December 2019

UTILIZING CONSISTENT USER INPUTS FOR SYSTEM-WIDE UNDO AND REDO OPERATIONS FOR COMPUTING DEVICES

Jeff Pierce

UTILIZING CONSISTENT USER INPUTS FOR SYSTEM-WIDE UNDO AND REDO OPERATIONS FOR COMPUTING DEVICES

ABSTRACT

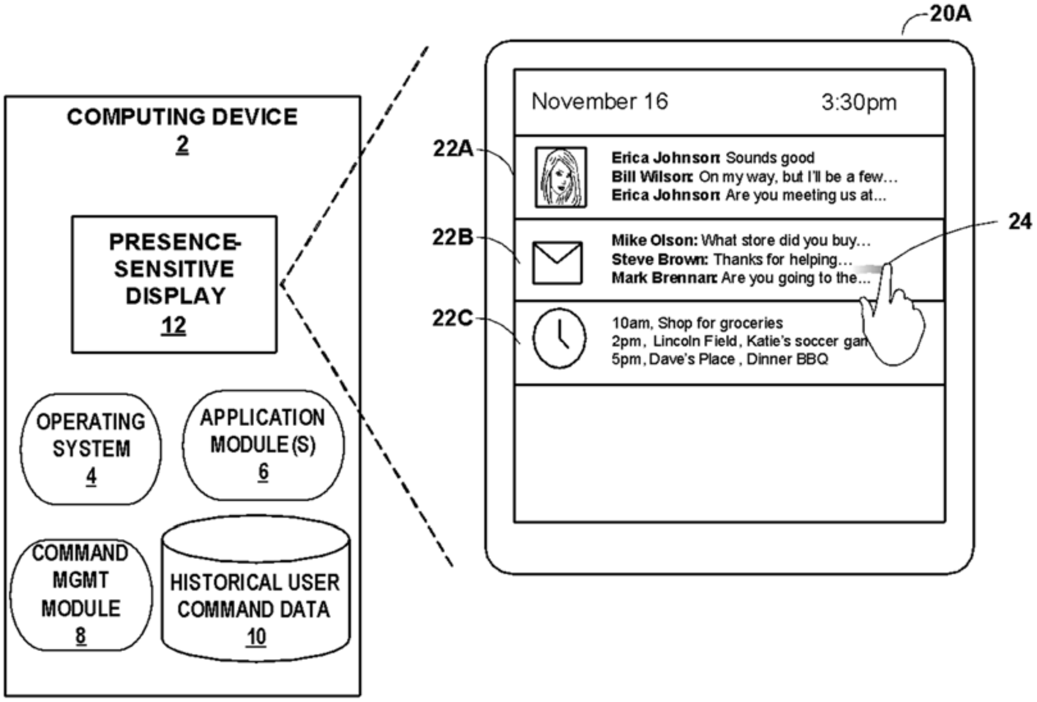
A computing device, such as a smartphone or tablet, provides a consistent mechanism across applications and/or the operating system by which a user may cause the computing device to perform editorial operations, such as an undo operation and/or a redo operation. The computing device includes a data store that stores information for a plurality of user commands (e.g., typing new text, deleting text, dismissing a message, etc.) for each application and the operating system of the computing device. When the computing device receives a user input or gesture indicative of a command to perform an editorial operation, the computing device queries the data store to determine the most recent user command performed by the currently active application or operating system. For example, the computing device may determine the most recent user command includes typing a sentence. In such examples, the computing device may undo the typing in response to receiving the user input indicative of an undo operation.

DESCRIPTION

Techniques are described that enable a computing device to perform editorial operations (e.g., undo and/or redo) in a consistent manner for each application and the operating system. FIGS. 1A–1C illustrate a block diagram of an example computing device 2 that includes operating system 4, one or more application modules 6, command management (mgmt) module (CMM) 8, historical user commands data 10, and presence-sensitive display (PSD) 12.

Examples of computing device 2 include a tablet computer, a smart phone, a countertop computer (e.g., a smart display), among others. Computing device 2 includes one or more processors. Examples of processors include, but are not limited to, digital signal processors

(DSPs), general purpose microprocessors, application specific integrated circuits (ASICs), field programmable logic arrays (FPGAs), or other equivalent integrated or discrete logic circuitry. In the example of FIG. 1, one or more processors execute the functionality of operating system 4, application modules 6, and CMM 8.



November 16 3:30pm

Erica Johnson: Sounds good
Bill Wilson: On my way, but I'll be a few…
Erica Johnson: Are you meeting us at...

Mike Olson: What store did you buy…
Steve Brown: Thanks for helping…
Mark Brennan: Are you going to the...

