

1620 may advantageously maintain a concept of touch screen pop-up focus distinct from the input focus maintained by the operating system for applications on the main display. Driver 1620 must use some reasonable rule to coordinate between multiple requests for auxiliary and pop-up images.

[0125] Driver 1620 may implement some icons, auxiliary screens, and pop-up screens entirely within the driver itself. The driver may include a mechanism for associating auxiliary screens with pre-existing applications that do not recognize API 1624. For example, if a known pre-existing presentation software application has the input focus, the driver could supply an auxiliary screen like that of FIG. 8A. Driver 1620 would interpret taps in the special icons of FIG. 8A by sending corresponding signals known to be recognized by the software application, such as simulated keystrokes or mouse clicks. Driver 1620 may also implement a mechanism to allow users to associate special icons or auxiliary screens with pre-existing applications. One such mechanism is a scripting language including commands to display images and icons and to change the interpretation of finger actions; scripts in such a language could be written and associated with a software application without modification to the application itself. Another such mechanism is a graphical control panel similar to the resource editors present in many interactive programming environments.

[0126] Driver 1620 may support a software control panel to allow the user to customize the operation of the touch screen. This control panel can include controls to choose, enable, disable, and rearrange the various icons on the default iconic screen. This control panel can also include controls to choose which touch screen activation mechanism(s) to use, and which auxiliary and pop-up images to allow access to the screen.

[0127] Driver 1620 may allow combinations of iconic, auxiliary and pop-up images on the touch screen. For example, driver 1620 could implement a concept of overlapping windows on the touch screen whereby an auxiliary screen could overlap part of but not all of the iconic screen image it replaces. One possible implementation of this approach is to use the existing display driver architecture of the operating system to manage the display of the touch screen. In the most general case, the touch screen would be viewed as a second display monitor by the operating system, and applications could open windows and dialogs on this display just as they would on the main display.

[0128] In an alternate embodiment, the touch screen would be treated distinctly from the main display. Applications would be forbidden from opening windows on the touch screen or operating the touch screen by means other than API 1624. This approach is less flexible but more appropriate, as the small size of the touch screen causes it to behave in the computer/human interface as a different class of device than main displays, even though the touch screen and main display might both be implemented by similar raster LCD technology.

[0129] While the present invention has been described with reference exemplary embodiments, it will be understood by those skilled in the art that various changes may be made and equivalents may be substituted for elements thereof without departing from the scope of the present invention. In addition, many modifications may be made to adapt a particular situation or material to the teachings

without departing from the essential scope thereof. Therefore, it is intended that the present invention not be limited to the particular embodiment disclosed as the best mode contemplated for carrying out this present invention, but that the present invention will include all embodiments falling within the scope of the appended claims.

What is claimed is:

1. In a computing device having a processor running an operating system and a display, a graphical user interface, comprising:

a touch screen; and

a driver coupling said touch screen to said operating system, said driver displaying a plurality of icons on said touch screen, at least one of said icons identifying at least one region on said touch screen that will cause an action on said touch screen and not on said display in response to contact by an object on said region.

2. In a computing device having a processor running an operating system and a display, a graphical user interface, comprising:

a touch screen, said touch screen supporting an unactivated state and an activated state; and

a driver coupling said touch screen to said operating system, said driver displaying a plurality of icons on said touch screen, at least one of said icons identifying at least one region on said touch screen that will cause an action on said touch screen and not on said display in response to contact by an object on said region.

3. In a computing device having a processor running an operating system and a display, a graphical user interface, comprising:

a touch screen; and

a driver coupling said touch screen to said operating system, said driver displaying a plurality of icons on said touch screen, at least one of said icons identifying at least one region on said touch screen that will cause an action on said touch screen and not on said display in response to contact by an object on said region;

wherein said driver includes an application programming interface that enables an application to display at least one image on said touch screen.

4. In a computing device having a processor running an operating system and a display, a graphical user interface, comprising:

a touch screen; and

a driver coupling said touch screen to said operating system, said driver displaying at least one icon identifying a region on said touch screen that will cause a first action on said touch screen and a second action different from said first action on said display in response to contact by an object on said region.

5. In a computing device having a processor running an operating system and a display, a graphical user interface, comprising:

a touch screen, said touch screen supporting an unactivated state and an activated state; and

a driver coupling said touch screen to said operating system, said driver displaying at least one icon identi-