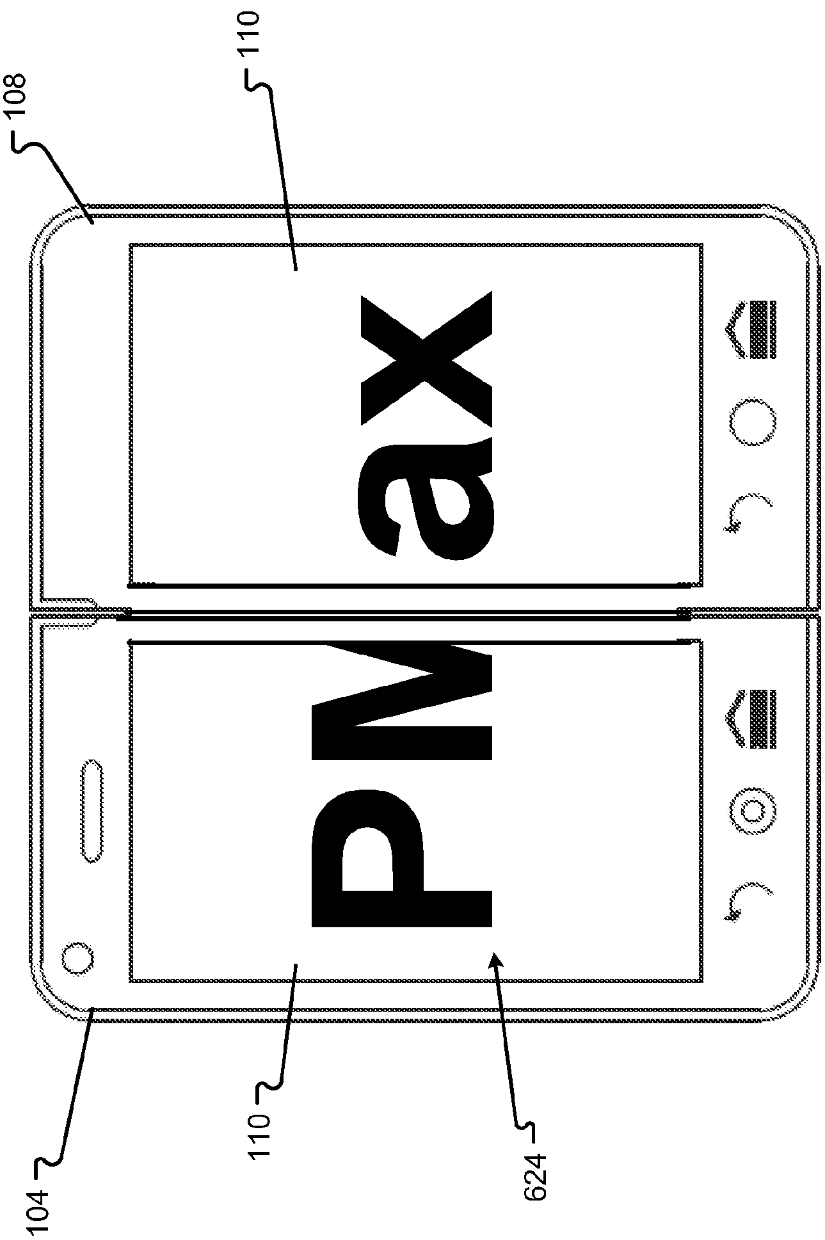


FIG. 6I

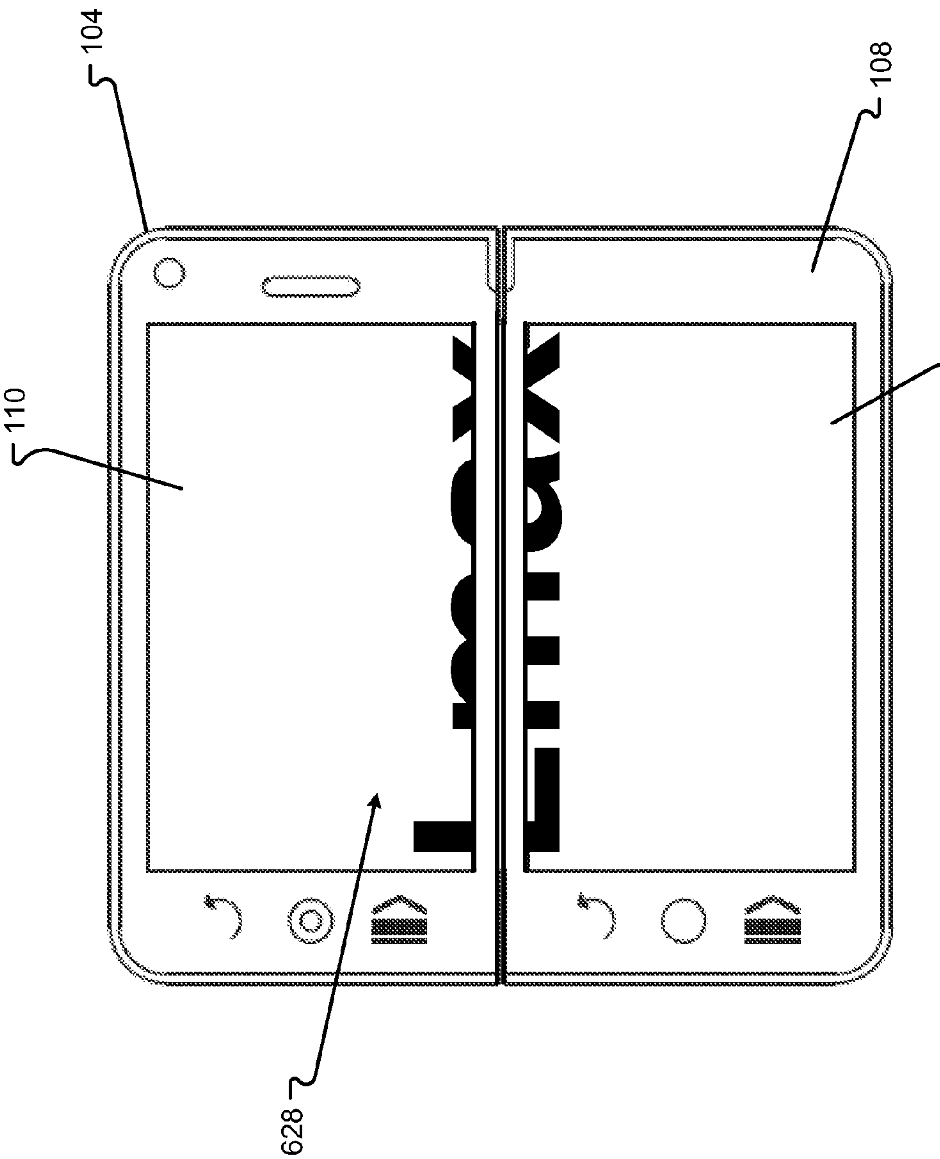


FIG. 6J

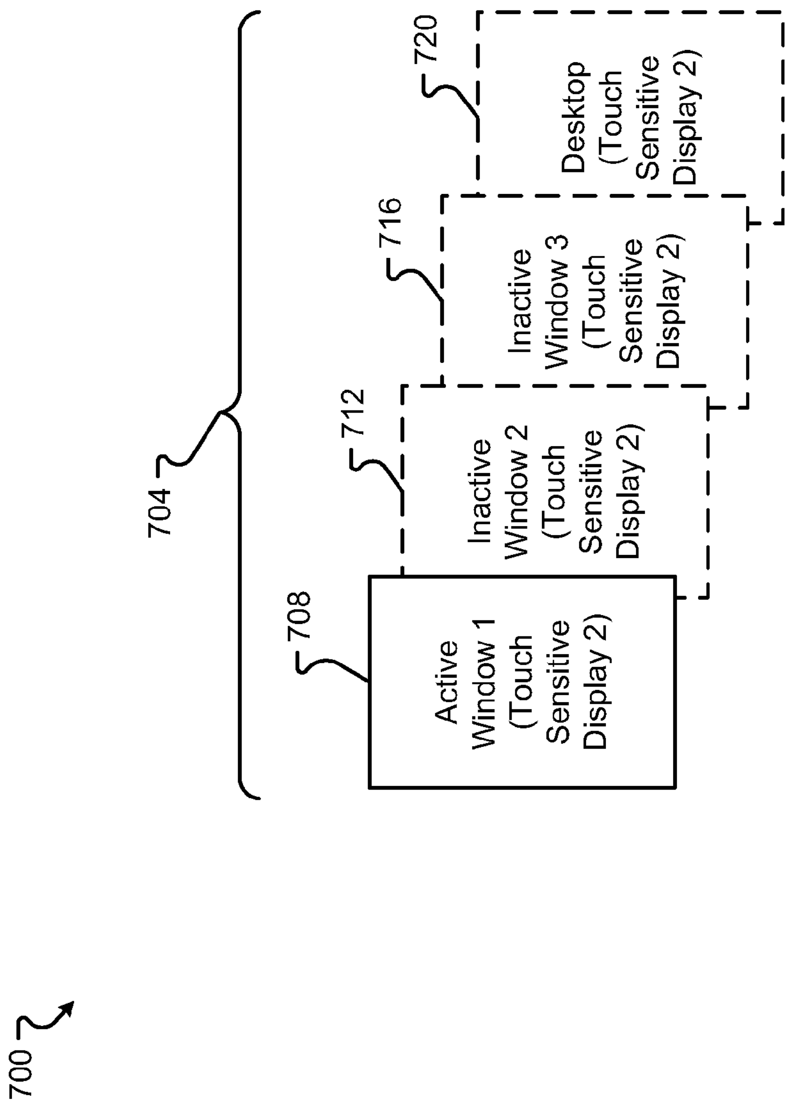

700
704
708
Active Window 1 (Touch Sensitive Display 2)
712
Inactive Window 2 (Touch Sensitive Display 2)
716
Inactive Window 3 (Touch Sensitive Display 2)
720
Desktop (Touch Sensitive Display 2)

FIG. 7A

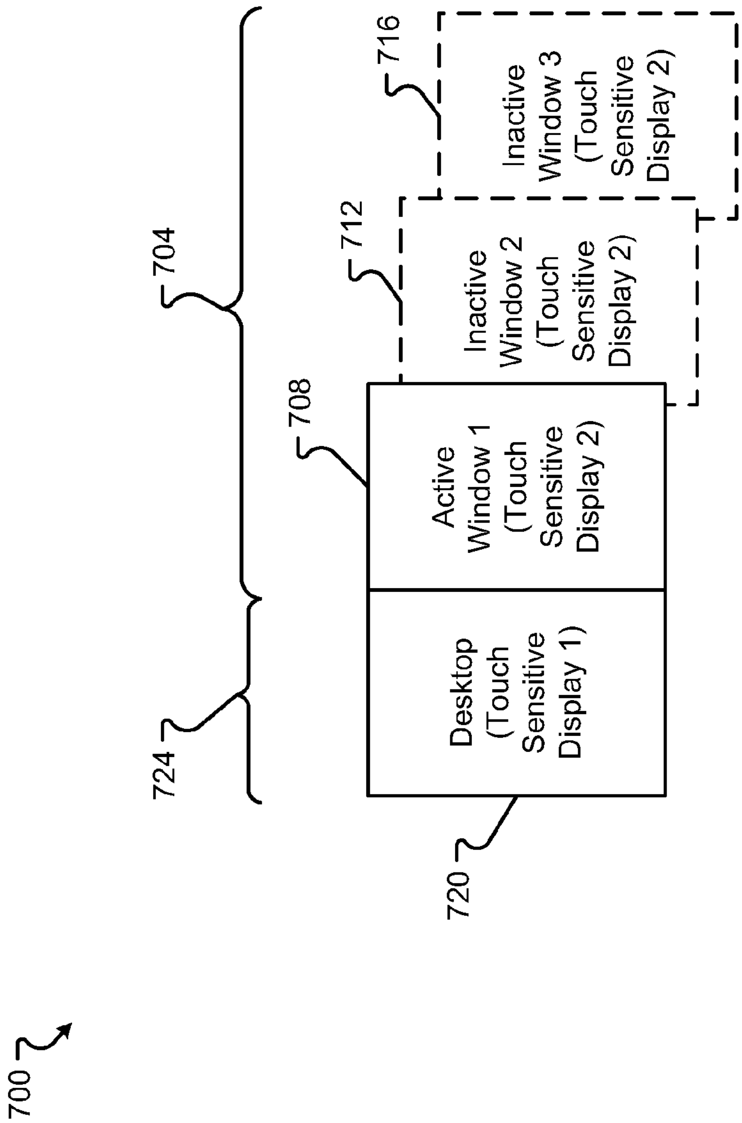

700
704
724
708
712
716
720
Active Window 1 (Touch Sensitive Display 2)
Inactive Window 2 (Touch Sensitive Display 2)
Inactive Window 3 (Touch Sensitive Display 2)
Desktop (Touch Sensitive Display 1)

FIG. 7B

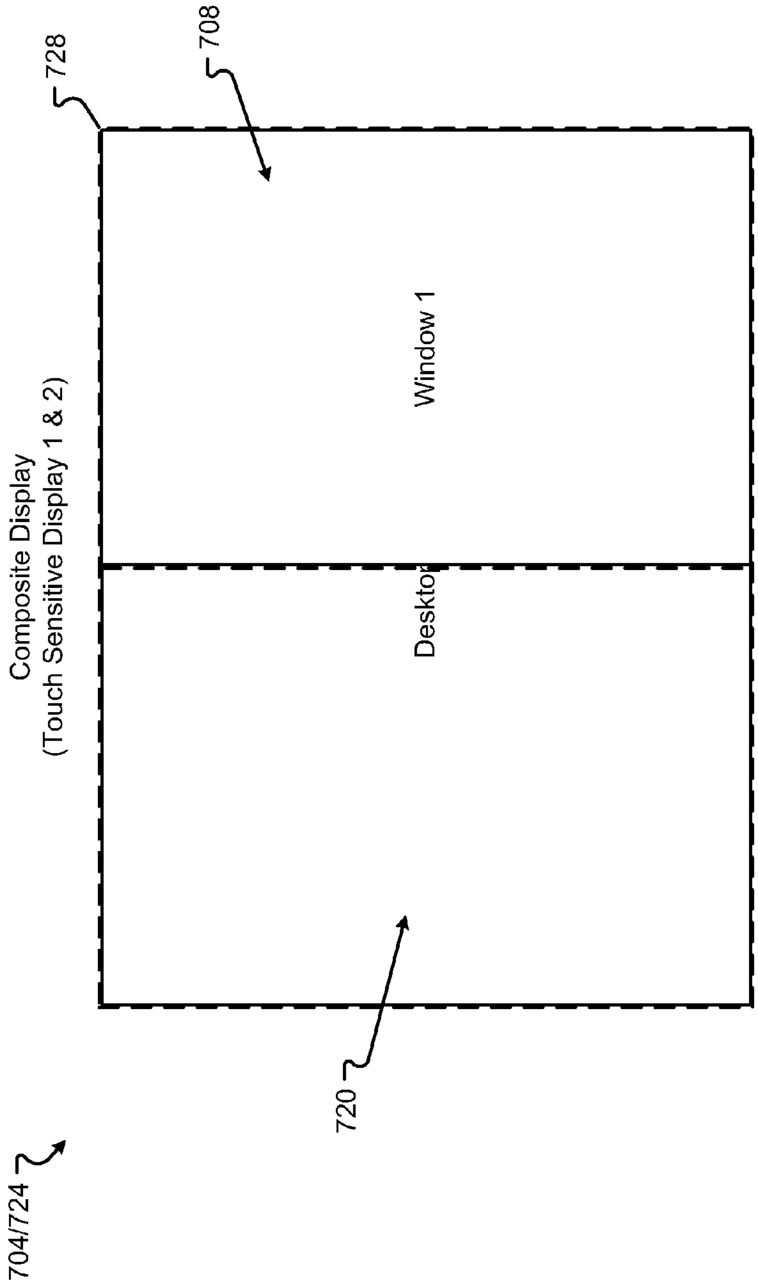

704/724
728
708
Composite Display
(Touch Sensitive Display 1 & 2)
Window 1
Deskto
720