10am, Shop for groceries
2pm, Lincoln Field, Katie's soccer gan
5pm, Dave's Place, Dinner BBQ

FIG. 1A

PSD 12 may function as a respective input and/or output device for computing device 2. PSD 12 may be implemented using various technologies. For instance, PSD 12 may function as an input device using presence-sensitive input screens, such as resistive touchscreens, surface acoustic wave touchscreens, capacitive touchscreens, projective capacitance touchscreens, pressure sensitive screens, acoustic pulse recognition touchscreens, or another presence-sensitive display technology. PSD 12 may also function as an output (e.g., display) device using any one or more display devices, such as liquid crystal displays (LCD), dot matrix displays, light emitting diode (LED) displays, organic light-emitting diode (OLED) displays, e-ink, or similar monochrome or color displays capable of outputting visible information to a user of computing device 2.

In the example of FIG. 1A, operating system 4 of computing device 2 outputs graphical user interface (GUI) 20A via presence-sensitive display (PSD) 12. GUI 20A includes a plurality of different notifications 22A-22B (collectively, notifications 22).

A user may provide user input 24 in an effort to command computing device 2 to perform an operation. For instance, as shown in the example of FIG. 1A, the user may perform user input 24 as a swipe gesture by dragging a finger across PSD 12.

PSD 12 may detect user input 24 and output data indicative of user input 24 to CMM 8. CMM 8 may determine a user command associated with user input 24. Examples of user commands include typing new text, deleting text, dismissing messages (e.g., emails, text messages, etc.), dismissing notifications, deleting icons, amongst many others. As illustrated in FIG. 1A, CMM 8 determines that user input 24 is associated with a user command to dismiss notification 22B.

In one example, CMM 8 stores data indicative of the user command within historical user command data 10. Historical user command data 10 may include data indicating one or more of the most recent user commands for each application module 6 and operation system 4. For each application module 6 and operating system 4, historical user command data 10 includes data for a predetermined number of user commands (e.g., the *N* most recently executed user commands) or data for user commands performed over a pre-defined amount of time (e.g., user commands executed in the past *M* minutes). In some examples, historical user command data 10 includes data for application modules 6 while application modules 6 are executing and computing device 2 deletes historical user command data 10 for a given application module 6 when that application module 6 ceases execution. In some instances, historical user command data 10 includes data indicative of the most recent user command for a particular application module 6 even after the application module 6 ceases executing, which may enable the user to undo or redo the particular user command upon re-executing the particular application module 6.

Responsive to receiving user input 24 to dismiss notification 22B, operating system 4 outputs graphical user interface 20B that does not include dismissed notification 22B, as shown in FIG. 1B.

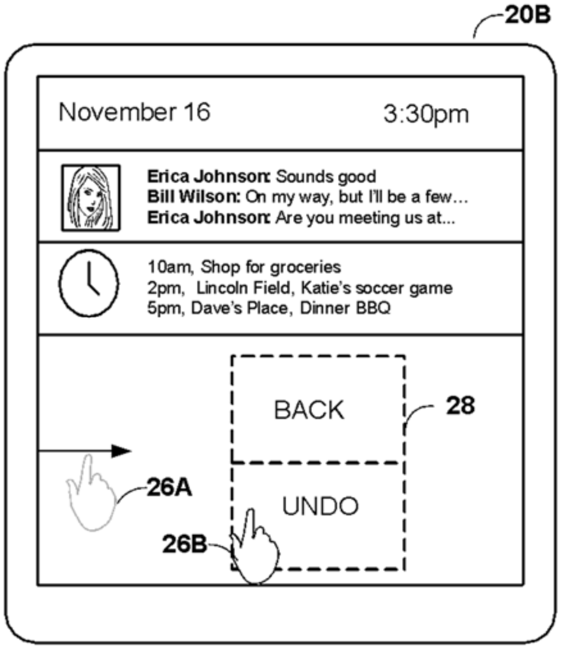

20B
November 16
3:30pm
Erica Johnson: Sounds good
Bill Wilson: On my way, but I'll be a few...
Erica Johnson: Are you meeting us at...
10am, Shop for groceries
2pm, Lincoln Field, Katie's soccer game
5pm, Dave's Place, Dinner BBQ
BACK
28
UNDO
26A
26B

