**ABSTRACT**

Methods and devices for configuring and displaying user interface elements of a multi-display device relative to a device state and/or device orientation. More particularly, the device is equipped with one or more sensors that facilitate the detectability of the relationship of the primary screen to the secondary screen and the general orientation of the device. The method includes accepting a set of user preferences for individual user interface elements for various device states and/or device orientations. The method and device may configure individual windows, per user settings, relative to device state and/or user orientation of the device. The method and device may present, for example, a single set of reconfigurable and/or re-locatable user interface elements in different locations depending on device orientation and/or display mode. Further, the method and device may provide the user the ability to switch both display mode and user interface action simultaneously.
FIG. 2
<table>
<thead>
<tr>
<th>Portrait</th>
<th>Landscape</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Open</strong></td>
<td><strong>Open</strong></td>
</tr>
<tr>
<td>X</td>
<td>AT</td>
</tr>
<tr>
<td>HT</td>
<td>HAT</td>
</tr>
<tr>
<td>HT</td>
<td>HAT</td>
</tr>
<tr>
<td>HT</td>
<td>HAT</td>
</tr>
<tr>
<td>P</td>
<td>I</td>
</tr>
<tr>
<td><strong>Closed</strong></td>
<td><strong>Closed</strong></td>
</tr>
<tr>
<td>HT</td>
<td>HAT</td>
</tr>
<tr>
<td>X</td>
<td>HAT</td>
</tr>
<tr>
<td>HT</td>
<td>HAT</td>
</tr>
<tr>
<td>X</td>
<td>HAT</td>
</tr>
<tr>
<td>P</td>
<td>I</td>
</tr>
<tr>
<td><strong>Easel</strong></td>
<td><strong>Easel</strong></td>
</tr>
<tr>
<td>HT</td>
<td>HAT</td>
</tr>
<tr>
<td>HT</td>
<td>HAT</td>
</tr>
<tr>
<td>X</td>
<td>HAT</td>
</tr>
<tr>
<td>P</td>
<td>I</td>
</tr>
<tr>
<td><strong>Portrait Phone</strong></td>
<td><strong>Landscape Phone</strong></td>
</tr>
<tr>
<td>HT</td>
<td>HAT</td>
</tr>
<tr>
<td>X</td>
<td>HAT</td>
</tr>
<tr>
<td>HT</td>
<td>HAT</td>
</tr>
<tr>
<td>X</td>
<td>HAT</td>
</tr>
<tr>
<td>P</td>
<td>I</td>
</tr>
<tr>
<td><strong>Image/Video</strong></td>
<td><strong>Image/Video</strong></td>
</tr>
<tr>
<td>HT</td>
<td>HAT</td>
</tr>
<tr>
<td>HT</td>
<td>HAT</td>
</tr>
<tr>
<td>X</td>
<td>HAT</td>
</tr>
<tr>
<td>P</td>
<td>I</td>
</tr>
<tr>
<td><strong>Docked</strong></td>
<td><strong>Docking Signal</strong></td>
</tr>
<tr>
<td>HAT</td>
<td>HAT</td>
</tr>
<tr>
<td>HAT</td>
<td>HAT</td>
</tr>
<tr>
<td>AT</td>
<td>HAT</td>
</tr>
<tr>
<td>HT</td>
<td>HAT</td>
</tr>
<tr>
<td>HT</td>
<td>HAT</td>
</tr>
<tr>
<td>P</td>
<td>X</td>
</tr>
</tbody>
</table>

**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**
Multi-Display Management Module

- Display Configuration Module
- Preferences Module
- Device State Module
- Gesture Module
- Requirements Module
- Event Module
- Binding Module

FIG. 5B
FIG. 6F

LD1

LD2
User selects targeted function, service or application of interest

User sets parameters of selected function/service (e.g. size, shape, orientation, arrangement) for modes of interest

Device stores user preferences

End
Receive Device State Information

Receive user display preferences and/or application display requirements

Determine nominal display orientation for each screen

Present each display for each screen and reconfigure as required

Has user changed display preferences?

Has device state information changed?

End

FIG. 8
Start

Receive Device State Information

Receive display preferences and/or display requirements for each application from user and/or application

Determine if conflict exists for display orientation characteristics for any application?

