November 2021

NAVIGATING USER INTERFACES ON A MOBILE COMPUTING DEVICE

Yun Sun Lee

Luke Brantingham

Kavinaath Murugan

Mike Bentz

Aaditya Kandibanda

NAVIGATING USER INTERFACES ON A MOBILE COMPUTING DEVICE

ABSTRACT

This publication describes systems and techniques for navigating user interfaces by moving or tapping a computing device (e.g., a smart phone, a computerized watch, computerized eyewear, laptop computer, or any other wearable or non-wearable mobile computing device). The computing device may include motion sensors (e.g., gyroscope, accelerometer, magnetometer, etc.) and/or pressure sensors (e.g., strain gauges, barometric sensors, switches, pressure plates, capacitive sensors, resistive sensors, etc.). Such sensors may detect tapping on the computing device and/or rotation of the computing device and provide the sensor data to the computing device. Using the data generated by one or more of these sensors, the computing device may determine that a user of the computing device is navigating a user interface displayed by the computing device and may move a cursor or other navigational element to different selectable user interface elements (e.g., buttons, links, text input fields, radio buttons, checkboxes, images, etc.) or may move the cursor within an editable region (e.g., a text edit region of the user interface).

DESCRIPTION

Currently, mobile computing devices, such as smartphones, are typically limited to a small number of ways of moving a cursor or navigational element in a user interface. For example, a user may wish to move a cursor by only a couple of characters within a text field instead of several words. However, due to the size of the text compared to the size of the user's finger, it is often challenging to accurately position the cursor within a word to edit a specific

character by holding down on a touch-enabled screen. The user may have to repeatedly use the tool until they are able to position the cursor in the desired location. Additionally, other tools that enable moving the cursor such as holding down on a spacebar and sliding back and forth on the key often have the same issues with the sensitivity of the tool rendering it difficult for precise placement of the cursor.

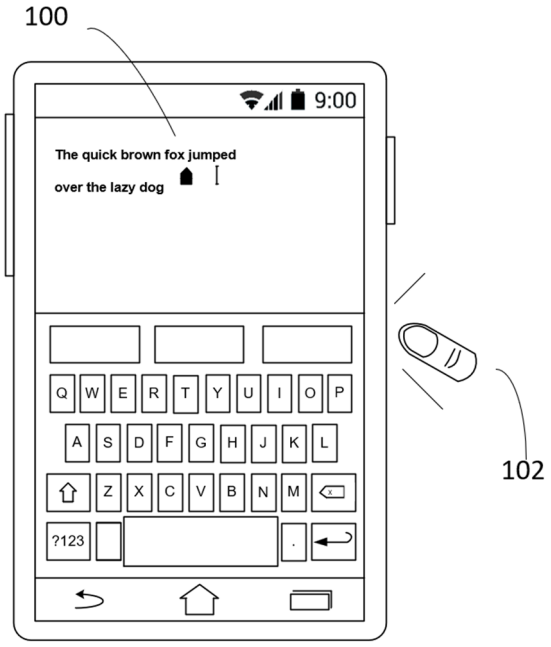

FIG. 1A

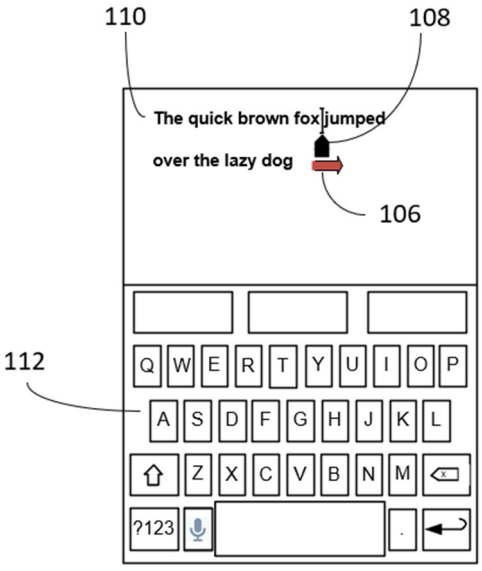

FIG. 1B

In accordance with the techniques described herein, a mobile computing device includes sensors that enable a user to change the location of a cursor within the graphical user interface (GUI) by tapping on the sides of the mobile computing device and/or rotating the mobile computing device. FIG. 1A illustrates an example operation of finger 102 tapping the side of mobile computing device 100 while a text box is active on mobile computing device 100.

Mobile computing device 100 detects a tap on mobile computing device 100 based on data generated by one or more sensors of mobile computing device 100. For example, an accelerometer may detect the acceleration of mobile computing device 100 from the impact of finger 102 striking the side of mobile computing device 100. As another example, mobile

computing device 100 may detect the tapping based on a change in capacitance along the edge of a capacitive touch display caused by the proximity of finger 102.