FIG. 1B

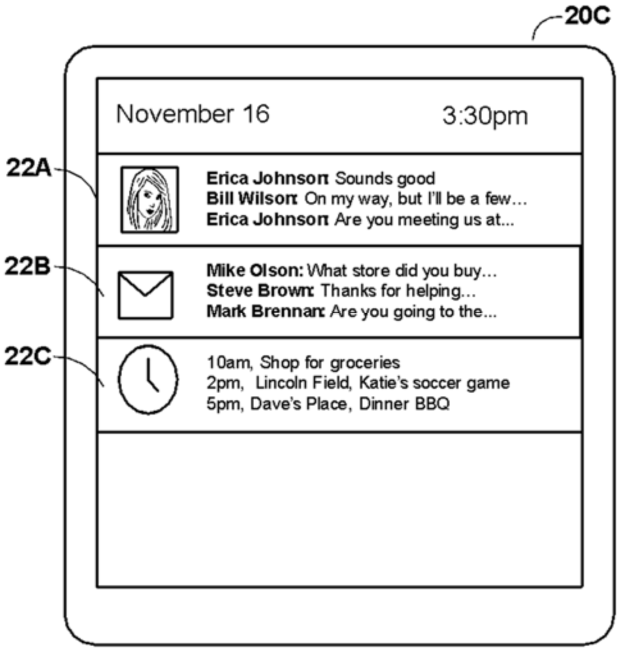

20C
November 16
3:30pm
22A
Erica Johnson: Sounds good
Bill Wilson: On my way, but I'll be a few...
Erica Johnson: Are you meeting us at...
22B
Mike Olson: What store did you buy...
Steve Brown: Thanks for helping...
Mark Brennan: Are you going to the...
22C
10am, Shop for groceries
2pm, Lincoln Field, Katie's soccer game
5pm, Dave's Place, Dinner BBQ

FIG. 1C

After dismissing notification 22B, computing device 4 may receive user input 26A. User input 26A may be associated with a navigational operation (e.g., navigate back), an editorial operation (e.g., undo and/or redo), or both. An undo operation may erase or negate the last change done to the document, reverting it to an older state. A redo operation may reverse an undo operation.

CMM 8 may determine whether to perform a navigation operation or an editorial operation in response to receiving user input 26A. In the example of FIG. 1B, CMM 8 determines whether to perform the navigation operation or the editorial operation based on an additional user input 26B. For example, CMM 8 may output menu 28 in response to receiving user input 26A. In some examples, menu 28 includes a prompt enabling the user to select between performing a navigation operation (e.g., back) or the most recent editorial operation (e.g., undo). In some examples, CMM 8 receives data indicative of user input 26B after outputting menu 28. In one example, as illustrated in FIG. 1B, CMM 8 determines that user input 26B includes a command to perform the editorial operation (e.g., undo the most recent user command) in response to determining that user selected the editorial operation "UNDO" from menu 28. While illustrated as separate or distinct user inputs, in some examples, user inputs 26A and 26B may be portions of a single (e.g., continuous) user input.

In some instances, CMM 8 may determine whether to perform an editorial operation or navigation operation without outputting menu 28. For instance, CMM 8 may determine whether to perform a navigation operation or an editorial operation based on the characteristics of user input 26A (e.g., direction, length, duration, and/or pressure of user input 26A). In one instance, CMM 8 determines to perform the editorial operation in response to determining that user input 26A travels in a first direction (e.g., horizontal) and that user input 26B travels in a second, different direction (e.g., vertical) without a pause between user inputs 26A and 26B. In other words, CMM 8 may determine to perform the editorial operation in response to detecting a continuous gesture having a first portion in substantially a first direction and a second portion in substantially a different direction. In another instance, CMM 8 determines to perform a navigation operation in response to detecting a user input (e.g., 26A) in substantially a single direction.

In some scenarios, CMM 8 determines whether to perform an editorial operation or navigation operation based on the characteristics of a single user input 26A. For example, CMM 8 may determine to perform an editorial operation in response to determining that user input 26A stopped at a location of PSD 12 for at least a threshold duration without releasing contact from PSD 12. In another example, CMM 8 may determine to perform a navigational operation (e.g., back) in response to determining the user input 26A did not stop for at least the threshold duration.

Responsive to determining that user input 26A and 26B represents an editorial operation, CMM 8 may identify the user command associated with the editorial operation. For example, CMM 8 may determine that GUI 20B was generated by operating system 4 and query historical user commands data 10 to identify the most recent user command performed by operating system 4. In the example of FIGS. 1A – 1C, CMM 8 determines that most recent user command includes a command to dismiss notification 22B. As illustrated in FIG. 1C, responsive to determining that user input 26 selects the editorial operation (e.g., undo), operating system 4 performs the editorial operation by outputting a graphical user interface 20C that includes the notification 22B.