FIG. 7C

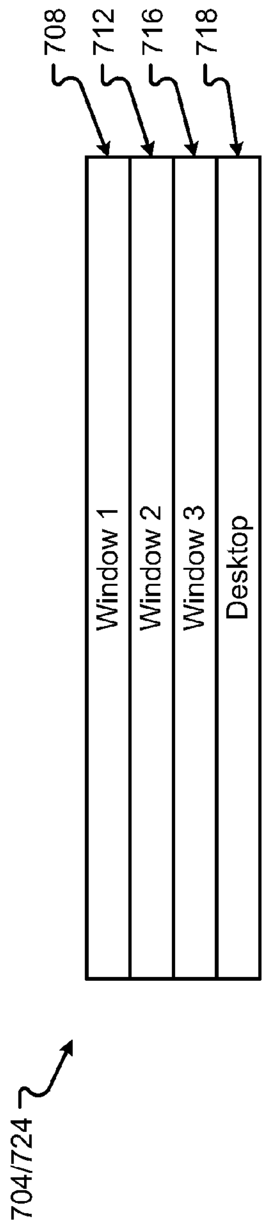

708
712
716
718
Window 1
Window 2
Window 3
Desktop
704/724

FIG. 7D

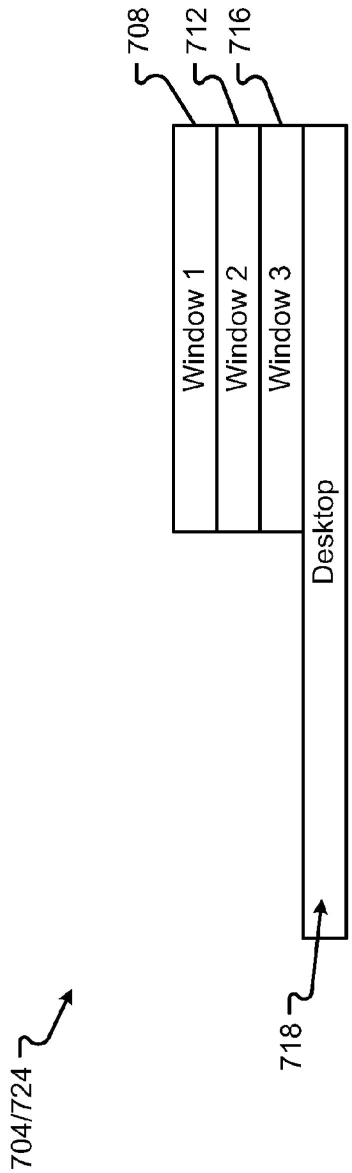

708
712
716
Window 1
Window 2
Window 3
Desktop
718
704/724

FIG. 7E

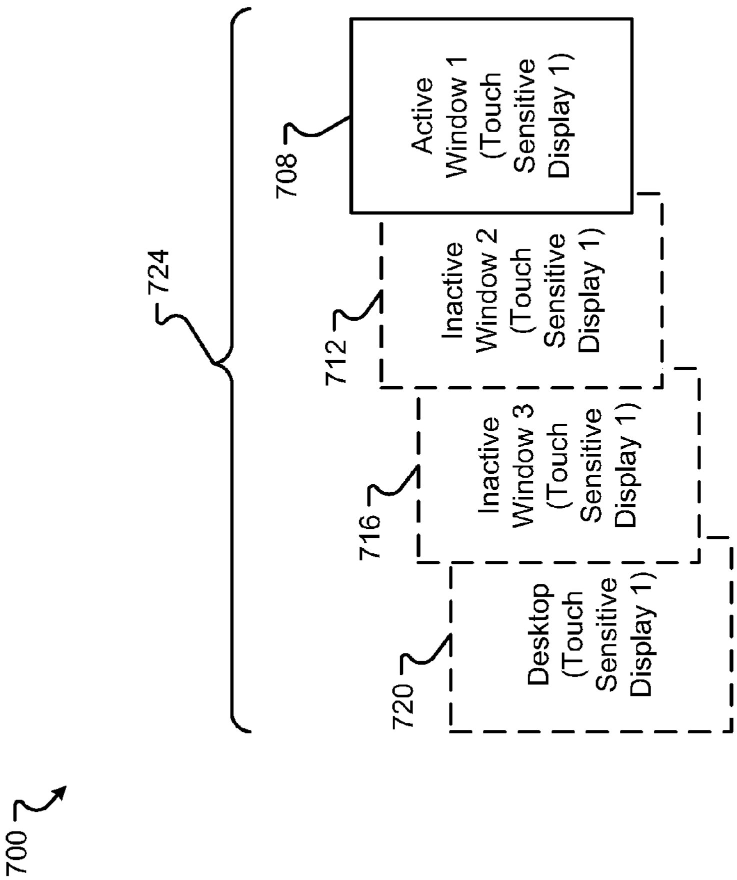

700
724
708
Active Window 1 (Touch Sensitive Display 1)
712
Inactive Window 2 (Touch Sensitive Display 1)
716
Inactive Window 3 (Touch Sensitive Display 1)
720
Desktop (Touch Sensitive Display 1)

FIG. 7F

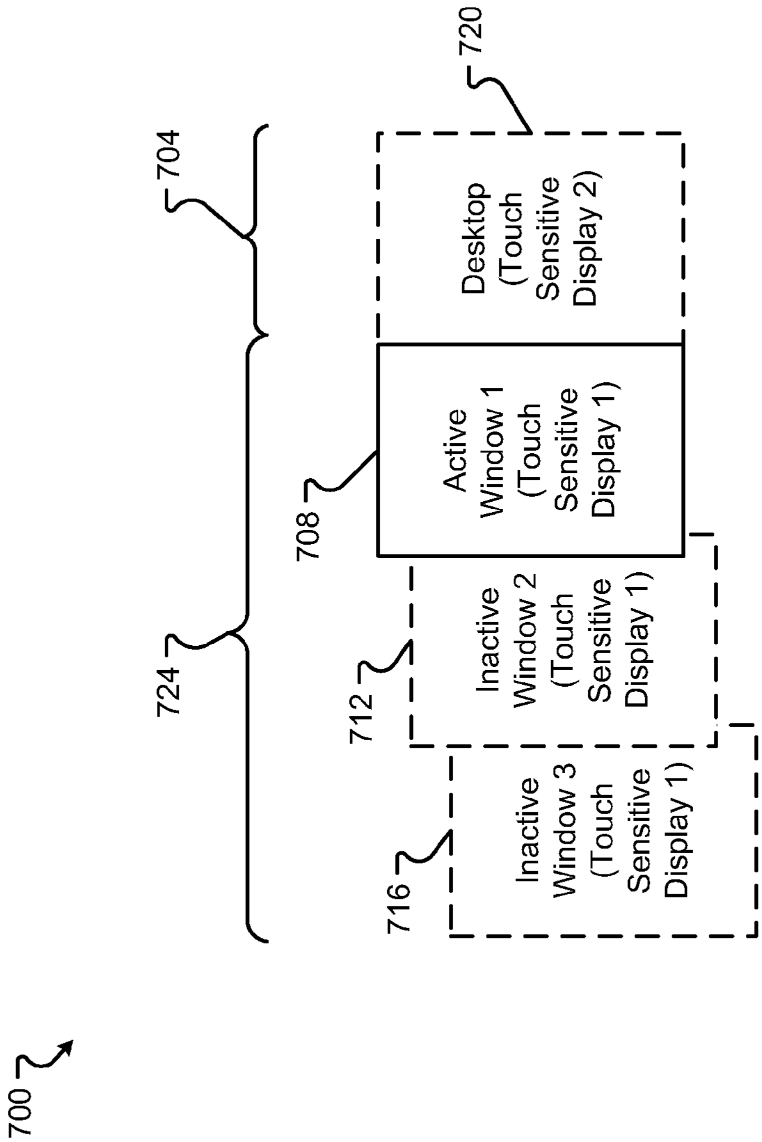

700
704
720
Desktop (Touch Sensitive Display 2)
708
Active Window 1 (Touch Sensitive Display 1)
724
712
Inactive Window 2 (Touch Sensitive Display 1)
716
Inactive Window 3 (Touch Sensitive Display 1)

FIG. 7G

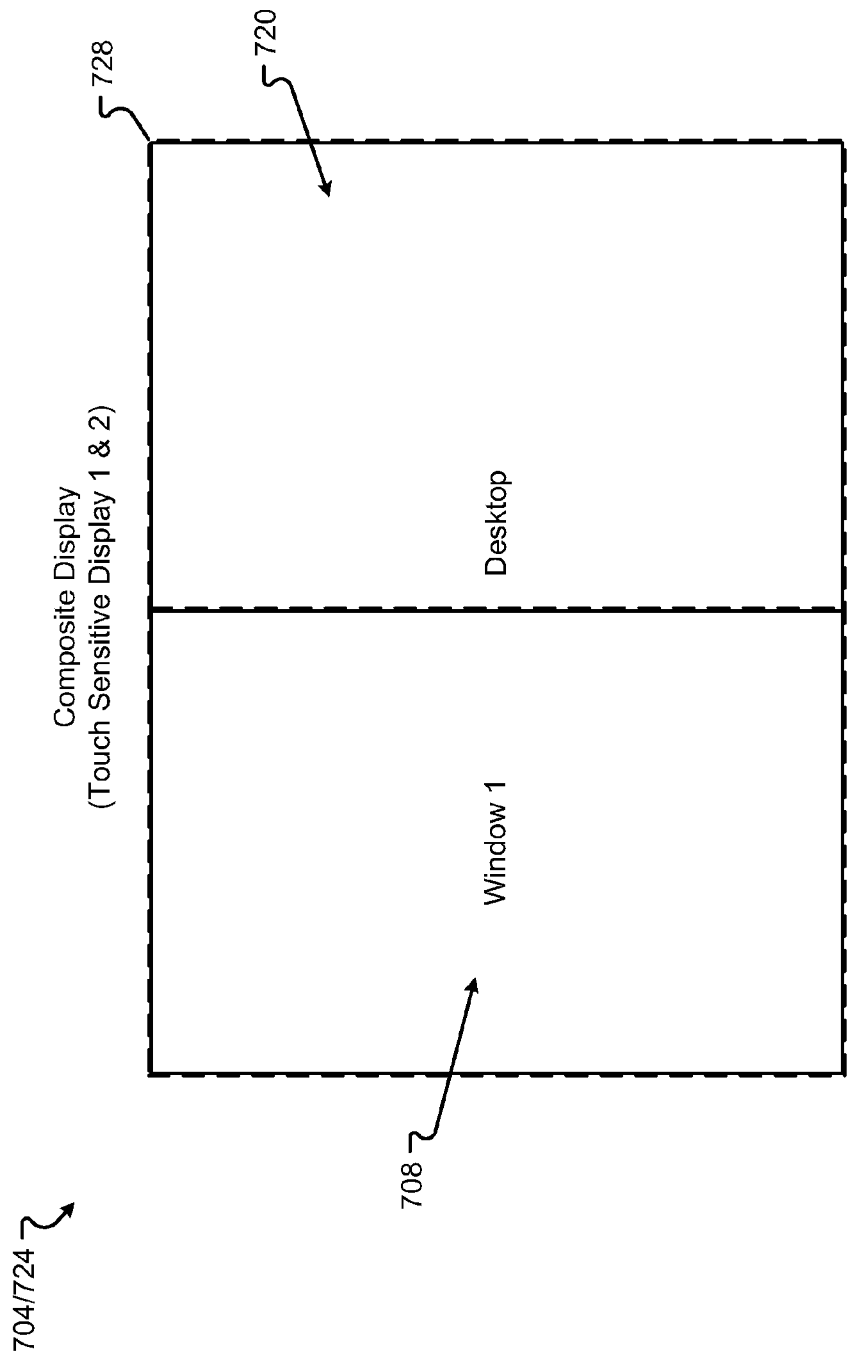
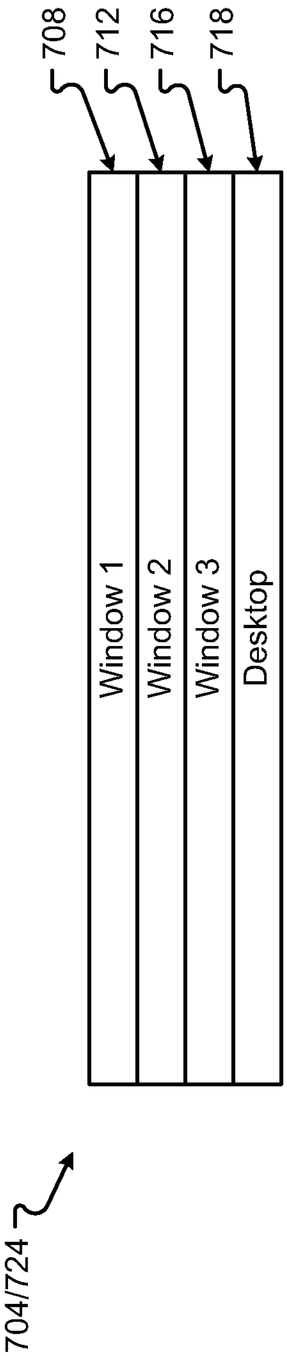
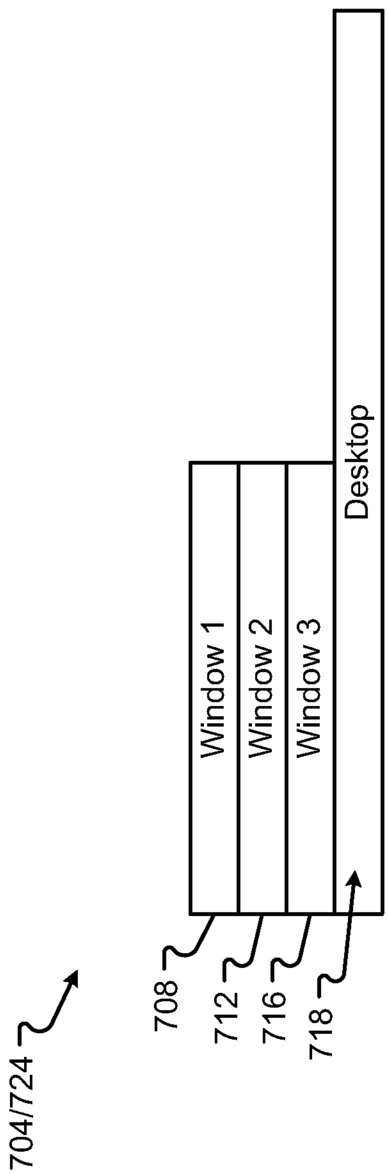

728
720
Composite Display
(Touch Sensitive Display 1 & 2)
Desktop
Window 1
708
704/724

FIG. 7H



708
712
716
718
Window 1
Window 2
Window 3
Desktop
704/724

FIG. 7I



Window 1
Window 2
Window 3
Desktop
708
712
716
718
704/724

FIG. 7J

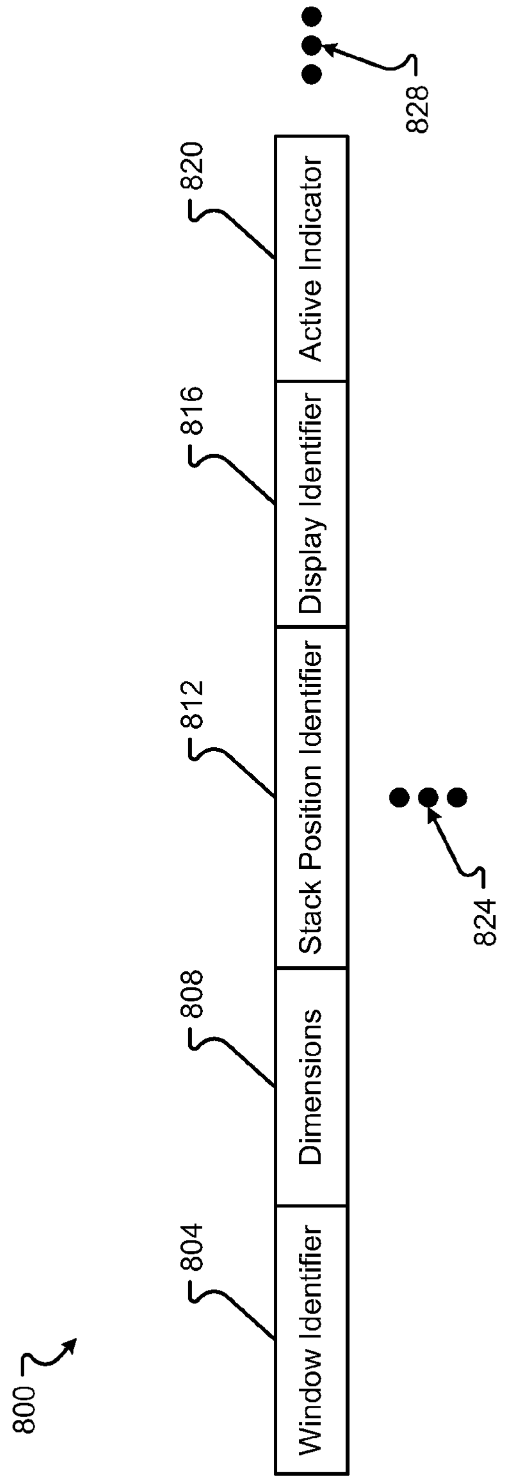

800
Window Identifier
804
Dimensions
808
Stack Position Identifier
812
Display Identifier
816
Active Indicator
820
824
828