YES

Resolve conflicts

Establish nominal display orientation characteristics for each display screen

NO

End

FIG. 9
RECONFIGURABLE USER INTERFACE ELEMENTS

CROSS REFERENCE TO RELATED APPLICATION

[0001] The present application claims the benefits of and priority, under 35 U.S.C. § 119(e), to U.S. Provisional Application Ser. No. 61/539,884, filed Sep. 27, 2011, entitled “MOBILE DEVICE.” The present application also cross-references U.S. patent application Ser. No. 13/484,951 filed May 31, 2012 entitled “ORIENTATION ARBITRATION.” 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 one embodiment of the invention, a method for displaying a user interface element is provided, the method comprising: providing a device having at least a first and second screens, wherein each of the first and second screens are configured to display at least one of a first display mode and a second display mode, wherein at least the first display mode is configured to display a first configuration of a user interface element and the second display mode is configured to display a second configuration of a user interface element; receiving a first input from at least one of a user and an application, wherein the first input at least requests the first configuration of the user interface element for the first display mode of the first screen and the second configuration of the user interface element for the second display mode of the first screen; receiving a second input from a sensor indicating an orientation of the device; determining a nominal display mode for the first screen in response to at least one of the first input and the second input, wherein the nominal display mode is at least one of the first display mode and the second display mode; determining a nominal configuration of the user interface element on the first screen in the nominal display mode in response to the first screen and the first input, wherein the configuration of the user interface element is at least one of the first configuration and second configuration; and displaying the nominal configuration of the user interface element in the nominal display mode on the first screen of the device.

[0007] In yet another embodiment of the invention, a non-transitory computer readable medium having stored thereon computer-executable instructions is provided, the computer executable instructions causing a processor of a device to execute a method for reconfiguring a user interface element, wherein the device has at least a first and second screens, wherein each of the first and second screens are configured to display at least in a portrait mode and a landscape mode, the computer-executable instructions comprising: instructions to receive a first input from at least one of a user and an application, wherein the first input requests a first configuration of the user interface element when a screen is in portrait mode and a second configuration of the user interface element when a screen is in landscape mode; instructions to receive a second
input from a sensor indicating an orientation of the device; instructions to determine a nominal display mode for the first screen in response to at least one of the first input and the second input, wherein the nominal display mode is at least one of portrait mode and landscape mode; and instructions to display the nominal configuration of the user interface element in the nominal display on the first screen of the device.

[0008] The present disclosure can provide a number of advantages depending on the particular aspect, embodiment, and/or configuration. Users of handheld computing devices configure their applications and desktops with configurable user interface elements to make access to their proposed functions and services convenient. Existing solutions lack intelligence to allow a user to configure individual user interface elements relative to device state and/or device orientation. Typically, existing devices limit the modification of user interface elements to modification of scale and size while preserving orientation and arrangement. As such, when an existing device is rotated from portrait to landscape mode (or vice versa), the set of user interface elements is arranged so as to preserve the spatial arrangement of the icons while modifying relative proportions to fit screen resolution and dimensions—the user is unable to reconfigure other traits of the user interface icons such as location. However, on a device with multiple display modes one static configuration for each user interface element may not be sufficient. Furthermore, a dual multi-display handheld computing device implies a new mode of use for the user interface elements beyond existing modes in which configurations for user interface elements are static. The disclosed device addresses and overcomes these challenges in that the device allows the user to set the configuration and location of user interface elements as a function of device state and device orientation. Allowing a user to reconfigure their interface elements such that the device is personalized and optimized for each display mode provides a novel method of increasing user satisfaction and increasing customization and personalization opportunities. The device is equipped with one or more sensors that facilitate the detectability of the relationship of the primary screen to the secondary screen, and the general orientation of the device, to enable the user to configure individual user interface elements relative to device state and/or device orientation. As such, the device may configure individual windows, per user settings, relative to device state and/or user orientation of the device. Furthermore, a single set of reconfigurable and/or re-locatable user interface elements may be located in different locations depending on the devices orientation and display mode. Additionally, reconfigurable user interface elements provide the user the ability to switch both display mode and user interface action simultaneously. These and other advantages will be apparent from the disclosure.

[0009] 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”, “A and/or B” or “A and/or B” means A alone, A alone, A and B together, A and C together, B and C together, or A, B and C together.

[0010] 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.

[0011] 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”.

[0012] 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 an EPROM, a FLASH-EPROM, a solid state medium like a memory card, any other memory chip or cartridge, a carrier wave as described hereinabove, 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. 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.

[0013] 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 are 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.

[0014] 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.

[0015] 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.

[0016] 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.

[0017] The term “composite 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.

[0018] 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.

[0019] 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.

[0020] 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.

[0021] 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.

[0022] 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.

[0023] 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 display.

[0024] 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.

[0025] 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.