In some examples, mobile computing device 100 may apply a machine learning engine to filter the sensor data so as to reduce or eliminate false cursor shift inputs. As one example, mobile computing device 100 may apply the machine learning model to filter out detected motion that has an acceleration profile (e.g., acceleration rate, amplitude, vector of movement, etc.) different from an acceleration profile of an intentional tap. The machine learning model may be stored locally (i.e., on device) or may be stored remotely (e.g., on a computing system distinct from mobile computing device 100). In some instances, mobile computing device 100 may update the machine learning model based on user feedback (e.g., undoing a cursor change).

FIG. 1B illustrates an example graphical user interface 110 with the user using the navigation mode in accordance with FIG. 1A. Graphical user interface 112 includes text box 110 and cursor 108. Mobile computing device 100 determines, based on the data generated by the sensors, that a tap has occurred. Responsive to detecting the tap, mobile computing device 100 updates graphical user interface 112 and visually “shifts” the relative position of cursor 108 in the same direction as the side of mobile computing device 100 that was tapped as illustrated by arrow 106. Arrow 106 indicates the direction cursor 108 visually “shifts” in graphical user interface 110. In the example of FIG. 1B, the user has tapped the side of mobile computing device 100 while the navigational mode is enabled. Responsive to detecting one or more taps, mobile computing device 100 determines, based on the sensor data, whether the user intended to tap the side of mobile computing device 100. If mobile computing device 100 determines the user intended to tap mobile computing device 100, mobile computing device 100 responds and

cursor 108 to visually "shift" a number of characters within text box 110 corresponding to the number of taps.

Mobile computing device 100 may be configured with a user-activated navigation mode. While navigation mode is active, mobile computing device 100 may determine that the detected tapping is intended to be a navigation input. That is, while in navigation mode, mobile computing device 100, in response to detecting one or more taps, may update the graphical user interface to move the cursor based on the detected taps. In instances where navigation mode is not active, mobile computing device 100 may perform a different action in response to detecting the tapping or may perform no action in response to detecting the tapping.

In various instances, mobile computing device 100 may automatically update the graphical user interface with the cursor in a new position in response to detecting a tap without requiring the navigation mode being enabled. For example, mobile computing device 100 may determine, based on a location of a cursor within a graphical user interface being displayed by a display of mobile computing device 100 and a motion signature of the tapping motion that the user intends to move the cursor within the graphical user interface.

Mobile computing device 100 may determine which direction to move cursor 108 based on the side of mobile computing device 100 that was tapped. For example, if the user taps the left side of mobile computing device 100, mobile computing device 100 may cause cursor 108 to "shift" left in the text field or box. Conversely, if the user taps the right side of mobile computing device 100, as illustrated in FIG. 1A, mobile computing device 100 may cause cursor 108 to move to the right in the text field or box. In some examples, mobile computing device 100 may be configured substantially opposite such that if the user taps the left side of mobile computing

device 100, rather than moving cursor 108 to the left, mobile computing device 100 may move cursor 108 to the right.

In various instances, mobile computing device 100 may require a specified number of taps within a set time frame, e.g., three taps within 5 seconds prior to activation navigation mode and/or moving cursor 108. In such instances, if mobile computing device 100 does not determine that the threshold number of taps is satisfied, mobile computing device 100 may not move cursor 108.

Additionally, mobile computing device 100 may determine how far to move cursor 108 based on a rate at which the user taps mobile computing device 100. For example, if mobile computing device 100 receives data from sensors indicating that the user is rapidly tapping the side of mobile computing device 100, mobile computing device 100 may determine that the user intends to shift the cursor a set number of characters, to the beginning or end of a current word, a current line, a current sentence, a current paragraph, a current page, etc. such that cursor 108 may be visually “shifted” more characters than the number of distinct taps detected by mobile computing device 100.

In addition to using acceleration data from an accelerometer, mobile computing device 100 may use the relative location of taps on mobile computing device 100 to filter inputs. Mobile computing device 100 uses data from the capacitive touch screen to determine the relative location of the taps on mobile computing device 100. For example, if mobile computing device 100 receives data from an accelerometer indicating a tap of sufficient force occurred but data from the capacitive touch screen indicates that the input was in the center of the capacitive touch screen, then mobile computing device 100 may not effect a change to graphical user interface 112.

While described with respect to moving a cursor within a graphical user interface, the techniques described here may also be used as control inputs in mobile games. For example, mobile computing device 100 may determine that taps detected on the side of mobile computing device 100 correspond to additional button inputs. Mobile computing device 100 may differentiate which side of mobile computing device 100 and where along the side of mobile computing device 100 that the user has tapped and may perform specific actions within a mobile game or other application based on the location of the tap input(s). As another example, rather than moving a cursor, mobile computing device 100 may move a character, view, or other game element based on the change in orientation of mobile computing device 100.