FIG. 8

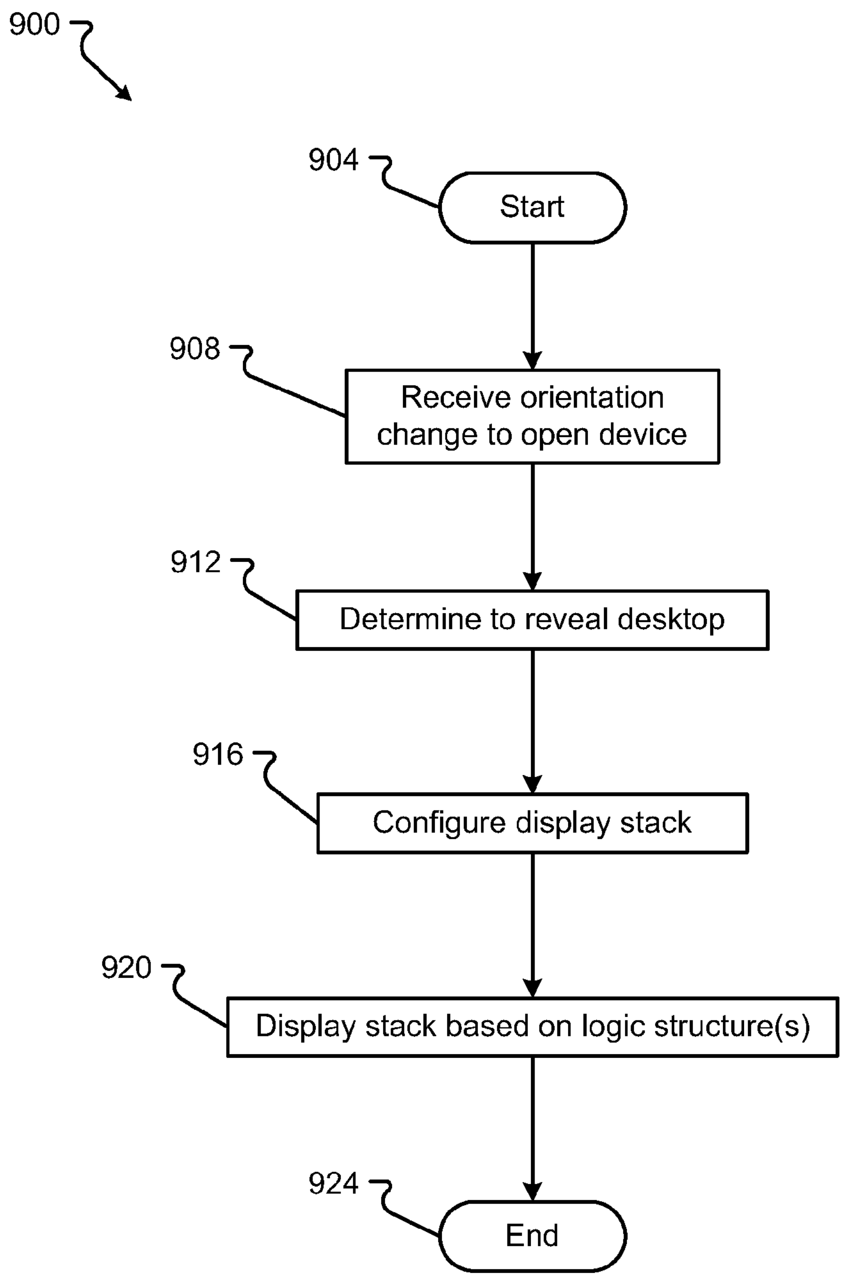

900
904
Start
908
Receive orientation change to open device
912
Determine to reveal desktop
916
Configure display stack
920
Display stack based on logic structure(s)
924
End

FIG. 9

DISPLAYING THE DESKTOP UPON DEVICE OPEN

CROSS REFERENCE TO RELATED APPLICATION

[0001] The present application is a continuation of and claims priority to U.S. patent application Ser. No. 13/248,138, filed Sep. 29, 2011, entitled "DISPLAYING THE DESKTOP UPON DEVICE OPEN," which claims the benefits of and priority, under 35 U.S.C. §119(e), to U.S. Provisional Application Ser. No. 61/389,000, filed Oct. 1, 2010, entitled "DUAL DISPLAY WINDOWING SYSTEM;" 61/389,117, filed Oct. 1, 2010, entitled "MULTI-OPERATING SYSTEM PORTABLE DOCKETING DEVICE;" 61/389,087, filed Oct. 1, 2010, entitled "TABLET COMPUTING USER INTERFACE;"61/458,150, filed Nov. 17, 2010, entitled "Dual Screen Email Client;" 61/539,884, filed Sep. 27, 2011, entitled "MOBILE DEVICE." Each of the aforementioned documents is incorporated herein by this reference in their entirety for all that they teach and for all purposes.

BACKGROUND

[0002] A substantial number of handheld computing devices, such as cellular phones, tablets, and E-Readers, make use of a touch screen display not only to deliver display information to the user but also to receive inputs from user interface commands. While touch screen displays may increase the configurability of the handheld device and provide a wide variety of user interface options, this flexibility typically comes at a price. The dual use of the touch screen to provide content and receive user commands, while flexible for the user, may obfuscate the display and cause visual clutter, thereby leading to user frustration and loss of productivity.

[0003] The small form factor of handheld computing devices requires a careful balancing between the displayed graphics and the area provided for receiving inputs. On the one hand, the small display constrains the display space, which may increase the difficulty of interpreting actions or results. On the other, a virtual keypad or other user interface scheme is superimposed on or positioned adjacent to an executing application, requiring the application to be squeezed into an even smaller portion of the display.

[0004] This balancing act is particularly difficult for single display touch screen devices. Single display touch screen devices are crippled by their limited screen space. When users are entering information into the device, through the single display, the ability to interpret information in the display can be severely hampered, particularly when a complex interaction between display and interface is required.

SUMMARY

[0005] There is a need for a dual multi-display handheld computing device that provides for enhanced power and/or versatility compared to conventional single display handheld computing devices. These and other needs are addressed by the various aspects, embodiments, and/or configurations of the present disclosure. Also, while the disclosure is presented in terms of exemplary embodiments, it should be appreciated that individual aspects of the disclosure can be separately claimed.

[0006] In an embodiment, a computer readable medium includes instructions causing a processor to manage a window stack for a multi-screen device. The computer-executable instructions comprise instructions to receive an orientation change in the multi-screen device, wherein the orientation change causes the multi-screen device to transition from a closed state to an open state, wherein, in the closed state, a first display is active and, in the open state, both the first display and a second display are active; instructions to determine if a desktop should be displayed on the second display; and, after determining the desktop should be displayed on the second display, instructions to display the desktop on the second display and a first window on the first display.

[0007] In another embodiment, a device comprises at least two displays, a memory, and a processor in communication with the memory and each of the at least two displays. The device is operable to provide a composite display, wherein the composite display comprises one touch sensitive displays in a closed state; receive an orientation change in the multi-screen device, wherein the orientation change causes the multi-screen device to transition from a closed state to an open state, wherein, in the open state, the composite display comprises a first portion associated with the first touch sensitive display and a second portion associated with a second touch sensitive display; expand a desktop to cover the composite display; and determine a first window is displayed in a first portion of the composite display and the desktop is displayed in a second portion of the composite display; and display the desktop on the second touch sensitive display and the first window on the first touch sensitive display.

[0008] In still another embodiment, a method for presenting a display for a multi-screen device comprises providing a composite display that spans at least a portion of a first touch sensitive display when the multi-screen device is in a closed state; displaying a first window at a top of a window stack when the device is in the closed state; receiving an orientation change for the multi-screen device, wherein the orientation change is a transition from the closed state to an open state; changing the composite display to span at least a portion of a first touch sensitive display and a second touch sensitive display of the multi-screen device, wherein a first portion of the composite display is associated with the first touch sensitive display and a second portion of the composite display is associated with the second touch sensitive display; determining that a desktop is associated with the composite display; modifying the desktop to expand over the composite display; determining the first window is at the top of the window stack in the first portion of the composite display; determining the desktop is at the top of the window stack in the second portion of the composite display; displaying the first window on the first touch sensitive display; and displaying the desktop on the second touch sensitive display.

[0009] The present disclosure can provide a number of advantages depending on the particular aspect, embodiment, and/or configuration. The window stack arrangement provides the advantage of maintain a logical arrangement of windows that is easily understood by the user. The "deck of cards" arrangement allows the user to quickly navigate active and inactive windows between the two displays. These and other advantages will be apparent from the disclosure.

[0010] The phrases "at least one", "one or more", and "and/or" are open-ended expressions that are both conjunctive and disjunctive in operation. For example, each of the expressions "at least one of A, B and C", "at least one of A, B, or C", "one or more of A, B, and C", "one or more of A, B, or C" and "A, B, and/or C" means A alone, B alone, C alone, A and B together, A and C together, B and C together, or A, B and C together.

[0011] The term "a" or "an" entity refers to one or more of that entity. As such, the terms "a" (or "an"), "one or more" and "at least one" can be used interchangeably herein. It is also to be noted that the terms "comprising", "including", and "having" can be used interchangeably.

[0012] The term "automatic" and variations thereof, as used herein, refers to any process or operation done without material human input when the process or operation is performed. However, a process or operation can be automatic, even though performance of the process or operation uses material or immaterial human input, if the input is received before performance of the process or operation. Human input is deemed to be material if such input influences how the process or operation will be performed. Human input that consents to the performance of the process or operation is not deemed to be "material".

[0013] The term "computer-readable medium" as used herein refers to any tangible storage and/or transmission medium that participate in providing instructions to a processor for execution. Such a medium may take many forms, including but not limited to, non-volatile media, volatile media, and transmission media. Non-volatile media includes, for example, NVRAM, or magnetic or optical disks. Volatile media includes dynamic memory, such as main memory. Common forms of computer-readable media include, for example, a floppy disk, a flexible disk, hard disk, magnetic tape, or any other magnetic medium, magneto-optical medium, a CD-ROM, any other optical medium, punch cards, paper tape, any other physical medium with patterns of holes, a RAM, a PROM, and EPROM, a FLASH-EPROM, a solid state medium like a memory card, any other memory chip or cartridge, a carrier wave as described hereinafter, or any other medium from which a computer can read. A digital file attachment to e-mail or other self-contained information archive or set of archives is considered a distribution medium equivalent to a tangible storage medium. When the computer-readable media is configured as a database, it is to be understood that the database may be any type of database, such as relational, hierarchical, object-oriented, and/or the like. Accordingly, the disclosure is considered to include a tangible storage medium or distribution medium and prior art-recognized equivalents and successor media, in which the software implementations of the present disclosure are stored.

[0014] The term "desktop" refers to a metaphor used to portray systems. A desktop is generally considered a "surface" that typically includes pictures, called icons, widgets, folders, etc. that can activate show applications, windows, cabinets, files, folders, documents, and other graphical items. The icons are generally selectable to initiate a task through user interface interaction to allow a user to execute applications or conduct other operations.

[0015] The term "screen," "touch screen," or "touchscreen" refers to a physical structure that includes one or more hardware components that provide the device with the ability to render a user interface and/or receive user input. A screen can encompass any combination of gesture capture region, a touch sensitive display, and/or a configurable area. The device can have one or more physical screens embedded in the hardware. However a screen may also include an external peripheral device that may be attached and detached from the device. In embodiments, multiple external devices may be attached to the device. Thus, in embodiments, the screen can enable the user to interact with the device by touching areas on the screen and provides information to a user through a display. The touch screen may sense user contact in a number of different ways, such as by a change in an electrical parameter (e.g., resistance or capacitance), acoustic wave variations, infrared radiation proximity detection, light variation detection, and the like. In a resistive touch screen, for example, normally separated conductive and resistive metallic layers in the screen pass an electrical current. When a user touches the screen, the two layers make contact in the contacted location, whereby a change in electrical field is noted and the coordinates of the contacted location calculated. In a capacitive touch screen, a capacitive layer stores electrical charge, which is discharged to the user upon contact with the touch screen, causing a decrease in the charge of the capacitive layer. The decrease is measured, and the contacted location coordinates determined. In a surface acoustic wave touch screen, an acoustic wave is transmitted through the screen, and the acoustic wave is disturbed by user contact. A receiving transducer detects the user contact instance and determines the contacted location coordinates.

[0016] The term "display" refers to a portion of one or more screens used to display the output of a computer to a user. A display may be a single-screen display or a multi-screen display, referred to as a composite display. A composite display can encompass the touch sensitive display of one or more screens. A single physical screen can include multiple displays that are managed as separate logical displays. Thus, different content can be displayed on the separate displays although part of the same physical screen.

[0017] The term "displayed image" refers to an image produced on the display. A typical displayed image is a window or desktop. The displayed image may occupy all or a portion of the display.

[0018] The term "display orientation" refers to the way in which a rectangular display is oriented by a user for viewing. The two most common types of display orientation are portrait and landscape. In landscape mode, the display is oriented such that the width of the display is greater than the height of the display (such as a 4:3 ratio, which is 4 units wide and 3 units tall, or a 16:9 ratio, which is 16 units wide and 9 units tall). Stated differently, the longer dimension of the display is oriented substantially horizontal in landscape mode while the shorter dimension of the display is oriented substantially vertical. In the portrait mode, by contrast, the display is oriented such that the width of the display is less than the height of the display. Stated differently, the shorter dimension of the display is oriented substantially horizontal in the portrait mode while the longer dimension of the display is oriented substantially vertical.

[0019] The term "composited display" refers to a logical structure that defines a display that can encompass one or more screens. A multi-screen display can be associated with a composite display that encompasses all the screens. The composite display can have different display characteristics based on the various orientations of the device.

[0020] The term "gesture" refers to a user action that expresses an intended idea, action, meaning, result, and/or outcome. The user action can include manipulating a device (e.g., opening or closing a device, changing a device orientation, moving a trackball or wheel, etc.), movement of a body part in relation to the device, movement of an implement or tool in relation to the device, audio inputs, etc. A gesture may be made on a device (such as on the screen) or with the device to interact with the device.

[0021] The term "module" as used herein refers to any known or later developed hardware, software, firmware, artificial intelligence, fuzzy logic, or combination of hardware and software that is capable of performing the functionality associated with that element.

[0022] The term "gesture capture" refers to a sense or otherwise a detection of an instance and/or type of user gesture. The gesture capture can occur in one or more areas of the screen, A gesture region can be on the display, where it may be referred to as a touch sensitive display or off the display where it may be referred to as a gesture capture area.

[0023] A "multi-screen application" refers to an application that is capable of multiple modes. The multi-screen application mode can include, but is not limited to, a single screen mode (where the application is displayed on a single screen) or a composite display mode (where the application is displayed on two or more screens). A multi-screen application can have different layouts optimized for the mode. Thus, the multi-screen application can have different layouts for a single screen or for a composite display that can encompass two or more screens. The different layouts may have different screen/display dimensions and/or configurations on which the user interfaces of the multi-screen applications can be rendered. The different layouts allow the application to optimize the application's user interface for the type of display, e.g., single screen or multiple screens. In single screen mode, the multi-screen application may present one window pane of information. In a composite display mode, the multi-screen application may present multiple window panes of information or may provide a larger and a richer presentation because there is more space for the display contents. The multi-screen applications may be designed to adapt dynamically to changes in the device and the mode depending on which display (single or composite) the system assigns to the multi-screen application. In alternative embodiments, the user can use a gesture to request the application transition to a different mode, and, if a display is available for the requested mode, the device can allow the application to move to that display and transition modes.

[0024] A "single-screen application" refers to an application that is capable of single screen mode. Thus, the single-screen application can produce only one window and may not be capable of different modes or different display dimensions. A single-screen application is incapable of the several modes discussed with the multi-screen application.

[0025] The term "window" refers to a, typically rectangular, displayed image on at least part of a display that contains or provides content different from the rest of the screen. The window may obscure the desktop.

[0026] The terms "determine", "calculate" and "compute," and variations thereof, as used herein, are used interchangeably and include any type of methodology, process, mathematical operation or technique.

[0027] It shall be understood that the term "means" as used herein shall be given its broadest possible interpretation in accordance with 35 U.S.C., Section **112**, Paragraph **6**. Accordingly, a claim incorporating the term "means" shall cover all structures, materials, or acts set forth herein, and all of the equivalents thereof. Further, the structures, materials or acts and the equivalents thereof shall include all those described in the summary of the invention, brief description of the drawings, detailed description, abstract, and claims themselves.

[0028] The preceding is a simplified summary of the disclosure to provide an understanding of some aspects of the disclosure. This summary is neither an extensive nor exhaustive overview of the disclosure and its various aspects, embodiments, and/or configurations. It is intended neither to identify key or critical elements of the disclosure nor to delineate the scope of the disclosure but to present selected concepts of the disclosure in a simplified form as an introduction to the more detailed description presented below. As will be appreciated, other aspects, embodiments, and/or configurations of the disclosure are possible utilizing, alone or in combination, one or more of the features set forth above or described in detail below.