[0026] 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 deline-
ate 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

[0027] FIG. 1A includes a first view of an embodiment of a multi-screen user device;
[0028] FIG. 1B includes a second view of an embodiment of a multi-screen user device;
[0029] FIG. 1C includes a third view of an embodiment of a multi-screen user device;
[0030] FIG. 1D includes a fourth view of an embodiment of a multi-screen user device;
[0031] FIG. 1E includes a fifth view of an embodiment of a multi-screen user device;
[0032] FIG. 1F includes a sixth view of an embodiment of a multi-screen user device;
[0033] FIG. 1G includes a seventh view of an embodiment of a multi-screen user device;
[0034] FIG. 1H includes an eighth view of an embodiment of a multi-screen user device;
[0035] FIG. 1I includes a ninth view of an embodiment of a multi-screen user device;
[0036] FIG. 1J includes a tenth view of an embodiment of a multi-screen user device;
[0037] FIG. 2 is a block diagram of an embodiment of the hardware of the device;
[0038] FIG. 3A is a block diagram of an embodiment of the state model for the device based on the device’s orientation and/or configuration;
[0039] FIG. 3B is a table of an embodiment of the state model for the device based on the device’s orientation and/or configuration;
[0040] FIG. 4A is a first representation of an embodiment of user gesture received at a device;
[0041] FIG. 4B is a second representation of an embodiment of user gesture received at a device;
[0042] FIG. 4C is a third representation of an embodiment of user gesture received at a device;
[0043] FIG. 4D is a fourth representation of an embodiment of user gesture received at a device;
[0044] FIG. 4E is a fifth representation of an embodiment of user gesture received at a device;
[0045] FIG. 4F is a sixth representation of an embodiment of user gesture received at a device;
[0046] FIG. 4G is a seventh representation of an embodiment of user gesture received at a device;
[0047] FIG. 4H is an eighth representation of an embodiment of user gesture received at a device;
[0048] FIG. 5A is a block diagram of an embodiment of the device software and/or firmware;
[0049] FIG. 5B is a second block diagram of an embodiment of the device software and/or firmware;
[0050] FIG. 6A is a first representation of an embodiment of a device configuration generated in response to the device state;
[0051] FIG. 6B is a second representation of an embodiment of a device configuration generated in response to the device state;
[0052] FIG. 6C is a third representation of an embodiment of a device configuration generated in response to the device state;
[0053] FIG. 6D is a fourth representation of an embodiment of a device configuration generated in response to the device state;
[0054] FIG. 6E is a fifth representation of an embodiment of a device configuration generated in response to the device state;
[0055] FIG. 6F is a sixth representation of an embodiment of a device configuration generated in response to the device state;
[0056] FIG. 6G is a seventh representation of an embodiment of a device configuration generated in response to the device state;
[0057] FIG. 6H is an eighth representation of an embodiment of a device configuration generated in response to the device state;
[0058] FIG. 6I is a ninth representation of an embodiment of a device configuration generated in response to the device state;
[0059] FIG. 6J is a tenth representation of an embodiment of a device configuration generated in response to the device state;
[0060] FIG. 7 is a flow diagram of an embodiment of a method for receiving user reconfigurable preferences for user interface elements;
[0061] FIG. 8 is a flow diagram of an embodiment of a method for creating, displaying and reconfiguring multiple display screens upon receipt of user preferences for user interface elements;
[0062] FIG. 9 is a flow diagram of an embodiment of a method for creating, displaying and reconfiguring multiple display screens upon receipt of user preferences for user interface elements and potential conflict with application display requirements; and
[0063] FIG. 10 illustrates a representation of reconfigurable user interface elements upon re-orientation of the device.

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

[0065] 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.

[0066] Mechanical Features:

[0067] FIGS. 1A-1J 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.

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 112a and 116a have been configured to receive a “back” input indicating that a user would like to view information previously displayed. Areas 112b and 116b have been configured to receive a “menu” input indicating that the user would like to view options from a menu. Areas 112c and 116c 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 112a-c and 116a-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 112a-C and 116a-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.

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.

The two screens 104 and 108 are connected together with a hinge 128, shown clearly in FIG. 1C (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.

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.

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.

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.

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).

In addition to the open position, device 100 may also have a "closed" position illustrated in Fig. 11. Again, position sensors 172A and 172B 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.

Device 100 can also be used in an "easel" position which is illustrated in Fig. 11. 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.

Fig. 11 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.

Transitional states are also possible. When the position sensors 172A 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 172A and B indicate that the screens are being opened or unfolded (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.

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. 1A-1J 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. 1A-1J and the description provided above are nonlimiting.

Hardware Features:

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.

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. One or more display controllers 216A, 216B 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 216a or 216b 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.

[0083] 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.

[0084] 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, SRAM, 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.

[0085] 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.

[0086] 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.

[0087] 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.

[0088] 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 talker or speaker phone operations. 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. 1A through 1J. 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.

[0089] 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.

[0090] 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, for example a Hall effect sensor, 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.

[0091] 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.

[0092] Device State:

[0093] FIGS. 3A and 3B 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.

[0094] As illustrated in FIG. 3A, there are twelve exemplary “physical” states: closed 304, transition 308 (or opening transitional state), eased 312, modified eased 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, respectively.

[0095] 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. 1H). 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.

[0096] 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 viewfinder 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.