FIGS. 2A and 2B illustrate additional examples of computing device 2 that performs editorial operations. In the example of FIG. 2A, an application module 6 (e.g., an image editing application) outputs GUI 30A. In one example, historical user command data 10 includes data indicating three commands associated with segments 32A-32C (collectively, segments 32) of GUI 30A. That is, the user of computing device 2 may draw segments 32 using three separate commands and application module 6 may store data for each command within historical user command data 10. In some examples, after receiving the commands to draw segments 32, the user may perform additional user inputs indicating commands to delete segments 32B and 32C. Thus, as illustrated in FIG. 2B, application module 6 may output GUI 30B that includes segment 32A but does not include segments 32B or 32C.

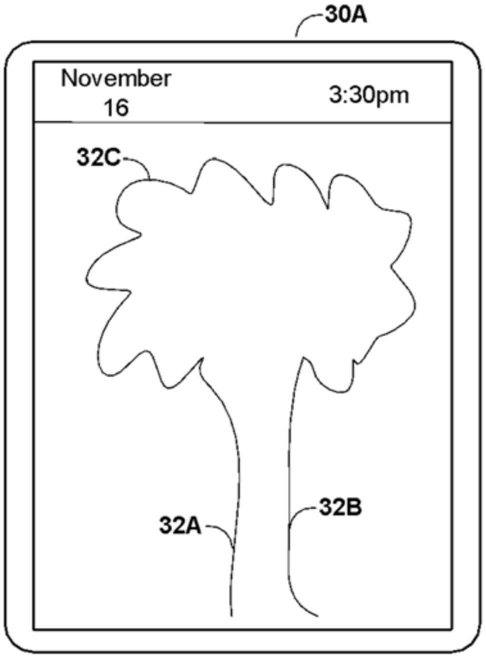


FIG. 2A

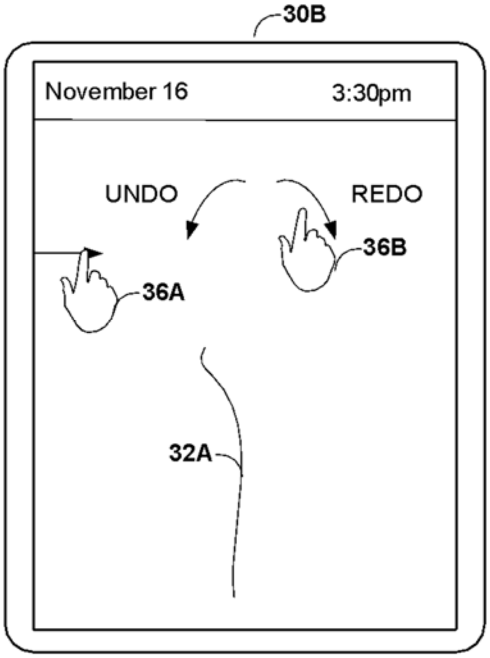


FIG. 2B

Responsive to receiving user input 36A associated with an editorial operation, application module 6 may update GUI 30B to include a menu to perform one or more editorial operations, such as “undo” and “redo”. A “redo” operation may reverse a prior undo operation. In one example, CMM 8 receives data indicative of user input 36B and determines that user input 36B is associated with the editorial operation to “redo”. For example, CMM 8 may query historical user command data 10 to identify the most recent undo operation that is to be reversed. In one example, CMM 8 determines the most recent undo operation was to delete segment 32B. In such examples, application module 6 performs the "redo" operation by reversing the deletion of segment 32B. In other words, application module 6 re-draws segment 32B.

In some examples, CMM 8 may perform multiple editorial operations based on user input 36B. For example, user input 36B may include a circular motion (e.g., clockwise) that draws multiple circles. CMM 8 may determine a quantity of editorial operations based on the quantity of circles drawn. For example, if user input 36B includes two clockwise circles, CMM 8 may perform two editorial operations associated with the user input 36B (e.g., two redo operations). In this way, the user may perform a single user input and cause CMM 8 to perform multiple editorial operations.

Associating the same type of gesture with editorial operations across a plurality of application modules 6 (e.g., for every application module) and operating system 4 may simplify the user experience by enabling the user of computing device 2 to perform a single type of user input or gesture to perform editorial operations (e.g., undo, redo) across different applications 6 and/or operating system 4.

While user inputs 26A and 36A are illustrated as swipe gestures from an edge of the respective GUIs, any other gesture (e.g., a double-tap, a tap with a certain number of fingers, etc.) may be associated with editorial operations.

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 Publication No. 2013/0227454 entitled "Method and Apparatus for Providing an Option to Undo a Delete Operation" by Thorsander et al.