BRIEF DESCRIPTION OF THE DRAWINGS

[0029] FIG. **1**A includes a first view of an embodiment of a multi-screen user device;

[0030] FIG. **1**B includes a second view of an embodiment of a multi-screen user device;

[0031] FIG. **1**C includes a third view of an embodiment of a multi-screen user device;

[0032] FIG. **1**D includes a fourth view of an embodiment of a multi-screen user device;

[0033] FIG. **1**E includes a fifth view of an embodiment of a multi-screen user device;

[0034] FIG. **1**F includes a sixth view of an embodiment of a multi-screen user device;

[0035] FIG. **1**G includes a seventh view of an embodiment of a multi-screen user device;

[0036] FIG. **1**H includes a eighth view of an embodiment of a multi-screen user device;

[0037] FIG. **1**I includes a ninth view of an embodiment of a multi-screen user device;

[0038] FIG. **1**J includes a tenth view of an embodiment of a multi-screen user device;

[0039] FIG. **2** is a block diagram of an embodiment of the hardware of the device;

[0040] FIG. **3**A is a block diagram of an embodiment of the state model for the device based on the device's orientation and/or configuration;

[0041] FIG. **3**B is a table of an embodiment of the state model for the device based on the device's orientation and/or configuration;

[0042] FIG. **4**A is a first representation of an embodiment of user gesture received at a device;

[0043] FIG. **4**B is a second representation of an embodiment of user gesture received at a device;

[0044] FIG. **4**C is a third representation of an embodiment of user gesture received at a device;

[0045] FIG. **4**D is a fourth representation of an embodiment of user gesture received at a device;

[0046] FIG. **4**E is a fifth representation of an embodiment of user gesture received at a device;

[0047] FIG. **4**F is a sixth representation of an embodiment of user gesture received at a device;

[0048] FIG. **4**G is a seventh representation of an embodiment of user gesture received at a device;

[0049] FIG. **4**H is a eighth representation of an embodiment of user gesture received at a device;

[0050] FIG. **5**A is a block diagram of an embodiment of the device software and/or firmware;

[0051] FIG. **5**B is a second block diagram of an embodiment of the device software and/or firmware;

[0052] FIG. **6**A is a first representation of an embodiment of a device configuration generated in response to the device state;

[0053] FIG. **6**B is a second representation of an embodiment of a device configuration generated in response to the device state;

[0054] FIG. **6**C is a third representation of an embodiment of a device configuration generated in response to the device state;

[0055] FIG. **6**D is a fourth representation of an embodiment of a device configuration generated in response to the device state;

[0056] FIG. **6**E is a fifth representation of an embodiment of a device configuration generated in response to the device state;

[0057] FIG. **6**F is a sixth representation of an embodiment of a device configuration generated in response to the device state;

[0058] FIG. **6**G is a seventh representation of an embodiment of a device configuration generated in response to the device state;

[0059] FIG. **6**H is a eighth representation of an embodiment of a device configuration generated in response to the device state;

[0060] FIG. **6**I is a ninth representation of an embodiment of a device configuration generated in response to the device state;

[0061] FIG. **6**J is a tenth representation of an embodiment of a device configuration generated in response to the device state;

[0062] FIG. **7**A is a representation of a logical window stack;

[0063] FIG. **7**B is another representation of an embodiment of a logical window stack;

[0064] FIG. **7**C is another representation of an embodiment of a logical window stack;

[0065] FIG. **7**D is another representation of an embodiment of a logical window stack;

[0066] FIG. **7**E is another representation of an embodiment of a logical window stack;

[0067] FIG. **7**F is another representation of an embodiment of a logical window stack;

[0068] FIG. **7**G is another representation of an embodiment of a logical window stack;

[0069] FIG. **7**H is another representation of an embodiment of a logical window stack;

[0070] FIG. **7**I is another representation of an embodiment of a logical window stack;

[0071] FIG. **7**J is another representation of an embodiment of a logical window stack;

[0072] FIG. **8** is block diagram of an embodiment of a logical data structure for a window stack;

[0073] FIG. **9** is a flow chart of an embodiment of a method for changing a display based on a window stack;

[0074] In the appended figures, similar components and/or features may have the same reference label. Further, various components of the same type may be distinguished by following the reference label by a letter that distinguishes among the similar components. If only the first reference label is used in the specification, the description is applicable to any one of the similar components having the same first reference label irrespective of the second reference label.

DETAILED DESCRIPTION

[0075] Presented herein are embodiments of a device. The device can be a communications device, such as a cellular telephone, or other smart device. The device can include two screens that are oriented to provide several unique display configurations. Further, the device can receive user input in unique ways. The overall design and functionality of the device provides for an enhanced user experience making the device more useful and more efficient.

[0076] Mechanical Features:

[0077] FIGS. **1**A-**1**J illustrate a device **100** in accordance with embodiments of the present disclosure. As described in greater detail below, device **100** can be positioned in a number of different ways each of which provides different functionality to a user. The device **100** is a multi-screen device that includes a primary screen **104** and a secondary screen **108**, both of which are touch sensitive. In embodiments, the entire front surface of screens **104** and **108** may be touch sensitive and capable of receiving input by a user touching the front surface of the screens **104** and **108**. Primary screen **104** includes touch sensitive display **110**, which, in addition to being touch sensitive, also displays information to a user. Secondary screen **108** includes touch sensitive display **114**, which also displays information to a user. In other embodiments, screens **104** and **108** may include more than one display area.

[0078] Primary screen **104** also includes a configurable area **112** that has been configured for specific inputs when the user touches portions of the configurable area **112**. Secondary screen **108** also includes a configurable area **116** that has been configured for specific inputs. Areas **112***a* and **116***a* have been configured to receive a "back" input indicating that a user would like to view information previously displayed. Areas **112***b* and **116***b* have been configured to receive a "menu" input indicating that the user would like to view options from a menu. Areas **112***c* and **116***c* have been configured to receive a "home" input indicating that the user would like to view information associated with a "home" view. In other embodiments, areas **112***a-c* and **116***a-c* may be configured, in addition to the configurations described above, for other types of specific inputs including controlling features of device **100**, some non-limiting examples including adjusting overall system power, adjusting the volume, adjusting the brightness, adjusting the vibration, selecting of displayed items (on either of screen **104** or **108**), operating a camera, operating a microphone, and initiating/terminating of telephone calls. Also, in some embodiments, areas **112***a*-C and **116***a*-C may be configured for specific inputs depending upon the application running on device **100** and/or information displayed on touch sensitive displays **110** and/or **114**.

[0079] In addition to touch sensing, primary screen **104** and secondary screen **108** may also include areas that receive input from a user without requiring the user to touch the display area of the screen. For example, primary screen **104** includes gesture capture area **120**, and secondary screen **108** includes gesture capture area **124**. These areas are able to receive input by recognizing gestures made by a user without the need for the user to actually touch the surface of the display area. In comparison to touch sensitive displays **110** and **114**, the gesture capture areas **120** and **124** are commonly not capable of rendering a displayed image.

[0080] The two screens **104** and **108** are connected together with a hinge **128**, shown clearly in FIG. **1**C (illustrating a back view of device **100**). Hinge **128**, in the embodiment shown in

FIGS. **1A-1J**, is a center hinge that connects screens **104** and **108** so that when the hinge is closed, screens **104** and **108** are juxtaposed (i.e., side-by-side) as shown in FIG. **1B** (illustrating a front view of device **100**). Hinge **128** can be opened to position the two screens **104** and **108** in different relative positions to each other. As described in greater detail below, the device **100** may have different functionalities depending on the relative positions of screens **104** and **108**.

[0081] FIG. **1D** illustrates the right side of device **100**. As shown in FIG. **1D**, secondary screen **108** also includes a card slot **132** and a port **136** on its side. Card slot **132** in embodiments, accommodates different types of cards including a subscriber identity module (SIM). Port **136** in embodiments is an input/output port (I/O port) that allows device **100** to be connected to other peripheral devices, such as a display, keyboard, or printing device. As can be appreciated, these are merely some examples and in other embodiments device **100** may include other slots and ports such as slots and ports for accommodating additional memory devices and/or for connecting other peripheral devices. Also shown in FIG. **1D** is an audio jack **140** that accommodates a tip, ring, sleeve (TRS) connector for example to allow a user to utilize headphones or a headset.

[0082] Device **100** also includes a number of buttons **158**. For example, FIG. **1E** illustrates the left side of device **100**. As shown in FIG. **1E**, the side of primary screen **104** includes three buttons **144**, **148**, and **152**, which can be configured for specific inputs. For example, buttons **144**, **148**, and **152** may be configured to, in combination or alone, control a number of aspects of device **100**. Some non-limiting examples include overall system power, volume, brightness, vibration, selection of displayed items (on either of screen **104** or **108**), a camera, a microphone, and initiation/termination of telephone calls. In some embodiments, instead of separate buttons two buttons may be combined into a rocker button. This arrangement is useful in situations where the buttons are configured to control features such as volume or brightness. In addition to buttons **144**, **148**, and **152**, device **100** also includes a button **156**, shown in FIG. **1F**, which illustrates the top of device **100**. In one embodiment, button **156** is configured as an on/off button used to control overall system power to device **100**. In other embodiments, button **156** is configured to, in addition to or in lieu of controlling system power, control other aspects of device **100**. In some embodiments, one or more of the buttons **144**, **148**, **152**, and **156** are capable of supporting different user commands. By way of example, a normal press has a duration commonly of less than about 1 second and resembles a quick tap. A medium press has a duration commonly of 1 second or more but less than about 12 seconds. A long press has a duration commonly of about 12 seconds or more. The function of the buttons is normally specific to the application that is currently in focus on the respective display **110** and **114**. In a telephone application for instance and depending on the particular button, a normal, medium, or long press can mean end call, increase in call volume, decrease in call volume, and toggle microphone mute. In a camera or video application for instance and depending on the particular button, a normal, medium, or long press can mean increase zoom, decrease zoom, and take photograph or record video.

[0083] There are also a number of hardware components within device **100**. As illustrated in FIG. **1C**, device **100** includes a speaker **160** and a microphone **164**. Device **100** also includes a camera **168** (FIG. **1B**). Additionally, device **100** includes two position sensors **172A** and **172B**, which are used to determine the relative positions of screens **104** and **108**. In one embodiment, position sensors **172A** and **172B** are Hall effect sensors. However, in other embodiments other sensors can be used in addition to or in lieu of the Hall effect sensors. An accelerometer **176** may also be included as part of device **100** to determine the orientation of the device **100** and/or the orientation of screens **104** and **108**. Additional internal hardware components that may be included in device **100** are described below with respect to FIG. **2**.

[0084] The overall design of device **100** allows it to provide additional functionality not available in other communication devices. Some of the functionality is based on the various positions and orientations that device **100** can have. As shown in FIGS. **1B-1G**, device **100** can be operated in an "open" position where screens **104** and **108** are juxtaposed. This position allows a large display area for displaying information to a user. When position sensors **172A** and **172B** determine that device **100** is in the open position, they can generate a signal that can be used to trigger different events such as displaying information on both screens **104** and **108**. Additional events may be triggered if accelerometer **176** determines that device **100** is in a portrait position (FIG. **1B**) as opposed to a landscape position (not shown).

[0085] In addition to the open position, device **100** may also have a "closed" position illustrated in FIG. **1H**. Again, position sensors **172A** and **172B** can generate a signal indicating that device **100** is in the "closed" position. This can trigger an event that results in a change of displayed information on screen **104** and/or **108**. For example, device **100** may be programmed to stop displaying information on one of the screens, e.g., screen **108**, since a user can only view one screen at a time when device **100** is in the "closed" position. In other embodiments, the signal generated by position sensors **172A** and **172B**, indicating that the device **100** is in the "closed" position, can trigger device **100** to answer an incoming telephone call. The "closed" position can also be a preferred position for utilizing the device **100** as a mobile phone.

[0086] Device **100** can also be used in an "easel" position which is illustrated in FIG. **1I**. In the "easel" position, screens **104** and **108** are angled with respect to each other and facing outward with the edges of screens **104** and **108** substantially horizontal. In this position, device **100** can be configured to display information on both screens **104** and **108** to allow two users to simultaneously interact with device **100**. When device **100** is in the "easel" position, sensors **172A** and **172B** generate a signal indicating that the screens **104** and **108** are positioned at an angle to each other, and the accelerometer **176** can generate a signal indicating that device **100** has been placed so that the edge of screens **104** and **108** are substantially horizontal. The signals can then be used in combination to generate events that trigger changes in the display of information on screens **104** and **108**.

[0087] FIG. **1J** illustrates device **100** in a "modified easel" position. In the "modified easel" position, one of screens **104** or **108** is used as a stand and is faced down on the surface of an object such as a table. This position provides a convenient way for information to be displayed to a user in landscape orientation. Similar to the easel position, when device **100** is in the "modified easel" position, position sensors **172A** and **172B** generate a signal indicating that the screens **104** and **108** are positioned at an angle to each other. The accelerometer **176** would generate a signal indicating that device **100** has been positioned so that one of screens **104** and **108** is faced downwardly and is substantially horizontal. The signals can then be used to generate events that trigger changes in the display of information of screens **104** and **108**. For example, information may not be displayed on the screen that is face down since a user cannot see the screen.

[0088] Transitional states are also possible. When the position sensors **172**A and B and/or accelerometer indicate that the screens are being closed or folded (from open), a closing transitional state is recognized. Conversely when the position sensors **172**A and B indicate that the screens are being opened or folded (from closed), an opening transitional state is recognized. The closing and opening transitional states are typically time-based, or have a maximum time duration from a sensed starting point. Normally, no user input is possible when one of the closing and opening states is in effect. In this manner, incidental user contact with a screen during the closing or opening function is not misinterpreted as user input. In embodiments, another transitional state is possible when the device **100** is closed. This additional transitional state allows the display to switch from one screen **104** to the second screen **108** when the device **100** is closed based on some user input, e.g., a double tap on the screen **110**,**114**.

[0089] As can be appreciated, the description of device **100** is made for illustrative purposes only, and the embodiments are not limited to the specific mechanical features shown in FIGS. **1**A-**1**J and described above. In other embodiments, device **100** may include additional features, including one or more additional buttons, slots, display areas, hinges, and/or locking mechanisms. Additionally, in embodiments, the features described above may be located in different parts of device **100** and still provide similar functionality. Therefore, FIGS. **1**A-**1**J and the description provided above are nonlimiting.

[0090] Hardware Features:

[0091] FIG. **2** illustrates components of a device **100** in accordance with embodiments of the present disclosure. In general, the device **100** includes a primary screen **104** and a secondary screen **108**. While the primary screen **104** and its components are normally enabled in both the opened and closed positions or states, the secondary screen **108** and its components are normally enabled in the opened state but disabled in the closed state. However, even when in the closed state a user or application triggered interrupt (such as in response to a phone application or camera application operation) can flip the active screen, or disable the primary screen **104** and enable the secondary screen **108**, by a suitable command. Each screen **104**, **108** can be touch sensitive and can include different operative areas. For example, a first operative area, within each touch sensitive screen **104** and **108**, may comprise a touch sensitive display **110**, **114**. In general, the touch sensitive display **110**, **114** may comprise a full color, touch sensitive display. A second area within each touch sensitive screen **104** and **108** may comprise a gesture capture region **120**, **124**. The gesture capture region **120**, **124** may comprise an area or region that is outside of the touch sensitive display **110**, **114** area, and that is capable of receiving input, for example in the form of gestures provided by a user. However, the gesture capture region **120**, **124** does not include pixels that can perform a display function or capability.

[0092] A third region of the touch sensitive screens **104** and **108** may comprise a configurable area **112**, **116**. The configurable area **112**, **116** is capable of receiving input and has display or limited display capabilities. In embodiments, the configurable area **112**, **116** may present different input options to the user. For example, the configurable area **112**, **116** may display buttons or other relatable items. Moreover, the identity of displayed buttons, or whether any buttons are displayed at all within the configurable area **112**, **116** of a touch sensitive screen **104** or **108**, may be determined from the context in which the device **100** is used and/or operated. In an exemplary embodiment, the touch sensitive screens **104** and **108** comprise liquid crystal display devices extending across at least those regions of the touch sensitive screens **104** and **108** that are capable of providing visual output to a user, and a capacitive input matrix over those regions of the touch sensitive screens **104** and **108** that are capable of receiving input from the user.

[0093] One or more display controllers **216***a*, **216***b* may be provided for controlling the operation of the touch sensitive screens **104** and **108**, including input (touch sensing) and output (display) functions. In the exemplary embodiment illustrated in FIG. **2**, a separate touch screen controller **216***a* or **216***b* is provided for each touch screen **104** and **108**. In accordance with alternate embodiments, a common or shared touch screen controller **216** may be used to control each of the included touch sensitive screens **104** and **108**. In accordance with still other embodiments, the functions of a touch screen controller **216** may be incorporated into other components, such as a processor **204**.