[0097] In state 308, a transition state from the closed state 304 to the semi-open state or eased 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 eased 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 eased-like configuration on a surface. In state 316, known as the modified eased 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 eased state 312, with the difference being one of the primary screen 104 or the secondary screen 108 are placed on a surface as shown.

[0098] 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 landscape state 348 the primary screen 104 and the secondary screen 108 are generally in a landscape-like orientation.

[0099] 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.

[0100] 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. FIG. 3B 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. 3B, 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. In FIG. 3B, 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.

[0101] 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 eased state—shown in bold—“HAT.” For this exemplary transition from the portrait open to the landscape eased 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. 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 100s, accessories, peripherals, smart docks, or the like.

[0102] User Interaction:

[0103] FIGS. 4A through 4H depict various graphical representations of gesture inputs that may be recognized by the screen 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.

[0104] With reference to FIGS. 4A-4H, 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 432b than the border 432a 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 a longer period of time 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.

[0105] With reference to FIG. 4C, 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.

[0106] With reference to FIG. 4D, 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.

[0107] With reference to FIG. 4E, 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 428b to the screen 104,108 by, for example, a second digit. The first and second contacts 428a,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 428a is held for a first amount of time, as represented by the border 432a, and the second contact 428b is held for a second amount of time, as represented by the border 432b. The first and second amounts of time are generally substantially the same, and the first and second contacts 428a, b generally occur substantially simultaneously. The first and second contacts 428a, b generally also include corresponding first and second contact movements 436a, b, respectively. The first and second contact movements 436a, b are generally in opposing directions. Stated another way, the first contact movement 436a is towards the second contact 436b, and the second contact movement 436b is towards the first contact 436a. More simply stated, the pinch gesture 408 may be accomplished by a user’s digits touching the screen 104,108 in a pinching motion.

[0108] With reference to FIG. 4F, a spread gesture 410 on the screen 104,108 is depicted. The spread gesture 410 may be initiated by a first contact 428a to the screen 104,108 by, for example, a first digit and a second contact 428b to the screen 104,108 by, for example, a second digit. The first and second contacts 428a, 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 428a is held for a first amount of time, as represented by the border 432a, and the second contact 428b is held for a second amount of time, as represented by the border 432b. The first and second amounts of time are generally substantially the same, and the first and second contacts 428a, b generally occur substantially simultaneously. The first and second contacts 428a, b generally also include corresponding first and second contact movements 436a, b, respectively. The first and second contact movements 436a, b are generally in a common direction. Stated another way, the first and second contact movements 436a, b are away from the first and second contacts 428a, 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.

[0109] The above gestures may be combined in any manner, such as those shown by FIGS. 4G and 4H, to produce a determined functional result. For example, in FIG. 4G a tap gesture 420 is combined with a drag or flick gesture 412 in a direction away from the tap gesture 420. In FIG. 4H, a tap gesture 420 is combined with a drag or flick gesture 412 in a direction towards the tap gesture 420.

[0110] 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.

[0111] A tap, when received by an on 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.

[0112] 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 increase 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).

[0113] The combined gestures of FIG. 4G, 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. 4H, 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.

[0114] Firmware and Software:

[0115] 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 and/or 564B 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™, etc. The OS 516 is operable to provide functionality to the phone by executing one or more operations, as described herein.

[0116] 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.

[0117] 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.

[0118] 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. 51. 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. 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.

[0119] 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 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.

[0120] 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 interaction 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.

[0121] 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 960x800, 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, 400), 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.

[0122] 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.

[0123] 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 500. 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. 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 sub-tasks 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 life-cycle 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.

[0124] 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.

[0125] 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.

[0126] 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 Store 564 in the Application Store 560 that can be provided to the Window Management Module 556 for rendering.

[0127] 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 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.

[0128] An embodiment of the MDM module 524 is shown in FIG. 5B. 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. 6A-6F. 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.

[0129] 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. 6A-6F.

[0130] 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. 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. 3A and 3B. 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. 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 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.

[0131] Further, the Gesture Module 576 can receive task information either from the Task Manager Module 540 or the Input Management Module 536. The gestures may be as defined in conjunction with FIGS. 4A through 4I. 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.

[0132] 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.

[0133] 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.

[0134] 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 with 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 condition changes associated with events, gestures, state changes, application preferences or requirements, etc.

[0135] User Interface Configurations:

[0136] With reference now to FIGS. 6A-J, various types of output configurations made possible by the device 100 will be described hereinafter.

[0137] FIGS. 6A and 6B depict two different output configurations of the device 100 being in a first state. Specifically, FIG. 6A 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.

[0138] FIG. 6B 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.

[0139] 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. An alternative output configuration may be accommodated by the device 100 being in a second state. Specifically, FIG. 6C 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.