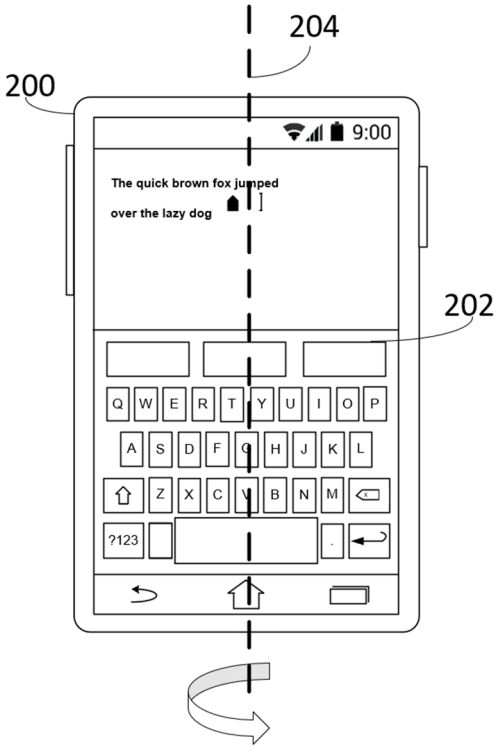

FIG. 2A

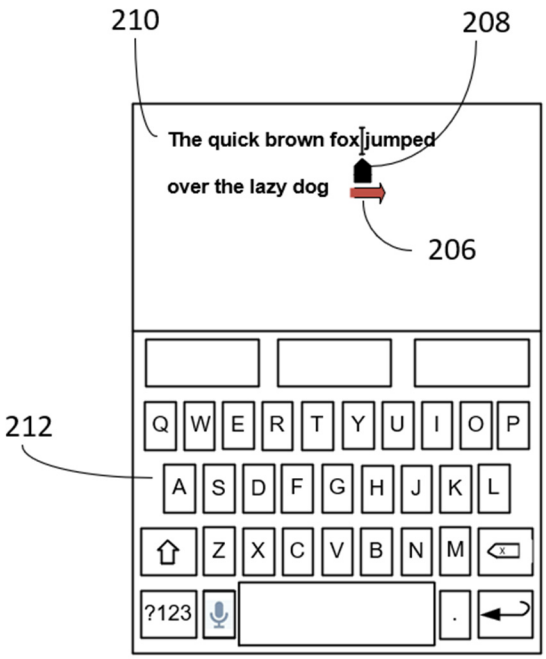

FIG. 2B

FIG. 2A illustrates an example mobile computing device 200, such as a smartphone or other mobile computing device, with touchscreen 202 and at least one gyroscopic sensor that detects rotation (e.g., around Y-axis 204). The user may hold mobile computing device 200

while using the virtual keyboard to type content, such as an email. The user may wish to move the cursor (e.g., to correct a typographical error), activate navigation mode, and rotate mobile computing device 200 about Y-axis 204. While the navigation mode is active, mobile computing device 200 detects a change in the orientation of mobile device 200 based on motion data, such as motion data generated by a gyroscope or an accelerometer. For example, when mobile computing device 200 detects a change in the Y-axis of mobile computing device 200, mobile computing device 200 moves the navigational elements accordingly (such as moving the cursor in the "left" or "right" direction on the display) through updating the position of the navigational elements in the graphical user interface. Mobile computing device 200 may also detect changes in the orientation along the X- and Z- axes using data generated by a gyroscope, an accelerometer, or other motion sensor in addition detecting changes in the Y-axis, with only a change in the Y-axis shown in the example FIG. 2A as Y-axis 204.

FIG. 2B illustrates graphical user interface 212 that includes text box 210 and cursor 208 where the user rotates mobile computing device 200 along the Y-axis while the navigation mode is active. Mobile computing device 200 detects the change in orientation based on data generated by the sensors and updates graphical user interface 212 with cursor 208. In the example shown in FIG. 2B, cursor 208 moves to the right as indicated by arrow 206. In other examples, mobile computing device 200 may move cursor 208 in the opposite direction (e.g., in response to detecting that the user rotated mobile computing device 200 in the opposite direction). Additionally, in instances where mobile computing device 200 detects rotation of mobile computing device 200 along the X-axis instead of the Y-axis, mobile computing device 200 may update graphical user interface 212 to move cursor 208 up or down in text box 210 based on the direction of the rotation about the X-axis.

The user may also control how far cursor 208 is shifted within graphical user interface 210 by varying the speed of the rotation. For example, if the user rapidly rotates mobile computing device 200 along Y-axis 204, mobile computing device 200 causes graphical user interface 210 to update with cursor 208 visually shifted several characters or entire words instead of a smaller number of characters. In an additional example, if the user more slowly rotates mobile computing device 200 along Y-axis 204, mobile computing device 200 causes graphical user interface 210 to update with cursor 208 visually shifted a smaller number of characters than if the user had more rapidly rotated mobile computing device 200.

It is noted that the techniques of this disclosure may be combined with any other suitable technique or combination of techniques. As one example, the techniques of this disclosure may be combined with the techniques described in U.S. Patent Application No. US20150169171A1. In another example, the techniques of this disclosure may be combined with the techniques described in U.S. Patent Application No. US20140152559A1. In yet another example, the techniques of this disclosure may be combined with the techniques described in U.S. Patent Application No. US20150268813A1.