[0094] The processor **204** may comprise a general purpose programmable processor or controller for executing application programming or instructions. In accordance with at least some embodiments, the processor **204** may include multiple processor cores, and/or implement multiple virtual processors. In accordance with still other embodiments, the processor **204** may include multiple physical processors. As a particular example, the processor **204** may comprise a specially configured application specific integrated circuit (ASIC) or other integrated circuit, a digital signal processor, a controller, a hardwired electronic or logic circuit, a programmable logic device or gate array, a special purpose computer, or the like. The processor **204** generally functions to run programming code or instructions implementing various functions of the device **100**.

[0095] A communication device **100** may also include memory **208** for use in connection with the execution of application programming or instructions by the processor **204**, and for the temporary or long term storage of program instructions and/or data. As examples, the memory **208** may comprise RAM, DRAM, SDRAM, or other solid state memory. Alternatively or in addition, data storage **212** may be provided. Like the memory **208**, the data storage **212** may comprise a solid state memory device or devices. Alternatively or in addition, the data storage **212** may comprise a hard disk drive or other random access memory.

[0096] In support of communications functions or capabilities, the device **100** can include a cellular telephony module **228**. As examples, the cellular telephony module **228** can comprise a GSM, CDMA, FDMA and/or analog cellular telephony transceiver capable of supporting voice, multimedia and/or data transfers over a cellular network. Alternatively or in addition, the device **100** can include an additional or other wireless communications module **232**. As examples, the other wireless communications module **232** can comprise a Wi-Fi, BLUETOOTH™, WiMax, infrared, or other wireless communications link. The cellular telephony module **228**

and the other wireless communications module **232** can each be associated with a shared or a dedicated antenna **224**.

[0097] A port interface **252** may be included. The port interface **252** may include proprietary or universal ports to support the interconnection of the device **100** to other devices or components, such as a dock, which may or may not include additional or different capabilities from those integral to the device **100**. In addition to supporting an exchange of communication signals between the device **100** and another device or component, the docking port **136** and/or port interface **252** can support the supply of power to or from the device **100**. The port interface **252** also comprises an intelligent element that comprises a docking module for controlling communications or other interactions between the device **100** and a connected device or component.

[0098] An input/output module **248** and associated ports may be included to support communications over wired networks or links, for example with other communication devices, server devices, and/or peripheral devices. Examples of an input/output module **248** include an Ethernet port, a Universal Serial Bus (USB) port, Institute of Electrical and Electronics Engineers (IEEE) 1394, or other interface.

[0099] An audio input/output interface/device(s) **244** can be included to provide analog audio to an interconnected speaker or other device, and to receive analog audio input from a connected microphone or other device. As an example, the audio input/output interface/device(s) **244** may comprise an associated amplifier and analog to digital converter. Alternatively or in addition, the device **100** can include an integrated audio input/output device **256** and/or an audio jack for interconnecting an external speaker or microphone. For example, an integrated speaker and an integrated microphone can be provided, to support near talk or speaker phone operations.

[0100] Hardware buttons **158** can be included for example for use in connection with certain control operations. Examples include a master power switch, volume control, etc., as described in conjunction with FIGS. **1**A through **1**J. One or more image capture interfaces/devices **240**, such as a camera, can be included for capturing still and/or video images. Alternatively or in addition, an image capture interface/device **240** can include a scanner or code reader. An image capture interface/device **240** can include or be associated with additional elements, such as a flash or other light source.

[0101] The device **100** can also include a global positioning system (GPS) receiver **236**. In accordance with embodiments of the present invention, the GPS receiver **236** may further comprise a GPS module that is capable of providing absolute location information to other components of the device **100**. An accelerometer(s) **176** may also be included. For example, in connection with the display of information to a user and/or other functions, a signal from the accelerometer **176** can be used to determine an orientation and/or format in which to display that information to the user.

[0102] Embodiments of the present invention can also include one or more position sensor(s) **172**. The position sensor **172** can provide a signal indicating the position of the touch sensitive screens **104** and **108** relative to one another. This information can be provided as an input, for example to a user interface application, to determine an operating mode, characteristics of the touch sensitive displays **110**, **114**, and/or other device **100** operations. As examples, a screen position sensor **172** can comprise a series of Hall effect sensors, a multiple position switch, an optical switch, a Wheatstone bridge, a potentiometer, or other arrangement capable of providing a signal indicating of multiple relative positions the touch screens are in.

[0103] Communications between various components of the device **100** can be carried by one or more buses **222**. In addition, power can be supplied to the components of the device **100** from a power source and/or power control module **260**. The power control module **260** can, for example, include a battery, an AC to DC converter, power control logic, and/or ports for interconnecting the device **100** to an external source of power.

[0104] Device State:

[0105] FIGS. **3**A and **3**B represent illustrative states of device **100**. While a number of illustrative states are shown, and transitions from a first state to a second state, it is to be appreciated that the illustrative state diagram may not encompass all possible states and/or all possible transitions from a first state to a second state. As illustrated in FIG. **3**, the various arrows between the states (illustrated by the state represented in the circle) represent a physical change that occurs to the device **100**, that is detected by one or more of hardware and software, the detection triggering one or more of a hardware and/or software interrupt that is used to control and/or manage one or more functions of device **100**.

[0106] As illustrated in FIG. **3**A, there are twelve exemplary "physical" states: closed **304**, transition **308** (or opening transitional state), easel **312**, modified easel **316**, open **320**, inbound/outbound call or communication **324**, image/video capture **328**, transition **332** (or closing transitional state), landscape **340**, docked **336**, docked **344** and landscape **348**. Next to each illustrative state is a representation of the physical state of the device **100** with the exception of states **324** and **328**, where the state is generally symbolized by the international icon for a telephone and the icon for a camera, respectfully.

[0107] In state **304**, the device is in a closed state with the device **100** generally oriented in the portrait direction with the primary screen **104** and the secondary screen **108** back-to-back in different planes (see FIG. **1**H). From the closed state, the device **100** can enter, for example, docked state **336**, where the device **100** is coupled with a docking station, docking cable, or in general docked or associated with one or more other devices or peripherals, or the landscape state **340**, where the device **100** is generally oriented with the primary screen **104** facing the user, and the primary screen **104** and the secondary screen **108** being back-to-back.

[0108] In the closed state, the device can also move to a transitional state where the device remains closed but the display is moved from one screen **104** to another screen **108** based on a user input, e.g., a double tap on the screen **110**, **114**. Still another embodiment includes a bilateral state. In the bilateral state, the device remains closed, but a single application displays at least one window on both the first display **110** and the second display **114**. The windows shown on the first and second display **110**, **114** may be the same or different based on the application and the state of that application. For example, while acquiring an image with a camera, the device may display the view finder on the first display **110** and displays a preview for the photo subjects (full screen and mirrored left-to-right) on the second display **114**.

[0109] In state **308**, a transition state from the closed state **304** to the semi-open state or easel state **312**, the device **100** is shown opening with the primary screen **104** and the secondary screen **108** being rotated around a point of axis coincidence with the hinge. Upon entering the easel state **312**, the primary screen **104** and the secondary screen **108** are separated from one another such that, for example, the device **100** can sit in an easel-like configuration on a surface.

[0110] In state **316**, known as the modified easel position, the device **100** has the primary screen **104** and the secondary screen **108** in a similar relative relationship to one another as in the easel state **312**, with the difference being one of the primary screen **104** or the secondary screen **108** are placed on a surface as shown.

[0111] State **320** is the open state where the primary screen **104** and the secondary screen **108** are generally on the same plane. From the open state, the device **100** can transition to the docked state **344** or the open landscape state **348**. In the open state **320**, the primary screen **104** and the secondary screen **108** are generally in the portrait-like orientation while in landscaped state **348** the primary screen **104** and the secondary screen **108** are generally in a landscape-like orientation.

[0112] State **324** is illustrative of a communication state, such as when an inbound or outbound call is being received or placed, respectively, by the device **100**. While not illustrated for clarity, it should be appreciated the device **100** can transition to the inbound/outbound call state **324** from any state illustrated in FIG. **3**. In a similar manner, the image/video capture state **328** can be entered into from any other state in FIG. **3**, with the image/video capture state **328** allowing the device **100** to take one or more images via a camera and/or videos with a video capture device **240**.

[0113] Transition state **322** illustratively shows primary screen **104** and the secondary screen **108** being closed upon one another for entry into, for example, the closed state **304**.

[0114] FIG. **3**B illustrates, with reference to the key, the inputs that are received to detect a transition from a first state to a second state. In FIG. **3**B, various combinations of states are shown with in general, a portion of the columns being directed toward a portrait state **352**, a landscape state **356**, and a portion of the rows being directed to portrait state **360** and landscape state **364**.

[0115] In FIG. **3**B, the Key indicates that "H" represents an input from one or more Hall Effect sensors, "A" represents an input from one or more accelerometers, "T" represents an input from a timer, "P" represents a communications trigger input and "I" represents an image and/or video capture request input. Thus, in the center portion **376** of the chart, an input, or combination of inputs, are shown that represent how the device **100** detects a transition from a first physical state to a second physical state.

[0116] As discussed, in the center portion of the chart **376**, the inputs that are received enable the detection of a transition from, for example, a portrait open state to a landscape easel state—shown in bold—"HAT." For this exemplary transition from the portrait open to the landscape easel state, a Hall Effect sensor ("H"), an accelerometer ("A") and a timer ("T") input may be needed. The timer input can be derived from, for example, a clock associated with the processor.

[0117] In addition to the portrait and landscape states, a docked state **368** is also shown that is triggered based on the receipt of a docking signal **372**. As discussed above and in relation to FIG. **3**, the docking signal can be triggered by the association of the device **100** with one or more other device **100***s*, accessories, peripherals, smart docks, or the like.

[0118] User Interaction:

[0119] FIGS. **4**A through **4**H depict various graphical representations of gesture inputs that may be recognized by the screens **104**, **108**. The gestures may be performed not only by a user's body part, such as a digit, but also by other devices, such as a stylus, that may be sensed by the contact sensing portion(s) of a screen **104**, **108**. In general, gestures are interpreted differently, based on where the gestures are performed (either directly on the display **110**, **114** or in the gesture capture region **120**, **124**). For example, gestures in the display **110**,**114** may be directed to a desktop or application, and gestures in the gesture capture region **120**, **124** may be interpreted as for the system.

[0120] With reference to FIGS. **4**A-**4**H, a first type of gesture, a touch gesture **420**, is substantially stationary on the screen **104**,**108** for a selected length of time. A circle **428** represents a touch or other contact type received at particular location of a contact sensing portion of the screen. The circle **428** may include a border **432**, the thickness of which indicates a length of time that the contact is held substantially stationary at the contact location. For instance, a tap **420** (or short press) has a thinner border **432***a* than the border **432***b* for a long press **424** (or for a normal press). The long press **424** may involve a contact that remains substantially stationary on the screen for longer time period than that of a tap **420**. As will be appreciated, differently defined gestures may be registered depending upon the length of time that the touch remains stationary prior to contact cessation or movement on the screen.

[0121] With reference to FIG. **4**C, a drag gesture **400** on the screen **104**,**108** is an initial contact (represented by circle **428**) with contact movement **436** in a selected direction. The initial contact **428** may remain stationary on the screen **104**,**108** for a certain amount of time represented by the border **432**. The drag gesture typically requires the user to contact an icon, window, or other displayed image at a first location followed by movement of the contact in a drag direction to a new second location desired for the selected displayed image. The contact movement need not be in a straight line but have any path of movement so long as the contact is substantially continuous from the first to the second locations.

[0122] With reference to FIG. **4**D, a flick gesture **404** on the screen **104**,**108** is an initial contact (represented by circle **428**) with truncated contact movement **436** (relative to a drag gesture) in a selected direction. In embodiments, a flick has a higher exit velocity for the last movement in the gesture compared to the drag gesture. The flick gesture can, for instance, be a finger snap following initial contact. Compared to a drag gesture, a flick gesture generally does not require continual contact with the screen **104**,**108** from the first location of a displayed image to a predetermined second location. The contacted displayed image is moved by the flick gesture in the direction of the flick gesture to the predetermined second location. Although both gestures commonly can move a displayed image from a first location to a second location, the temporal duration and distance of travel of the contact on the screen is generally less for a flick than for a drag gesture.

[0123] With reference to FIG. **4**E, a pinch gesture **408** on the screen **104**,**108** is depicted. The pinch gesture **408** may be initiated by a first contact **428** to the screen **104**,**108** by, for example, a first digit and a second contact **428***b* to the screen **104**,**108** by, for example, a second digit. The first and second contacts **428***a*,*b* may be detected by a common contact sensing portion of a common screen **104**,**108**, by different contact sensing portions of a common screen **104** or **108**, or by different contact sensing portions of different screens. The first contact **428***a* is held for a first amount of time, as represented by the border **432***a*, and the second contact **428***b* is held for a second amount of time, as represented by the border **432***b*. The first and second amounts of time are generally substantially the same, and the first and second contacts **428** *a, b* generally occur substantially simultaneously. The first and second contacts **428** *a, b* generally also include corresponding first and second contact movements **436** *a, b*, respectively. The first and second contact movements **436** *a, b* are generally in opposing directions. Stated another way, the first contact movement **436***a* is towards the second contact **436***b*, and the second contact movement **436***b* is towards the first contact **436***a*. More simply stated, the pinch gesture **408** may be accomplished by a user's digits touching the screen **104**,**108** in a pinching motion.

[0124] With reference to FIG. **4**F, a spread gesture **410** on the screen **104**,**108** is depicted. The spread gesture **410** may be initiated by a first contact **428***a* to the screen **104**,**108** by, for example, a first digit and a second contact **428***b* to the screen **104**,**108** by, for example, a second digit. The first and second contacts **428***a,b* may be detected by a common contact sensing portion of a common screen **104**,**108**, by different contact sensing portions of a common screen **104**,**108**, or by different contact sensing portions of different screens. The first contact **428***a* is held for a first amount of time, as represented by the border **432***a*, and the second contact **428***b* is held for a second amount of time, as represented by the border **432***b*. The first and second amounts of time are generally substantially the same, and the first and second contacts **428** *a, b* generally occur substantially simultaneously. The first and second contacts **428** *a, b* generally also include corresponding first and second contact movements **436***a, b*, respectively. The first and second contact movements **436** *a, b* are generally in a common direction. Stated another way, the first and second contact movements **436** *a, b* are away from the first and second contacts **428***a, b*. More simply stated, the spread gesture **410** may be accomplished by a user's digits touching the screen **104**,**108** in a spreading motion.

[0125] The above gestures may be combined in any manner, such as those shown by FIGS. **4**G and **4**H, to produce a determined functional result. For example, in FIG. **4**G a tap gesture **420** is combined with a drag or flick gesture **412** in a direction away from the tap gesture **420**. In FIG. **4**H, a tap gesture **420** is combined with a drag or flick gesture **412** in a direction towards the tap gesture **420**.

[0126] The functional result of receiving a gesture can vary depending on a number of factors, including a state of the device **100**, display **110**, **114**, or screen **104**, **108**, a context associated with the gesture, or sensed location of the gesture. The state of the device commonly refers to one or more of a configuration of the device **100**, a display orientation, and user and other inputs received by the device **100**. Context commonly refers to one or more of the particular application (s) selected by the gesture and the portion(s) of the application currently executing, whether the application is a single- or multi-screen application, and whether the application is a multi-screen application displaying one or more windows in one or more screens or in one or more stacks. Sensed location of the gesture commonly refers to whether the sensed set(s) of gesture location coordinates are on a touch sensitive display **110**, **114** or a gesture capture region **120**, **124**, whether the sensed set(s) of gesture location coordinates are associated with a common or different display or screen **104**,**108**, and/or what portion of the gesture capture region contains the sensed set(s) of gesture location coordinates.

[0127] A tap, when received by an a touch sensitive display **110**, **114**, can be used, for instance, to select an icon to initiate or terminate execution of a corresponding application, to maximize or minimize a window, to reorder windows in a stack, and to provide user input such as by keyboard display or other displayed image. A drag, when received by a touch sensitive display **110**, **114**, can be used, for instance, to relocate an icon or window to a desired location within a display, to reorder a stack on a display, or to span both displays (such that the selected window occupies a portion of each display simultaneously). A flick, when received by a touch sensitive display **110**, **114** or a gesture capture region **120**, **124**, can be used to relocate a window from a first display to a second display or to span both displays (such that the selected window occupies a portion of each display simultaneously). Unlike the drag gesture, however, the flick gesture is generally not used to move the displayed image to a specific user-selected location but to a default location that is not configurable by the user.

[0128] The pinch gesture, when received by a touch sensitive display **110**, **114** or a gesture capture region **120**, **124**, can be used to minimize or otherwise increase the displayed area or size of a window (typically when received entirely by a common display), to switch windows displayed at the top of the stack on each display to the top of the stack of the other display (typically when received by different displays or screens), or to display an application manager (a "pop-up window" that displays the windows in the stack). The spread gesture, when received by a touch sensitive display **110**, **114** or a gesture capture region **120**, **124**, can be used to maximize or otherwise decrease the displayed area or size of a window, to switch windows displayed at the top of the stack on each display to the top of the stack of the other display (typically when received by different displays or screens), or to display an application manager (typically when received by an off-screen gesture capture region on the same or different screens).