[0140] FIGS. 6D and 6E depict two further output configurations of the device 100 being in a third state. Specifically, FIG. 6D 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.

[0141] FIG. 6E 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.

[0142] FIG. 6F 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
FIGS. 6G and 6H 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. 6G shows that a first easel output configuration 618 may be displayed on the touch sensitive display 110. FIG. 6H 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.

FIG. 6I 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.

FIG. 6J depicts another output configuration that may be accommodated while the device 100 is in an open landscape state 340. 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.

The device 100 may reconfigure user interface elements, as will be described in FIGS. 7-10. More specifically, the device 100 may allow a user to configure individual user interface elements relative to device state and/or device orientation. The device may allow the user to reconfigure their interface elements such that the device is personalized and optimized for each display mode, thereby increasing customization and personalization opportunities. The device is equipped with one or more sensors that facilitate the detectability of the relationship of the primary screen to the secondary screen and the general orientation of the device. Furthermore, the device may allow configuring of a single set of reconfigurable and/or re-locatable user interface elements to be located in different locations depending on the device's orientation and display mode. Additionally, reconfigurable user interface elements may provide the user the ability to switch both display mode and user interface action simultaneously. The device and method may allow the creating, displaying and reconfiguring of multiple display screens upon receipt of user preferences for user interface elements and with respect to potential conflict with application display requirements.
bination of traits of the targeted user interface element, to comprise color, font, size, aspect ratio, location, brightness and other traits known to those skilled in the art. Additionally, the user may request preferences for one or more user interfaces as a function of the display mode, for example as a function of single screen, dual screen dual function, dual screen single function and extended screen modes. In one embodiment, the device 100 automatically adjusts scale and/or size for all user interface elements upon change from portrait to landscape or vice versa. In another embodiment, the device does not automatically adjust size and/or scale of user interface elements upon change from landscape to portrait. Steps 708 and 712 may be repeated one or more times to allow the user to target and set a plurality of characteristics for a given application, function or service, wherein characteristics comprise size, shape, orientation, functionality, location and relative arrangement among user interface elements.

In step 716, the device 100 stores the set of targeted user preferences input by the user in step 716 as targeted in step 708. Steps 708, 712 and 716 may be repeated one or more times to allow the user to target and store a plurality of characteristics for a given application, function or service. The method 700 ends at step 732.

In step 716, the device 100 stores the set of targeted user preferences input by the user in step 716 as targeted in step 708. Steps 708, 712 and 716 may be repeated one or more times to allow the user to target and store a plurality of characteristics for a given application, function or service. The method 700 ends at step 732.

[0150] An embodiment of a method 800 for creating, displaying and reconfiguring multiple display screens upon receipt of user preferences for user interface elements is shown in FIG. 8. A general order for the steps of the method 800 is shown in FIG. 8. Hereinafter, the method 800 shall be explained with reference to the systems, components, modules, software, data structures, user interfaces, etc. described in conjunction with FIGS. 1-8. Generally, the method 800 starts with a start operation 804 and ends with an end operation 832. The method 800 can include more or fewer steps or can arrange the order of the steps differently than those shown in FIG. 8.

[0152] Generally, the method 800 allows the device 100 to reconfigure the user interface of the device 100 by accepting a set of user preferences for individual user interface elements for various device states and/or device orientations. The method 800 may modify individual windows, per user settings, relative to device state and/or user orientation of the device. The method 800 may present, for example, a single set of reconfigurable and/or re-locatable user interface elements in different locations depending on device orientation and/or display mode. Further, the method 800 may provide the user the ability to switch both display mode and user interface action simultaneously. The method 800 can be executed as a set of computer-executable instructions executed by a computer system and encoded or stored on a computer readable medium. In embodiments, the method 800 is implemented by, but is not limited to, the Multi-Display Management (MDM) module 524, a Window Management module 532, an Input Management module 536, a Task Management module 540, an Application Model Manager 542, and/or a Display Controller 544.

[0153] A multi-screen device 100 can receive device state information in step 808. As discussed previously, the position sensor 172 can provide a signal indicating the position of the display screens 104 and 108 relative to one another. Broadly, the position sensor 172 can provide a signal as to the device 100 oriented in upright (i.e. vertical) configurations or sideways (i.e. horizontal) configurations. Furthermore, the state information can determine if there has been a configuration change to the device 100. The configuration change may be an event (see FIG. 3A and 3B) triggered by one or more signals from one or more hardware sensor 172, 176, etc. For example, if the device 100 is changed from upright or vertical configurations to sideways or horizontal configurations, Hall effect sensors 172 may indicate that a display configuration change should be considered. As previously discussed, for example with respect to FIGS. 5A-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. 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. The Device State Module 574 is operable to determine or receive the state of the device. As such, in embodiments step 808 may be implemented by the Device State Module 574.

[0154] In step 812, user display preferences (such as display orientation characteristics, e.g. display mode such as portrait or landscape) and/or application display requirements for each display screen and/or application are received. For example, as discussed above with regard to FIG. 7 and method 700, the user may identify her email application as an application in which she wishes specific functions to be located in specific relative positions depending on whether the device is oriented in portrait rather than landscape mode. The user may further identify a service, such as phone service, in which she also targets or identities as one in which one or more functions or icons or displays vary as a function of device orientation and/or configuration and/or state. These preferences are stored in the device 100 (as discussed in regards, for example, to step 716 of method 700). In embodiments, the user input for display preferences are input by the Input Management module 536. The Input Management module 536 receives the events and logically stores the events in an event buffer 556. Similarly, display preferences (and in some situations display requirements), for an application are received in step 812. In embodiments, 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 portrait or a landscape display. The Preferences Module 572 can determine an application’s display preference (e.g., by inspecting the application’s preference settings). In embodiments, the Requirements Module 580, similar to the Preferences Module 572, is operable to determine display preferences and/or display requirements for an application 564 or other component. An application can have a set display requirement that must be observed, i.e. 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. Additional disclosure regarding potential conflicts, and resolving actual conflicts, between user preferences and application preferences is provided with respect to FIG. 9.

[0155] In step 816, upon receipt of the display preferences and/or application display requirements for each display screen and/or application, the nominal display mode for each display is determined. The display mode for each display is nominal because, for example, the device 100 orientation
may have been changed so as to require reconsideration of the display mode for one or more displays, and/or a user may have input a new display preference or requirement. Also, a new and/or different application may have been launched for that display. In this step, display preferences, such as a user preference for primary screen to always display in portrait mode or a location preference for a particular user interface element of an application, are adjudicated or arbitrated against display preferences provided by an application and also display requirements that may be provided by a user and/or an application. For example, the adjudication logic may provide that an application's display requirement override any other preferences input by the user and/or the application. This would be the logic for the aforementioned requirement that the application Angry Birds must always display in landscape mode, i.e. that an application requirement would override a contrary preference by the user to display Angry Birds in portrait mode. As another example, the display adjudication logic may require that user preferences and/or requirements always override preferences of an application for a particular display window, yet not allow such user preferences and/or requirements to override an application display requirement. The adjudication logic may be reconfigurable. In step 820, the nominal display mode for each display screen is presented.

[0156]  In step 824, a query is made to determine if a user change in any user interface functions or display preferences has occurred. For example, the user may have input a new and different location preference for a user interface element of an application or requested the primary display to present in landscape mode from an initial preference for portrait mode. In embodiments, the Input Management module 536 is operable to manage such events that occur with the device 100. In embodiments, these events are received, stored, and forwarded to other modules by the Input Management module 536. If no user change in any display orientations has occurred, the method 800 proceeds to step 828. If a user change in any display orientations has occurred, the method proceeds to step 812.

[0157]  In step 828, a query is made to determine if device state information has changed. For example, whether the device 100 has moved from a generally vertical orientation to a generally horizontal orientation. If the device state information has changed, the method proceeds to step 808. If the device state information has not changed, the method proceeds to step 832.

[0158]  An embodiment of a method 900 for creating, displaying and reconfiguring multiple display screens upon receipt of user preferences for user interface elements and potential conflict with application display requirements is shown in FIG. 9. The method 900 generally provides more detail as to steps 812 and 816 of method 800. 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 928. 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. 1-9. In embodiments, the method 900 is implemented by, but is not limited to, the Multi-Display Management (MDM) module 524, a Window Management module 532, an Input Management module 536, a Task Management module 540, an Application Model Manager 542, and/or a Display Controller 544.

[0159]  A multi-screen device 100 can receive device state information in step 908. This step is as described in step 808 of FIG. 8. In step 912, user interface element preferences (such as the send email interface within an email application), and/or display requirements, for each application are received. For example, a preference request that the application “App9” be presented in a first display mode, such as a portrait mode P1, and another application “App10” be presented in a second display mode, such as landscape mode L1. The user could also input, for example, a preference request that user interface element UI17 of App9 be located in the upper right corner and be of square shape when in portrait mode, yet be located across the entire bottom in a rectangular shape when in landscape mode. In embodiments, the user input for application preferences are input by the Input Management module 536. The Input Management module 536 receives the events and logically stores the events in an event buffer 556. In embodiments, 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 portrait or landscape displays. The Preferences Module 572 can determine an application’s display preference (e.g., by inspecting the application’s preference settings). In embodiments, the Requirements Module 580, similar to the Preferences Module 572, is operable to determine display preferences and/or 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. This type of display requirement can be determined or received by, the Requirements Module 580.