[0129] The combined gestures of FIG. **4**G, when received by a common display capture region in a common display or screen **104**,**108**, can be used to hold a first window stack location in a first stack constant for a display receiving the gesture while reordering a second window stack location in a second window stack to include a window in the display receiving the gesture. The combined gestures of FIG. **4**H, when received by different display capture regions in a common display or screen **104**,**108** or in different displays or screens, can be used to hold a first window stack location in a first window stack constant for a display receiving the tap part of the gesture while reordering a second window stack location in a second window stack to include a window in the display receiving the flick or drag gesture. Although specific gestures and gesture capture regions in the preceding examples have been associated with corresponding sets of functional results, it is to be appreciated that these associations can be redefined in any manner to produce differing associations between gestures and/or gesture capture regions and/or functional results.

[0130] Firmware and Software:

[0131] The memory **508** may store and the processor **504** may execute one or more software components. These components can include at least one operating system (OS) **516**, an application manager **562**, a desktop **566**, and/or one or more applications **564***a* and/or **564***b* from an application store **560**. The OS **516** can include a framework **520**, one or more frame buffers **548**, one or more drivers **512**, previously described in conjunction with FIG. **2**, and/or a kernel **518**. The OS **516** can be any software, consisting of programs and data, which manages computer hardware resources and provides common services for the execution of various applications **564**. The OS **516** can be any operating system and, at least in some embodiments, dedicated to mobile devices, including, but not limited to, Linux, ANDROID™, iPhone OS (IOS™), WINDOWS PHONE 7 ™, etc. The OS **516** is operable to provide functionality to the phone by executing one or more operations, as described herein.

[0132] The applications **564** can be any higher level software that executes particular functionality for the user. Applications **564** can include programs such as email clients, web browsers, texting applications, games, media players, office suites, etc. The applications **564** can be stored in an application store **560**, which may represent any memory or data storage, and the management software associated therewith, for storing the applications **564**. Once executed, the applications **564** may be run in a different area of memory **508**.

[0133] The framework **520** may be any software or data that allows the multiple tasks running on the device to interact. In embodiments, at least portions of the framework **520** and the discrete components described hereinafter may be considered part of the OS **516** or an application **564**. However, these portions will be described as part of the framework **520**, but those components are not so limited. The framework **520** can include, but is not limited to, a Multi-Display Management (MDM) module **524**, a Surface Cache module **528**, a Window Management module **532**, an Input Management module **536**, a Task Management module **540**, an Application Model Manager **542**, a Display Controller, one or more frame buffers **548**, a task stack **552**, one or more window stacks **550** (which is a logical arrangement of windows and/or desktops in a display area), and/or an event buffer **556**.

[0134] The MDM module **524** includes one or more modules that are operable to manage the display of applications or other data on the screens of the device. An embodiment of the MDM module **524** is described in conjunction with FIG. **5**B. In embodiments, the MDM module **524** receives inputs from the other OS **516** components, such as, the drivers **512**, and from the applications **564** to determine continually the state of the device **100**. The inputs assist the MDM module **524** in determining how to configure and allocate the displays according to the application's preferences and requirements, and the user's actions. Once a determination for display configurations is made, the MDM module **524** can bind the applications **564** to a display. The configuration may then be provided to one or more other components to generate a window with a display.

[0135] The Surface Cache module **528** includes any memory or storage and the software associated therewith to store or cache one or more images of windows. A series of active and/or non-active windows (or other display objects, such as, a desktop display) can be associated with each display. An active window (or other display object) is currently displayed. A non-active windows (or other display objects) were opened and, at some time, displayed but are now not displayed. To enhance the user experience, before a window transitions from an active state to an inactive state, a "screen shot" of a last generated image of the window (or other display object) can be stored. The Surface Cache module **528** may be operable to store a bitmap of the last active image of a window (or other display object) not currently displayed. Thus, the Surface Cache module **528** stores the images of non-active windows (or other display objects) in a data store.

[0136] In embodiments, the Window Management module **532** is operable to manage the windows (or other display objects) that are active or not active on each of the displays. The Window Management module **532**, based on information from the MDM module **524**, the OS **516**, or other components, determines when a window (or other display object) is visible or not active. The Window Management module **532** may then put a non-visible window (or other display object) in a "not active state" and, in conjunction with the Task Management module Task Management **540** suspends the application's operation. Further, the Window Management module **532** may assign, through collaborative interaction with the MDM module **524**, a display identifier to the window (or other display object) or manage one or more other items of data associated with the window (or other display object). The Window Management module **532** may also provide the stored information to the application **564**, the Task Management module **540**, or other components interacting with or associated with the window (or other display object). The Window Management module **532** can also associate an input task with a window based on window focus and display coordinates within the motion space.

[0137] The Input Management module **536** is operable to manage events that occur with the device. An event is any input into the window environment, for example, a user interface interactions with a user. The Input Management module **536** receives the events and logically stores the events in an event buffer **556**. Events can include such user interface interactions as a "down event," which occurs when a screen **104**, **108** receives a touch signal from a user, a "move event," which occurs when the screen **104**, **108** determines that a user's finger is moving across a screen(s), an "up event, which occurs when the screen **104**, **108** determines that the user has stopped touching the screen **104**, **108**, etc. These events are received, stored, and forwarded to other modules by the Input Management module **536**. The Input Management module **536** may also map screen inputs to a motion space which is the culmination of all physical and virtual display available on the device.

[0138] The motion space is a virtualized space that includes all touch sensitive displays **110**,**114** "tiled" together to mimic the physical dimensions of the device **100**. For example, when the device **100** is unfolded, the motion space size may be 960×800, which may be the number of pixels in the combined display area for both touch sensitive displays **110**, **114**. If a user touches on a first touch sensitive display **110** on location (**40**, **40**), a full screen window can receive touch event with location (**40**, **40**). If a user touches on a second touch sensitive display **114**, with location (**40**, **40**), the full screen window can receive touch event with location (**520**, **40**), because the second touch sensitive display **114** is on the right side of the first touch sensitive display **110**, so the device **100** can offset the touch by the first touch sensitive display's **110** width, which is 480 pixels. When a hardware event occurs with location info from a driver **512**, the framework **520** can up-scale the physical location to the motion space because the location of the event may be different based on the device orientation and state. The motion space may be as described in U.S. patent application Ser. No. 13/187,026, filed Jul. 20, 2011, entitled "Systems and Methods for Receiving Gesture Inputs Spanning Multiple Input Devices," which is hereby incorporated by reference in its entirety for all that it teaches and for all purposes.

[0139] A task can be an application and a sub-task can be an application component that provides a window with which users can interact to do something, such as dial the phone, take a photo, send an email, or view a map. Each task may be given a window in which to draw a user interface. The window typically fills a display (for example, touch sensitive display **110**,**114**), but may be smaller than the display **110**, **114** and float on top of other windows. An application usually consists of multiple sub-tasks that are loosely bound to each other. Typically, one task in an application is specified as the "main" task, which is presented to the user when launching the application for the first time. Each task can then start another task or sub-task to perform different actions.

[0140] The Task Management module **540** is operable to manage the operation of one or more applications **564** that may be executed by the device. Thus, the Task Management module **540** can receive signals to launch, suspend, terminate, etc. an application or application sub-tasks stored in the application store **560**. The Task Management module **540** may then instantiate one or more tasks or sub-tasks of the application **564** to begin operation of the application **564**. Further, the Task Management Module **540** may launch, suspend, or terminate a task or sub-task as a result of user input or as a result of a signal from a collaborating framework **520** component. The Task Management Module **540** is responsible for managing the lifecycle of applications (tasks and sub-task) from when the application is launched to when the application is terminated.

[0141] The processing of the Task Management Module **540** is facilitated by a task stack **552**, which is a logical structure associated with the Task Management Module **540**. The task stack **552** maintains the state of all tasks and subtasks on the device **100**. When some component of the operating system **516** requires a task or sub-task to transition in its lifecycle, the OS **516** component can notify the Task Management Module **540**. The Task Management Module **540** may then locate the task or sub-task, using identification information, in the task stack **552**, and send a signal to the task or sub-task indicating what kind of lifecycle transition the task needs to execute. Informing the task or sub-task of the transition allows the task or sub-task to prepare for the lifecycle state transition. The Task Management Module **540** can then execute the state transition for the task or sub-task. In embodiments, the state transition may entail triggering the OS kernel **518** to terminate the task when termination is required.

[0142] Further, the Task Management module **540** may suspend the application **564** based on information from the Window Management Module **532**. Suspending the application **564** may maintain application data in memory but may limit or stop the application **564** from rendering a window or user interface. Once the application becomes active again, the Task Management module **540** can again trigger the application to render its user interface. In embodiments, if a task is suspended, the task may save the task's state in case the task is terminated. In the suspended state, the application task may not receive input because the application window is not visible to the user.

[0143] The frame buffer **548** is a logical structure(s) used to render the user interface. The frame buffer **548** can be created and destroyed by the OS kernel **518**. However, the Display Controller **544** can write the image data, for the visible windows, into the frame buffer **548**. A frame buffer **548** can be associated with one screen or multiple screens. The association of a frame buffer **548** with a screen can be controlled dynamically by interaction with the OS kernel **518**. A composite display may be created by associating multiple screens with a single frame buffer **548**. Graphical data used to render an application's window user interface may then be written to the single frame buffer **548**, for the composite display, which is output to the multiple screens **104**,**108**. The Display Controller **544** can direct an application's user interface to a portion of the frame buffer **548** that is mapped to a particular display **110**,**114**, thus, displaying the user interface on only one screen **104** or **108**. The Display Controller **544** can extend the control over user interfaces to multiple applications, controlling the user interfaces for as many displays as are associated with a frame buffer **548** or a portion thereof. This approach compensates for the multiple physical screens **104**, **108** that are in use by the software component above the Display Controller **544**.

[0144] The Application Manager **562** is an application that provides a presentation layer for the window environment. Thus, the Application Manager **562** provides the graphical model for rendering by the Task Management Module **540**. Likewise, the Desktop **566** provides the presentation layer for the Application Store **560**. Thus, the desktop provides a graphical model of a surface having selectable application icons for the Applications **564** in the Application Store **560** that can be provided to the Window Management Module **556** for rendering.

[0145] Further, the framework can include an Application Model Manager (AMM) **542**. The Application Manager **562** may interface with the AMM **542**. In embodiments, the AMM **542** receives state change information from the device **100** regarding the state of applications (which are running or suspended). The AMM **542** can associate bit map images from the Surface Cache Module **528** to the tasks that are alive (running or suspended). Further, the AMM **542** can convert the logical window stack maintained in the Task Manager Module **540** to a linear ("film strip" or "deck of cards") organization that the user perceives when the using the off gesture capture area **120** to sort through the windows. Further, the AMM **542** may provide a list of executing applications to the Application Manager **562**.

[0146] An embodiment of the MDM module **524** is shown in FIG. **5**B. The MDM module **524** is operable to determine the state of the environment for the device, including, but not limited to, the orientation of the device, whether the device **100** is opened or closed, what applications **564** are executing, how the applications **564** are to be displayed, what actions the user is conducting, the tasks being displayed, etc. To configure the display, the MDM module **524** interprets these environmental factors and determines a display configuration, as described in conjunction with FIGS. **6**A-**6**J. Then, the MDM module **524** can bind the applications **564** or other device components to the displays. The configuration may then be sent to the Display Controller **544** and/or the other components within the OS **516** to generate the display. The MDM module **524** can include one or more of, but is not limited to, a Display Configuration Module **568**, a Preferences Module **572**, a Device State Module **574**, a Gesture Module **576**, a Requirements Module **580**, an Event Module **584**, and/or a Binding Module **588**.

[0147] The Display Configuration Module **568** determines the layout for the display. In embodiments, the Display Configuration Module **568** can determine the environmental factors. The environmental factors may be received from one or more other MDM modules **524** or from other sources. The Display Configuration Module **568** can then determine from the list of factors the best configuration for the display. Some embodiments of the possible configurations and the factors associated therewith are described in conjunction with FIGS. **6**A-**6**F.

[0148] The Preferences Module **572** is operable to determine display preferences for an application **564** or other component. For example, an application can have a preference for Single or Dual displays. The Preferences Module **572** can determine an application's display preference (e.g., by inspecting the application's preference settings) and may allow the application **564** to change to a mode (e.g., single screen, dual screen, max, etc.) if the device **100** is in a state that can accommodate the preferred mode. However, some user interface policies may disallow a mode even if the mode is available. As the configuration of the device changes, the preferences may be reviewed to determine if a better display configuration can be achieved for an application **564**.

[0149] The Device State Module **574** is operable to determine or receive the state of the device. The state of the device can be as described in conjunction with FIGS. **3**A and **3**B. The state of the device can be used by the Display Configuration Module **568** to determine the configuration for the display. As such, the Device State Module **574** may receive inputs and interpret the state of the device. The state information is then provided to the Display Configuration Module **568**.

[0150] The Gesture Module **576** is shown as part of the MDM module **524**, but, in embodiments, the Gesture module **576** may be a separate Framework **520** component that is separate from the MDM module **524**. In embodiments, the Gesture Module **576** is operable to determine if the user is conducting any actions on any part of the user interface. In alternative embodiments, the Gesture Module **576** receives user interface actions from the configurable area **112**,**116** only. The Gesture Module **576** can receive touch events that occur on the configurable area **112**,**116** (or possibly other user interface areas) by way of the Input Management Module **536** and may interpret the touch events (using direction, speed, distance, duration, and various other parameters) to determine what kind of gesture the user is performing. When a gesture is interpreted, the Gesture Module **576** can initiate the processing of the gesture and, by collaborating with other Framework **520** components, can manage the required window animation. The Gesture Module **576** collaborates with the Application Model Manager **542** to collect state information with respect to which applications are running (active or paused) and the order in which applications must appear when a user gesture is performed. The Gesture Module **576** may also receive references to bitmaps (from the Surface Cache Module **528**) and live windows so that when a gesture occurs it can instruct the Display Controller **544** how to move the window(s) across the display **110**,**114**. Thus, suspended applications may appear to be running when those windows are moved across the display **110**,**114**.

[0151] Further, the Gesture Module **576** can receive task information either from the Task Manage Module **540** or the Input Management module **536**. The gestures may be as defined in conjunction with FIGS. **4**A through **4**H. For example, moving a window causes the display to render a series of display frames that illustrate the window moving. The gesture associated with such user interface interaction can be received and interpreted by the Gesture Module **576**. The information about the user gesture is then sent to the Task Management Module **540** to modify the display binding of the task.

[0152] The Requirements Module **580**, similar to the Preferences Module **572**, is operable to determine display requirements for an application **564** or other component. An application can have a set display requirement that must be observed. Some applications require a particular display orientation. For example, the application "Angry Birds" can only be displayed in landscape orientation. This type of display requirement can be determined or received, by the Requirements Module **580**. As the orientation of the device changes, the Requirements Module **580** can reassert the display requirements for the application **564**. The Display Configuration Module **568** can generate a display configuration that is in accordance with the application display requirements, as provided by the Requirements Module **580**.

[0153] The Event Module **584**, similar to the Gesture Module **576**, is operable to determine one or more events occurring with an application or other component that can affect the user interface. Thus, the Event Module **584** can receive event information either from the event buffer **556** or the Task Management module **540**. These events can change how the tasks are bound to the displays. The Event Module **584** can collect state change information from other Framework **520** components and act upon that state change information. In an example, when the phone is opened or closed or when an orientation change has occurred, a new message may be rendered in a secondary screen. The state change based on the event can be received and interpreted by the Event Module **584**. The information about the events then may be sent to the Display Configuration Module **568** to modify the configuration of the display.

[0154] The Binding Module **588** is operable to bind the applications **564** or the other components to the configuration determined by the Display Configuration Module **568**. A binding associates, in memory, the display configuration for each application with the display and mode of the application. Thus, the Binding Module **588** can associate an application with a display configuration for the application (e.g. landscape, portrait, multi-screen, etc.). Then, the Binding Module **588** may assign a display identifier to the display. The display identifier associated the application with a particular display of the device **100**. This binding is then stored and provided to the Display Controller **544**, the other components of the OS **516**, or other components to properly render the display. The binding is dynamic and can change or be updated based on configuration changes associated with events, gestures, state changes, application preferences or requirements, etc.

[0155] User Interface Configurations:

[0156] With reference now to FIGS. **6**A-J, various types of output configurations made possible by the device **100** will be described hereinafter.

[0157] FIGS. **6**A and **6**B depict two different output configurations of the device **100** being in a first state. Specifically, FIG. **6**A depicts the device **100** being in a closed portrait state **304** where the data is displayed on the primary screen **104**. In this example, the device **100** displays data via the touch sensitive display **110** in a first portrait configuration **604**. As can be appreciated, the first portrait configuration **604** may only display a desktop or operating system home screen. Alternatively, one or more windows may be presented in a portrait orientation while the device **100** is displaying data in the first portrait configuration **604**.