[0160]  In step 916, upon receipt of the display preferences and/or requirements for each application (step 908) and the display preferences and/or requirements for each display screen (step 912), a query is made to determine if a conflict exists between the nominal display mode for each display. For example, the user may have requested that the user interface element UI17 of App9 be located in the upper right corner and be of square shape when in portrait mode, yet be located across the entire bottom as a rectangular shape when in landscape mode. However, App9 may only support UI17 as a square shape. As such, a conflict exists between the user request for presentation of UI17 as a rectangle when in landscape mode and the application requirement that UI17 cannot be presented as a square whether in portrait or in landscape mode. If a conflict exists, the method proceeds to step 920. If a conflict does not exist, the method proceeds to step 924. In step 920, if a conflict exists the conflict is resolved. For example, continuing the scenario described above, because the application (App9) only supports UI17 presented in a square shape, the application’s requirement for presentation of UI17 only as a square will override the user’s preference for a rectangular shape when in landscape mode and present UI17 in square shape in landscape (and in portrait) mode. However, the user’s location preferences for UI17, in both landscape and portrait mode, will be honored and implemented given there is no conflict with application App9 requirements with respect to UI17 location. Thus, the adjudication logic may provide that an application override some or all conflicting preferences input by the user and/or the application. As
Another example, the display adjudication logic may require that user preferences and/or requirements always override preferences of an application for a particular display window, yet not allow such user preferences and/or requirements to override an application display requirement. The adjudication logic may be reconfigurable. In step 924, the nominal display mode for each display screen is established. The method 900 ends at step 928. FIG. 10 illustrates a representation of reconfigurable user interface elements upon re-orientation of the device. Primary display 1013 is first set or configured in landscape mode L1, and secondary display 1011 is set or configured in landscape mode L1. Here, both primary display 1004 and secondary display 1011 are unlocked, that is, the device is configured such that each of the two displays rotate from a landscape presentation to a portrait presentation mode. Thus, upon rotation 1020, primary display 1013 changes from landscape mode L1 to portrait mode P1, and secondary display 1011 changes from landscape mode L1 to portrait mode P2. In the example of FIG. 10, the user has entered a preference that the user interface element portrayed as 1030 operates so as to be positioned at the lower edge of 1011 when its application is running or presented in landscape mode (L1), but that upon rotation or re-orientation to portrait mode (P2), the corresponding user interface element 1031 operates so as to be positioned at the upper right corner of 1011 (P2). As such, the user interface element of 1030 presents in one manner in landscape mode (L1) and in another in portrait mode (P2). Note that the user interface element 1030, 1031 has, in addition to changing location, changed with regard to shape and proportion (i.e. aspect ratio) as, for example, the element 1030 spans the entire width of 1011 when in landscape mode (L1) but spans only about half the width of portrait display 1011 (P2) as corresponding element 1031. This is as intended per preferences input by the user.

In contrast, a conventional device would typically position element 1031 along the right-side edge of portrait display 1011 (P2) in attempting to preserve user interface element characteristics between landscape and portrait modes, and would thus typically adjust user interface element scale and proportion to best match the scale and proportion as presented as 1030. Stated another way, a conventional device would generally maintain the position or location of user interface elements to the same relative quadrant of a display. Further note that in FIG. 10 user interface element 1030 in landscape display mode is centered at a significantly different location with respect to the landscape display than it is as element 1031 in the portrait display mode. Stated another way, element 1030 has a geometric vertical (i.e. up-down) center slightly above the bottom of display 1011 and has a geometric horizontal (i.e. left-right) center at the centerline of display 1011, whereas element 1031 has a vertical center slightly down from the upper edge of display 1011 and a horizontal center slightly to the left of the right-side edge of display 1011.

Another depiction of device 100 functionality with respect to the implementation of reconfigurable user preferences for user interface elements is shown by user interface elements 1040 and 1041 of FIG. 10. User interface element 1040, positioned in the lower right corner of landscape display 1013 (L1), maintains its relative position (i.e. bottom right display corner) to corresponding 1041 when display 1013 moves from landscape (L1) to portrait (P2), but in contrast to that of element 1030, 1031, does not reconfigure proportion beyond that required to best match scale and proportion of its representation in landscape mode.

In one embodiment, the device 100 may reconfigure user interface elements to a different quadrant of a screen or display, based on for example, user preference, application preference or requirement, and/or device orientation, to include movement from portrait to landscape display and vice versa. In one embodiment, the device 100 may reconfigure user interface elements to other than the same quadrant of a screen or display, based on for example, user preference, application preference or requirement, and/or device orientation, to include movement from portrait to landscape display and vice versa.

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.

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 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 collocated 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. 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.

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.

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.
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.

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.

In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on a processor computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the disclosure can be implemented as a program encoded 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.

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.

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. 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.

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.

What is claimed is:

1. A method for displaying a user interface element, comprising:

   providing a device having at least a first and second screens, wherein each of the first and second screens are configured to display at least one of a first display mode and a second display mode, wherein at least the first display mode is configured to display a first configuration of a user interface element and the second display mode is configured to display a second configuration of a user interface element;

   receiving a first input from at least one of a user and an application, wherein the first input at least requests the first configuration of the user interface element for the first display mode of the first screen and the second
configuration of the user interface element for the second display mode of the first screen; receiving a second input from a sensor indicating an orientation of the device; determining a nominal display mode for the first screen in response to at least one of the first input and the second input, wherein the nominal display mode is at least one of the first display mode and the second display mode; determining a nominal configuration of the user interface element on the first screen in the nominal display mode in response to the first screen and the first input, wherein the configuration of the user interface element is at least one of the first configuration and second configuration; and displaying the nominal configuration of the user interface element in the nominal display mode on the first screen of the device.

2. The method of claim 1, wherein the nominal display mode for the first screen is determined in response to the second input and the first display mode is displayed on the first screen.

3. The method of claim 1, wherein the nominal display mode for the first screen is determined in response to the second input and the second display mode is displayed on the first screen.

4. The method of claim 1, wherein the first display mode is a portrait mode and the second display mode is a landscape mode.

5. The method of claim 1, wherein the first input is from a user.

6. The method of claim 1, wherein the first input is from an application.

7. The method of claim 1, wherein the user interface element in the first configuration in the first display mode of the first screen is centered at a significantly different location in the first screen than in the second configuration in the second display mode of the first screen.

8. The method of claim 7, wherein the first display mode is a portrait mode and the second display mode is a landscape mode.

9. The method of claim 8, wherein the first input is from a user.

10. A device, comprising:
    a first screen, the first screen configured to display at least in a first display mode and a second display mode, wherein at least the first display mode is configured to display a first configuration of a user interface element and the second display mode is configured to display a second configuration of a user interface element;
    a second screen, the second screen configured to display at least in a first display mode and a second display mode, wherein at least the first display mode is configured to display a first configuration of a user interface element and the second display mode is configured to display a second configuration of a user interface element;
    a memory;
    a processor in communication with the memory, the first screen, and the second screen, the processor operable to:
    receive a second input from at least one of a user and an application, wherein the first input at least requests the first configuration of the user interface element for the first display mode of the first screen and the second configuration of the user interface element for the second display mode of the first screen;
    receive a second input from a sensor indicating an orientation of the device;
    determine a nominal display mode for the first screen in response to at least one of the first input and the second input, wherein the nominal display mode is at least one of the first display mode and the second display mode; and
    determine a nominal configuration of the user interface element on the first screen in the nominal display mode in response to the first screen and the first input, wherein the configuration of the user interface element is at least one of the first configuration and second configuration; and
    display the nominal configuration of the user interface element in the nominal display mode on the first screen of the device.

11. The device of claim 10, wherein the first display mode is a portrait mode and the second display mode is a landscape mode.

12. The device of claim 10, wherein the first input is from a user.

13. The device of claim 10, wherein the first input is from an application.

14. The device of claim 10, wherein the user interface element in the first configuration in the first display mode of the first screen is centered at a significantly different location in the first screen than in the second configuration in the second display mode of the first screen.

15. The device of claim 14, wherein the first display mode is a portrait mode and the second display mode is a landscape mode and the first input is from a user.

16. A non-transitory computer readable medium having stored thereon computer-executable instructions, the computer-executable instructions causing a processor of a device to execute a method for reconfiguring a user interface element, wherein the device has at least a first and second screens, wherein each of the first and second screens are configured to display at least in a portrait mode and a landscape mode, the computer-executable instructions comprising:
    instructions to receive a first input from at least one of a user and an application, wherein the first input requests a first configuration of the user interface element when a screen is in portrait mode and a second configuration of the user interface element when a screen is in landscape mode;
    instructions to receive a second input from a sensor indicating an orientation of the device;
    instructions to determine a nominal display mode for the first screen in response to at least one of the first input and the second input, wherein the nominal display mode is at least one of portrait mode and landscape mode; and
    instructions to display the nominal configuration of the user interface element in the nominal display on the first screen of the device.

17. The medium of claim 16, wherein the instructions to determine a nominal display mode for the first screen are determined by the second input and the portrait mode is displayed on the first screen.

18. The medium of claim 16, wherein the instructions to determine a nominal display mode for the first screen are determined by the second input and the landscape mode is displayed on the first screen.
19. The medium of claim 16, wherein the user interface element in the first configuration in the first display mode of the first screen is centered at a significantly different location in the first screen than in the second configuration in the second display mode of the first screen.

20. The medium of claim 19, wherein the first display mode is a portrait mode and the second display mode is a landscape mode and the first input is from a user.

* * * * *