[0158] FIG. **6**B depicts the device **100** still being in the closed portrait state **304**, but instead data is displayed on the secondary screen **108**. In this example, the device **100** displays data via the touch sensitive display **114** in a second portrait configuration **608**.

[0159] It may be possible to display similar or different data in either the first or second portrait configuration **604**, **608**. It may also be possible to transition between the first portrait configuration **604** and second portrait configuration **608** by providing the device **100** *a* user gesture (e.g., a double tap gesture), a menu selection, or other means. Other suitable gestures may also be employed to transition between configurations. Furthermore, it may also be possible to transition the device **100** from the first or second portrait configuration **604**, **608** to any other configuration described herein depending upon which state the device **100** is moved.

[0160] An alternative output configuration may be accommodated by the device **100** being in a second state. Specifically, FIG. **6**C depicts a third portrait configuration where data is displayed simultaneously on both the primary screen **104** and the secondary screen **108**. The third portrait configuration may be referred to as a Dual-Portrait (PD) output configuration. In the PD output configuration, the touch sensitive display **110** of the primary screen **104** depicts data in the first portrait configuration **604** while the touch sensitive display **114** of the secondary screen **108** depicts data in the second portrait configuration **608**. The simultaneous presentation of the first portrait configuration **604** and the second portrait configuration **608** may occur when the device **100** is in an open portrait state **320**. In this configuration, the device **100** may display one application window in one display **110** or **114**, two application windows (one in each display **110** and **114**), one application window and one desktop, or one desktop. Other configurations may be possible. It should be appreciated that it may also be possible to transition the device **100** from the simultaneous display of configurations **604**, **608** to any other configuration described herein depending upon which state the device **100** is moved. Furthermore, while in this state, an application's display preference may place the device into bilateral mode, in which both displays are active to display different windows in the same application. For example, a Camera application may display a viewfinder and controls on one side, while the other side displays a mirrored preview that can be seen by the photo subjects. Games involving simultaneous play by two players may also take advantage of bilateral mode.

[0161] FIGS. **6**D and **6**E depicts two further output configurations of the device **100** being in a third state. Specifically, FIG. **6**D depicts the device **100** being in a closed landscape state **340** where the data is displayed on the primary screen **104**. In this example, the device **100** displays data via the touch sensitive display **110** in a first landscape configuration **612**. Much like the other configurations described herein, the first landscape configuration **612** may display a desktop, a home screen, one or more windows displaying application data, or the like.

[0162] FIG. **6**E depicts the device **100** still being in the closed landscape state **340**, but instead data is displayed on the secondary screen **108**. In this example, the device **100** displays data via the touch sensitive display **114** in a second landscape configuration **616**. It may be possible to display similar or different data in either the first or second portrait configuration **612**, **616**. It may also be possible to transition between the first landscape configuration **612** and second landscape configuration **616** by providing the device **100** with one or both of a twist and tap gesture or a flip and slide gesture. Other suitable gestures may also be employed to transition between configurations. Furthermore, it may also be possible to transition the device **100** from the first or second landscape configuration **612**, **616** to any other configuration described herein depending upon which state the device **100** is moved.

[0163] FIG. **6**F depicts a third landscape configuration where data is displayed simultaneously on both the primary screen **104** and the secondary screen **108**. The third landscape configuration may be referred to as a Dual-Landscape (LD) output configuration. In the LD output configuration, the touch sensitive display **110** of the primary screen **104** depicts data in the first landscape configuration **612** while the touch sensitive display **114** of the secondary screen **108** depicts data in the second landscape configuration **616**. The simultaneous presentation of the first landscape configuration **612** and the second landscape configuration **616** may occur when the device **100** is in an open landscape state **340**. It should be appreciated that it may also be possible to transition the device **100** from the simultaneous display of configurations **612**, **616** to any other configuration described herein depending upon which state the device **100** is moved.

[0164] FIGS. **6**G and **6**H depict two views of a device **100** being in yet another state. Specifically, the device **100** is depicted as being in an easel state **312**. FIG. **6**G shows that a first easel output configuration **618** may be displayed on the touch sensitive display **110**. FIG. **6**H shows that a second easel output configuration **620** may be displayed on the touch sensitive display **114**. The device **100** may be configured to depict either the first easel output configuration **618** or the second easel output configuration **620** individually. Alternatively, both the easel output configurations **618**, **620** may be presented simultaneously. In some embodiments, the easel output configurations **618**, **620** may be similar or identical to the landscape output configurations **612**, **616**. The device **100** may also be configured to display one or both of the easel output configurations **618**, **620** while in a modified easel state **316**. It should be appreciated that simultaneous utilization of the easel output configurations **618**, **620** may facilitate two-person games (e.g., Battleship®, chess, checkers, etc.), multi-user conferences where two or more users share the same device **100**, and other applications. As can be appreciated, it may also be possible to transition the device **100** from the display of one or both configurations **618**, **620** to any other configuration described herein depending upon which state the device **100** is moved.

[0165] FIG. **6**I depicts yet another output configuration that may be accommodated while the device **100** is in an open portrait state **320**. Specifically, the device **100** may be configured to present a single continuous image across both touch sensitive displays **110**, **114** in a portrait configuration referred to herein as a Portrait-Max (PMax) configuration **624**. In this configuration, data (e.g., a single image, application, window, icon, video, etc.) may be split and displayed partially on one of the touch sensitive displays while the other portion of the data is displayed on the other touch sensitive display. The Pmax configuration **624** may facilitate a larger display and/or better resolution for displaying a particular image on the device **100**. Similar to other output configurations, it may be possible to transition the device **100** from the Pmax configuration **624** to any other output configuration described herein depending upon which state the device **100** is moved.

[0166] FIG. **6**J depicts still another output configuration that may be accommodated while the device **100** is in an open landscape state **348**. Specifically, the device **100** may be configured to present a single continuous image across both touch sensitive displays **110**, **114** in a landscape configuration referred to herein as a Landscape-Max (LMax) configuration **628**. In this configuration, data (e.g., a single image, application, window, icon, video, etc.) may be split and displayed partially on one of the touch sensitive displays while the other portion of the data is displayed on the other touch sensitive display. The Lmax configuration **628** may facilitate a larger display and/or better resolution for displaying a particular image on the device **100**. Similar to other output configurations, it may be possible to transition the device **100** from the Lmax configuration **628** to any other output configuration described herein depending upon which state the device **100** is moved.

[0167] The device **100** manages desktops and/or windows with at least one window stack **700**, as shown in FIGS. **7**A and **7**B. A window stack **700** is a logical arrangement of active and/or inactive windows or display objects for a multi-screen device. For example, the window stack **700** may be logically similar to a deck of cards or a stack of bricks, where one or more windows or display objects (e.g., desktops) are arranged in an order, as shown in FIGS. **7**A and **7**B. An active window is a window that is currently being displayed on at least one of the touch sensitive displays **110**, **114**. For example, window **1 708** is an active windows and displayed on only one of the touch sensitive displays **114**. In the embodiment shown in FIG. **7**A, the device **100** is in a closed state **304** displaying window **1 708** in configuration **608**. An inactive window is a window that was opened and displayed but is now "behind" an active window and not being displayed. In embodiments, an inactive window may be for an application that is suspended, and thus, the window is not displaying active content. For example, window **2 712** and window **3 716** are inactive windows.

[0168] A window stack **700** may have various arrangements or organizational structures. In the embodiment shown in FIG. **7**A, the device **100** includes a first stack **704** associated with a first touch sensitive display **114**, while the device **100** is in the closed state **304** in the second portrait configuration **608**. Thus, each touch sensitive display **114** can have an associated window stack **704**. In other embodiments, there is a single window stack that encompasses all of a composite display. The composite display is a logical structure that defines the entire display space including the two touch sensitive displays **110**, **114**. The device **100** can have a single stack for the composite display where the windows or display objects are sized to occupy a portion or all of the composite display. Thus, the stack **704** can represent a portion of a larger composite window stack, part of which is not shown because the device **100** is in a closed state **304**.

[0169] In FIG. **7**B, the two window stacks (or two portions of the window stack) **704**, **724** may have a different number of windows or display objects arranged in the respective stacks **704**, **724**. Further, the two window stacks **704**, **724** can also be identified differently and managed separately. As shown in FIG. **7**A, the first window stack **704** can be arranged in order from a first window **708** to a next window **712** to a last window **716** and finally to a desktop **720**, which, in embodiments, is at the "bottom" of the window stack **704**. In embodiments, the desktop **720** is not always at the "bottom" as application windows can be arranged in the window stack below the desktop **720**, and the desktop **720** can be brought to the "top" of a stack over other windows during a desktop or other orientation change. For example, as shown in FIG. **7**B, the device **100** transitions to an open state **320**. The device **100** transitions to a display configuration as shown in FIG. **6**C. As such, the touch sensitive display **110** has no windows associated with the touch sensitive display **110**. Therefore, the desktop **720** is displayed for touch sensitive display **110**. Thus, the second stack **724** can includes on the desktop **720**, which, in embodiments, is a single desktop area, with desktop **720** under all the windows in both window stack **704** and window stack **724**. A logical data structure for managing the two window stacks or single window stack having two portions **704**, **724** may be as described in conjunction with FIG. **8**.

[0170] The arrangement of the window stack **704** after the device is opened is shown in FIGS. **7**C through **7**E. The window stack **704** is shown in three "elevation" views. In FIG. **7**C, the top of the window stack **704/724** is shown. Two adjacent sides of the window stack **704/724** are shown in FIGS. **7**D and **7**E. In this embodiment, the window stack **704/724** resembles a stack of bricks. The windows are stacked on each other. Looking from the top of the window stack **704/724** in FIG. **7**C, only the top most windows in the window stack **704/724** are seen in different portions of a composite display **728**. A desktop **786** or a window can occupy part or all of the composite display **728**. Fig. C shows a composite display **728**. The composite display **728** encompasses or includes the display area of all the touch sensitive displays **110**,**114**. The size of the composite display **728** can change based on the orientation of the device **100**. For example, the size of the composite display **728** of the device **100** when in the closed state may only include the area for one of the touch sensitive displays **110** or **114**, as shown in FIG. **7**A. The composite display **728** expands to include both touch sensitive displays **110**,**114** when the device **100** is opened, as shown in FIGS. **7**B and **7**C. Some windows or display objects are associated with the composite display **728** and also change dimensions when the composite display **728** changes dimensions. In embodiments, one such display object may be the desktop **720**, which can change dimensions upon a device **100** open to fill the composite display **728**.

[0171] In the embodiment shown, the desktop **720** is the lowest display object, window, or "brick" in the window stack **704/724**. Thereupon, window **1 708**, window **2 712**, and window **3 716** are layered. Window **1 708**, window **2 712**, and window **3 716** only occupy a portion of the composite display **728**. Thus, another part of the stack **724** includes only the desktop **718**. Only the top window or display object in any portion of the composite display **728** is actually rendered and displayed. Thus, as shown in the top view in FIG. **7**C, window **1 708** and the desktop **718** are displayed by being at the top of the different portions of the window stack **704/724**. A window can be dimensioned to occupy only a portion of the composite display **728** to "reveal" windows lower in the window stack **704**. For example, the desktop **718** is lower in the stack that window **1 708**, window **2 712**, and window **3 716** but is still displayed. This arrangement of windows and the desktop occurs when the device is opened when displaying top window in the window stack **704**.

[0172] Where and how the desktop **718** is positioned within the stack can be a function of the orientation of the device **100**, the context of what programs, functions, software, etc. are being executed on the device **100**, how the stack is positioned when the device **100** is opened, etc. A logical data structure associated with the desktop **718** or the other windows may not change when the device **100** opens, but the logical data structures can determine how to display the windows and the desktop. When user interface or other events or tasks change the arrangement of the stack, the logical structures of the windows or desktop(s) can be changed to reflect the change in arrangement.

[0173] Another embodiment of a window stack arrangement **724**, which has changed because a device has opened from a different configuration from FIGS. **7A-7C**, is shown in FIGS. **7F** through **7J**. In the embodiment shown in FIG. **7F**, the device **100** is in a closed state **304** displaying window **1 708** in configuration **608**. Window **2 712** and window **3 716** are inactive windows. In the embodiment shown in FIG. **7F**, the device **100** includes a second stack **724** associated with a second touch sensitive display **110**, while the device **100** is in the closed state **304** in first portrait configuration **604**. Thus, the touch sensitive display **110** can have an associated window stack **724**. In other embodiments, there is a single window stack that encompasses all of a composite display. The device **100** can have a single stack for the composite display where the windows or display objects are sized to occupy a portion or all of the composite display. Thus, the stack **724** can represent a portion of a larger composite display stack, part of which is not shown because the device **100** is in a closed state **304**.

[0174] In FIG. **7G**, the two window stacks (or two portions of the window stack) **704**, **724** may have a different number of windows or display objects arranged in the respective stacks **704**, **724**. Further, the two window stacks **704**, **724** can also be identified differently and managed separately. As shown in FIG. **7F**, the second window stack **724** can be arranged in order from a first window **708** to a next window **712** to a last window **716** and finally to a desktop **720**, which, in embodiments, is at the "bottom" of the window stack **704**. In embodiments, the desktop **720** is not always at the "bottom" as application windows can be arranged in the window stack below the desktop **720**, and the desktop **720** can be brought to the "top" of a stack over other windows, such as during an orientation change. For example, as shown in FIG. **7G**, the device **100** transitions to an open state **320**. The device **100** transitions to a display configuration as shown in FIG. **6C**. As such, the touch sensitive display **114** has no windows associated with the touch sensitive display **114**. Therefore, the desktop **720** is displayed for touch sensitive display **114**. Thus, the second stack **704** can include the desktop **720**, which, in embodiments, is a single desktop area, with desktop **720** under all the windows in both window stack **704** and window stack **724**. A logical data structure for managing the two window stacks or single window stack, having two portions **704**, **724**, may be as described in conjunction with FIG. **8**.

[0175] The arrangement of the window stack **724** after the device is opened is shown in FIGS. **7H** through **7J**. The window stack **704** is shown in three "elevation" views. In FIG. **7H**, the top of the window stack **704/724** is shown. Two adjacent "sides" of the window stack **704/724** are shown in FIGS. **7I** and **7J**. In this embodiment, the window stack **704/724** resembles a stack of bricks. The windows are stacked on each other. Looking from the top of the window stack **704/724** in FIG. **7H**, only the top most windows/display objects in the window stack **704/724** are seen in different portions of a composite display **728**. A desktop **720** or a window can occupy part or all of the composite display **728**.

[0176] In the embodiment shown, the desktop **720** is the lowest display object, window, or "brick" in the window stack **704/724**. Thereupon, window **1 708**, window **2 712**, and window **3 716** are layered. Window **1 708**, window **2 712**, and window **3 716** only occupy a portion of the composite display **728**. Thus, another part of the stack **724** includes only the desktop **718**. Only the top window or display object in any portion of the composite display **728** is actually rendered and displayed. Thus, as shown in the top view in FIG. **7H**, window **1 708** and the desktop **720** are displayed by being at the top of the different portions of the window stack **704/724**. A window can be dimensioned to occupy only a portion of the composite display **728** to "reveal" windows lower in the window stack **704**. For example, the desktop **718** is lower in the stack that window **1 708**, window **2 712**, and window **3 716** but is still displayed. This arrangement of windows and the desktop occurs when the device is opened when displaying a top window in the window stack **704**.

[0177] Where and how the desktop **718** is positioned within the stack can be a function of the orientation of the device **100**, the context of what programs, functions, software, etc. are being executed on the device **100**, how the stack is positioned when the device **100** is opened, etc. A logical data structure associated with the desktop **718** or the other windows may not change when the device **100** opens, but the logical data structures can determine how to display the windows and the desktop. When user interface or other events or tasks change the arrangement of the stack, the logical structures of the windows or desktop(s) can be changed to reflect the change in arrangement.

[0178] A logical data structure **800** for managing the arrangement of windows or desktops in a window stack is shown in FIG. **8**. The logical data structure **800** can be any data structure used to store data whether an object, record, file, etc. The logical data structure **800** can be stored in any type of database or data storage system, regardless of protocol or standard. In embodiments, the logical data structure **800** includes one or more portions, fields, attributes, etc. that store data in a logical arrangement that allows for easy storage and retrieval of the information. Hereinafter, these one or more portions, fields, attributes, etc. shall be described simply as fields. The fields can store data for a window identifier **804**, dimensions **808**, a stack position identifier **812**, a display identifier **816**, and/or an active indicator **820**. Each window in a window stack can have an associated logical data structure **800**. While only a single logical data structure **800** is shown in FIG. **8**, there may be more or fewer logical data structures **800** used with a window stack (based on the number of windows or desktops in the stack), as represented by ellipses **824**. Further, there may be more or fewer fields than those shown in FIG. **8**, as represented by ellipses **828**.

[0179] A window identifier **804** can include any identifier (ID) that uniquely identifies the associated window or display object in relation to other windows or display objects in the window stack. The window identifier **804** can be a globally unique identifier (GUID), a numeric ID, an alphanumeric ID, or other type of identifier. In embodiments, the window identifier **804** can be one, two, or any number of digits based on the number of windows or display objects that can be opened. In alternative embodiments, the size of the window identifier **804** may change based on the number of windows or display objects opened. While the window or display object is open, the window identifier **804** may be static and remain unchanged.

[0180] Dimensions **808** can include dimensions for a window or display object in the composite display **704**. For example, the dimensions **808** can include coordinates for two or more corners of the window or display object or may include one coordinate and dimensions for the width and height of the window or display object. These dimensions **808** can delineate what portion of the composite display **704** the window or display object may occupy, which may the entire composite display **704** or only part of composite display **704**. For example, window **1 708** may have dimensions **880** that indicate that the window **1 708** will occupy only part of the display area for composite display **728**, as shown in FIGS. **7**C and **7**H. As windows or display objects are moved or inserted in the window stack, the dimensions **808** may change.

[0181] A stack position identifier **812** can be any identifier that can identify the position in the stack for the window or display object or may be inferred from the window's control record within a data structure, such as a list or a stack. The stack position identifier **812** can be a GUID, a numeric ID, an alphanumeric ID, or other type of identifier. Each window or display object can include a stack position identifier **812**. For example, as shown in FIG. **7**A, window **1 708** in stack **1 704** can have a stack position identifier **812** of 1 identifying that window **708** is the first window in the stack **704** and the active window. Similarly, window **2 712** can have a stack position identifier **812** of 2 representing that window **2 712** is the second window in the stack **704**. Thus, depending on the type of stack, the stack position identifier **812** can represent a window's or display object's location in the stack.

[0182] A display identifier **816** can identify that the window or display object is associated with a particular display, such as the first display **110**, the second display **114**, or the composite display **728** composed of both displays. While this display identifier **816** may not be needed for a multi-stack system, as shown in FIG. **7**A, the display identifier **816** can indicate whether a window or display object in the serial stack of FIG. **7**B is displayed on a particular display. Thus, the desktop **720** may have two portions shown in FIG. **7**C. The first portion may have a display identifier **816** for the first display **110** while the second portion may have a display identifier **816** for the second display **114**. However, in alternative embodiments, the desktop **720** may have a single display identifier **816** identifying the composite display **728**.

[0183] Similar to the display identifier **816**, an active indicator **820** may not be needed with the dual stack system of FIG. **7**A, as the window or display object in stack position **1** is active and displayed. In alternative embodiments, the active indicator **820** can indicate which window(s) in the stack is being displayed. Thus, window **1 708** may be shown and have an active indicator **820**. The active indicator **820** can be a simple flag or bit that represents that the window or display object is active or displayed.

[0184] An embodiment of a method **900** for changing a window stack is shown in FIG. **9**. While a general order for the steps of the method **900** is shown in FIG. **9**. Generally, the method **900** starts with a start operation **904** and ends with an end operation **924**. The method **900** can include more or fewer steps or can arrange the order of the steps differently than those shown in FIG. **9**. The method **900** can be executed as a set of computer-executable instructions executed by a computer system and encoded or stored on a computer readable medium. Hereinafter, the method **900** shall be explained with reference to the systems, components, modules, software, data structures, user interfaces, etc. described in conjunction with FIGS. **1-8**.

[0185] A multi-screen device **100** can receive an orientation change, as described in FIGS. **3**A-**3**B, which changes the device **100** from a closed state **304** to an open state **320**, in step **908**. The orientation change can be detected and signaled by a hardware input from Hall Effect sensors, a timer, etc. The orientation change can be received by the Task Management Module **540** and sent to the Multi-Display Management Module **524**. The Multi-Display Management Module **524** may interpret the change to changes the configuration of the display from a closed portrait display **604**, **608** to an open portrait display (as shown in FIG. **6**C) or from a landscape closed display **612**, **616** to an open landscape configuration (as shown in FIG. **6**F, as described in conjunction with FIGS. **6**A through **6**F. In embodiments, the Task Management Module **540** places the user interface interaction in the task stack **552** to be acted upon by the Multi-Display Management Module **524**. Further, the Task Management Module **540** waits for information from the Multi-Display Management Module **524** to send instructions to the Window Management Module **532** to create the window in the window stack **704**.

[0186] The Multi-Display Management Module **524**, upon receiving instruction from the Task Management Module **540**, determines whether to reveal the desktop **786**, in step **912**. In embodiments, the desktop **786** may be at the bottom of the window stack **704/724** before the device **100** is opened. However, the device **100** may be displaying the last window in the stack **704/724**. In other words, the top window in the stack **704/724** is being displayed and there are no windows in a position to fill the newly exposed display in the open state. For example, in FIG. **7**A, there are no windows to the "left" of window **1 708**. As such, the Multi-Display Management Module **524** needs to present the desktop **720** in display **110**, as shown in FIG. **7**B. As the desktop **720** generally is spread across the composite display **728**, the desktop **720** is always display on a device **100** open unless another window covers the desktop **720**. As there are no windows covering the desktop **720**, as shown in FIGS. **7**B and **7**F, the Multi-Display Management Module **524** determines to display the desktop **720** in the opened display.

[0187] In embodiments, the device state module **574** of the Multi-Display Management Module **524** may determine how the device is oriented or in what state the device is in, e.g., open, closed, portrait, etc. Further, the preferences module **572** and/or requirements module **580** may determine how the desktop **786** is to be displayed based on the preferences of the desktop **786**. The Display Configuration Module **568** may then use the input from the device state module **574**, preferences module **572**, and/or other framework components **520** to evaluate the current window stack **704/724**. The desktop **720** being shown in the new display typically does not affect other windows in the window stack **704/724** because the other windows do not move. However, the dimensions **808** of the desktop **720** can change as the desktop **720** is modified to fill the composite display **728**, which expanded upon the opening of the device **100**.

[0188] The visibility algorithm, in embodiments, determines for all portions of the composite display **728**, which windows/display objects are at the top of the stack **704/724**. For example, the visibility algorithm determines that, after the device **100** is opened, the desktop **786** is revealed in one portion of the stack **704**, as viewed in FIG. 7B or 7F. Further, window **1 708** is displayed in the other portion of the stack **704**, as viewed in FIG. 7B or 7F. Upon determining where to reveal the desktop **720**, the Display Configuration Module **568** can change the display dimensions **808**, display identifiers **816**, and/or stack position identifiers **812** for the desktop **720**. The Multi-Display Management Module **524** can then send the dimensions **808**, display identifiers **816**, and/or stack position identifiers **812** back to the Task Management Module **540**.

[0189] In embodiments, the Task Management Module **540** sends the dimensions **808**, display identifiers **816**, and/or stack position identifiers **812**, and/or other information and instructions to render the desktop **786** to the Window Management Module **532**. The Window Management Module **532** and the Task Management Module **540** can change the logical data structure(s) **800**, in step **920**. Both the Task Management Module **540** and the Window Management Module **532** may manage copies of the window stack **704/724**. These copies of the window stack **704/724** can be synchronized or kept similar through communications between the Window Management Module **532** and the Task Management Module **540**. Thus, the Window Management Module **532** and the Task Management Module **540**, based on the information determined by the Multi-Display Management Module **524**, can change dimensions **808**, display identifiers **816**, and/or stack position identifiers **812** for the desktop **720** and the one or more window(s). The logical data structure **800** may then be stored by both the Window Management Module **532** and the Task Management Module **540**. Further, the Window Management Module **532** and the Task Management Module **540** may thereinafter manage the window stack **704/724** and the logical data structure(s) **800**.

[0190] The exemplary systems and methods of this disclosure have been described in relation to a window stack associated with a multi-screen device. However, to avoid unnecessarily obscuring the present disclosure, the preceding description omits a number of known structures and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set forth to provide an understanding of the present disclosure. It should however be appreciated that the present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[0191] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of the system can be located remotely, at distant portions of a distributed network, such as a LAN and/or the Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, such as the software modules of FIGS. **5**A and **5**B, or collocated on a particular node of a distributed network, such as an analog and/or digital telecommunications network, a packet-switch network, or a circuit-switched network. It will be appreciated from the preceding description, and for reasons of computational efficiency, that the components of the system can be arranged at any location within a distributed network of components without affecting the operation of the system. For example, the various components can be located in a switch such as a PBX and media server, gateway, in one or more communications devices, at one or more users' premises, or some combination thereof. Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[0192] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed element(s) that is capable of supplying and/or communicating data to and from the connected elements. These wired or wireless links can also be secure links and may be capable of communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics, and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0193] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[0194] A number of variations and modifications of the disclosure can be used. It would be possible to provide for some features of the disclosure without providing others. For example in one alternative embodiment, the window stack **700** resembles a carousel or rolodex rather than a deck of cards. Thus, windows may circulate from one touch sensitive display **110** to another touch sensitive display **114**. For example, window may be push to the right and end up at the end of stack behind another window. If the stack continues to move to the right, eventually the window would appear on the second touch sensitive display even though the window was opened in the first touch sensitive display **110**. These movements and changes in the stacks may be managed using the methods and logical data structures discussed above. In another alternative embodiment, there may be other arrangement for the window stack.

[0195] In yet another embodiment, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit, a programmable logic device or gate array such as PLD, PLA, FPGA, PAL, special purpose computer, any comparable means, or the like. In general, any device(s) or means capable of implementing the methodology illustrated herein can be used to implement the various aspects of this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular, Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art. Some of these devices include processors (e.g., a single or multiple microprocessors), memory, nonvolatile storage, input devices, and output devices. Furthermore, alternative software implementations including, but not limited to, distributed processing or component/object distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[0196] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments that provide portable source code that can be used on a variety of computer or workstation platforms. Alternatively, the disclosed system may be implemented partially or fully in hardware using standard logic circuits or VLSI design. Whether software or hardware is used to implement the systems in accordance with this disclosure is dependent on the speed and/or efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[0197] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be implemented as program embedded on personal computer such as an applet, JAVA® or CGI script, as a resource residing on a server or computer workstation, as a routine embedded in a dedicated measurement system, system component, or the like. The system can also be implemented by physically incorporating the system and/or method into a software and/or hardware system.

[0198] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards and protocols. Other similar standards and protocols not mentioned herein are in existence and are considered to be included in the present disclosure. Moreover, the standards and protocols mentioned herein and other similar standards and protocols not mentioned herein are periodically superseded by faster or more effective equivalents having essentially the same functions. Such replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[0199] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various aspects, embodiments, configurations embodiments, subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations after understanding the present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted and/or described herein or in various aspects, embodiments, and/or configurations hereof, including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and\or reducing cost of implementation.

[0200] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an intention that the claims require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[0201] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the skill and knowledge of those in the art, after understanding the present disclosure. It is intended to obtain rights which include alternative aspects, embodiments, and/or configurations to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions, ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

1-20. (canceled)

21. A non-transitory computer readable medium, having stored thereon, computer-executable instructions executable by a processor, the computer-executable instructions causing the processor to execute a method for managing window stacks for a mobile device, the computer-executable instructions comprising:

instructions to create a composite display having a first portion and a second portion;

instructions to create a first window stack logically associated with the first portion of the composite display, wherein the first window stack is a data structure representing a logical arrangement of one or more of active windows, inactive windows, and a desktop executing on the mobile device;

instructions to display a first window of a first open application that is at a logical top position of the first window stack on the first portion of the composite display;

instructions to receive a change in the mobile device from a first state to a second state;

in response to the change to the second state, instructions to create a second window stack logically associated with the second portion of the composite display, wherein the second window stack is a second data structure representing a second logical arrangement of one or more active windows, inactive windows, and the desktop executing on the mobile device;

instructions to determine a second window or desktop associated with the second window stack to display on the second portion of the composite display, wherein the second window or desktop comprises a window or desktop in the second window stack that is active and that is at a logical top position of the second window stack; and

after determining the desktop is at the logical top position of the second window stack and should be displayed on the second portion of the composite display, instructions to display the desktop on the second portion of the composite display and the first window on the first portion of the composite display.

22. The non-transitory computer readable medium as defined in claim **21**, wherein each of the first and second window stacks include one or more of:

a window identifier adapted to identify each window and the desktop in relation to other windows and the desktop in one of the first and second window stacks; and

a window stack position identifier adapted to identify a logical position of each window or desktop in one of the first and second window stacks.

23. The non-transitory computer readable medium as defined in claim **21**, wherein the first window stack can have a different number of windows or desktops than the second window stack, and wherein a window can be moved from one of the first and second window stacks to the other of the first and second window stacks.

24. A device comprising:

a composite display having a first portion and a second portion;

a memory; and

a processor in communication with the memory and the composite display, the processor operable to:

create a first window stack logically associated with the first portion of the composite display, wherein the first window stack is a data structure representing a logical arrangement of one or more of active windows, inactive windows, and a desktop executing on the device;

display a first window that is at a logical top position of the first window stack on the first portion of the composite display;

receive a change in the device from a first state to a second state;

in response to the change to the second state, create a second window stack logically associated with the second portion of the composite display, wherein the second window stack is a second data structure representing a second logical arrangement of one or more of active windows, inactive windows, and the desktop executing on the device;

determine a second window or desktop associated with the second window stack to display on the second portion of the composite display;

determine that the second window is at a logical top position of the second window stack; and

display the first window on the first portion of the composite display and the second window on the second portion of the composite display.

25. The device as defined in claim **24**, wherein a first display comprises the first portion of the composite display and a second display comprises the second portion of the composite display.

26. The device as defined in claim **24**, wherein each of the first and second window stacks include one or more of:

a window identifier adapted to identify each window and the desktop in relation to other windows and the desktop in one of the first and second window stacks; and

a window stack position identifier adapted to identify a logical position of each window or desktop in one of the first and second window stacks.

27. The device as defined in claim **25**, wherein the first and second displays are on a common first screen.

28. The device as defined in claim **25**, wherein the first display is on a first screen, and wherein the second display is on a different second screen.

29. A method for presenting a display of window or desktops for a device, the method comprising:

creating a composite display having a first portion and a second portion;

creating a first window stack logically associated with the first portion of the composite display, wherein the first window stack is a data structure representing a logical arrangement of one or more of active windows, inactive windows, and a desktop executing on the device;

displaying a first window that is at a logical top position of the first window stack on the first portion of the composite display;

receiving a change for the device from a first state to a second state;

in response to the change to the second state, creating a second window stack logically associated with the second portion of the composite display, wherein the second window stack is a second data structure representing a second logical arrangement of one or more of active windows, inactive windows, and the desktop executing on the device;

instructions to determine a second window or desktop associated with the second window stack to display on the second portion of the composite display;

determining that the second window is at a logical top position of the second window stack; and

displaying the first window on the first portion of the composite display and the second window on the second portion of the composite display.

30. The method defined in claim **29**, wherein the desktop is at a logical bottom position of the second window stack and is inactive, wherein the second window stack includes two or more other windows that are inactive, and wherein the desktop and the two or more other windows are not displayed on the second portion of the composite display.

31. The method defined in claim **29**, wherein a first display comprises the first portion of the composite display and a second display comprises the second portion of the composite display.

32. The method defined in claim **31**, wherein the first display is on a first screen, and wherein the second display is on a different second screen.

33. The method defined in claim **31**, wherein the first and second displays are on a common first screen.

34. The non-transitory computer readable medium as defined in claim **21**, further comprising:

instructions to receive an input to move the first window from the first window stack to the second window stack;

instructions to modify the second window stack, wherein the first window is moved to an active position at the logical top position of the second window stack and the desktop is moved to an inactive position in the second window stack; and

instructions to display the first window on the second portion of the composite display.

35. The non-transitory computer readable medium as defined in claim **34**, further comprising:

in response to moving the first window to the second window stack, instructions to determine a third window or desktop to display on the first portion of the composite display; and

after determining that the desktop is at the logical top position of the first window stack and should be displayed on the first portion of the composite display, instructions to display the desktop on the first portion of the composite display.

36. The non-transitory computer readable medium as defined in claim **21**, wherein a first display comprises the first portion of the composite display and a second display comprises the second portion of the composite display.

37. The non-transitory computer readable medium as defined in claim **36**, wherein the first display is on a first screen, and wherein the second display is on a different second screen.

38. The non-transitory computer readable medium as defined in claim **36**, wherein the first and second displays are on a common first screen.

39. The method defined in claim **30**, further comprising:

receiving an input to move the second window from the second portion of the composite display to the first portion of the composite display;

modifying the first window stack to move the first window to an inactive position in the first window stack;

modifying the first window stack to move the second window to the logical top position of the first window stack; and

displaying the second window on the first portion of the composite display.

40. The method defined in claim **39**, further comprising:

in response to moving the second window to the first window stack, determining a third window or desktop to display on the second portion of the composite display; and

after determining that the third window is one of the tow or more other windows that is at the logical top position of the second window stack, instructions to display the third window on the second portion of the composite display.

* * * * *