



- (51) International Patent Classification:
G06F 9/44 (2006.01)
- (21) International Application Number:
PCT/IB2016/001078
- (22) International Filing Date:
28 June 2016 (28.06.2016)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
PCT/CN2015/082705 29 June 2015 (29.06.2015) CN
- (71) Applicant: ORANGE [FR/FR]; 78, rue Olivier de Serres, 75015 Paris (FR).
- (72) Inventors: HUANG, Menghuai; 23F, Tower A, Beijing Global Trade Center 36 North Third Ring Road East Dongcheng District, Beijing 100013 (CN). WAN, Jie; 23F, Tower A, Beijing Global Trade Center 36 North Third Ring Road East Dongcheng District, Beijing 100013 (CN).

YE, Nan; 23F, Tower A, Beijing Global Trade Center 36 North Third Ring Road East Dongcheng District, Beijing 100013 (CN). GUO, Zhihong; 23F, Tower A, Beijing Global Trade Center 36 North Third Ring Road East Dongcheng District, Beijing 100013 (CN). LIU, Xin-Lucienne; 23F, Tower A, Beijing Global Trade Center 36 North Third Ring Road East Dongcheng District, Beijing 100013 (CN).

(74) Agent: ORANGE; Frédéric THIEL, 38-40 rue du Général Leclerc, 92794 Issy Les Moulineaux Cedex 09 (FR).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC,

[Continued on next page]

(54) Title: METHOD FOR CONTROLLING THE EXECUTION OF A PROGRAM CONFIGURABLE INTO A DISABLED STATE AND ENABLED STATE

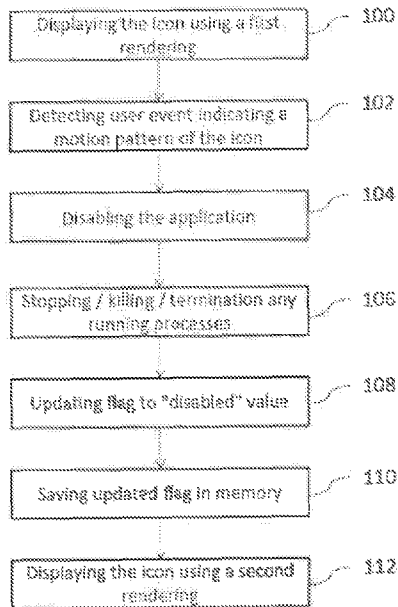


FIG. 2

(57) Abstract: The invention deals a method for controlling the execution of a program by a terminal (1) comprising a screen (12), wherein the method comprises the steps of: configuring the program into an enabled state, wherein at least one process of the program is allowed to be executed by the terminal (1), and a disabled state, wherein execution of any process of the program by the terminal (1) is forbidden, wherein the method further comprises the steps of: displaying (100) on the screen (12) an icon (G) for starting the program according to a first rendering if the program is configured in the enabled state, and displaying (114) said icon (G) according to a second rendering different from the first rendering if the program is configured in the disabled state.

WO 2017/001933 A1

SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN,
TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA,
GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

(84) Designated States (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE,

Published:

- *with international search report (Art. 21(3))*
- *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

Method for controlling the execution of a program configurable into a disabled state and enabled state

TECHNICAL FIELD

5 The invention deals with a method for controlling the execution of a program configurable into an enabled state and a disabled state.

STATE OF THE ART

10 Most applications installed in a terminal comprise different types of processes or services: foreground processes interacting with a user of the terminal through a graphical user interface, and background processes.

An application can typically started when a user of the terminal clicks on or taps on an icon of the application displayed on a screen of the terminal. This icon is commonly displayed in a specific menu showing icons of all applications installed in the terminal. This icon can also be displayed in a default menu displayed in the first place whenever the user 15 turns on the screen of the terminal and commonly referred to as “home screen”.

However, a background process may be secretly running even though the user is not currently using the application. A background process can be started whenever an operating system of the terminal is booted, or upon detecting some other device events. 20 Such a background process can perform actions on the device by checking the device location, the app updates or news from the internet which secretly consumes the terminal resources such as CPU, memory, consequently the battery and so on.

To block out any activities issued from an installed application for a period of time, the user could choose to uninstall the whole application, and reinstall it whenever needed 25 afterwards.

However, re-installing an application requires downloading it from a remote server.

Some existing operating systems allow the user to “disable” a pre-installed application. The operating system configures alternately the application into an enabled state, wherein at least one process of the program is allowed to be executed, and a 30 disabled state, wherein execution of any process of the program is forbidden.

Disabling an application, i.e. configuring the application in its disabled state, is performed by clicking on a graphical button displayed in a dedicated settings menu of the operating system. After disabling, all icons of the application are removed from the home menu and from the application menu.

As a consequence, in order to launch a disabled application, the user has to find, and enable the application from the settings menu, then go back to the home screen to launch it. It is not a simple and intuitive way for a user who'd like to temporally disable and enable any application installed on the device quickly.

5 Besides, the user has to access the settings menu in order to clearly know if an application is enabled or disabled.

SUMMARY OF THE INVENTION

10 The invention aims at warning the user that an application is currently enabled or disabled in a simpler way than the prior art.

For this purpose, the invention proposes a method for controlling the execution of a program by a terminal comprising a screen, wherein the method comprises the steps of: configuring the program into an enabled state, wherein at least one process of the program is allowed to be executed by the terminal, and a disabled state, wherein execution of any
15 process of the program by the terminal is forbidden, wherein the method further comprises the steps of: displaying on the screen an icon for starting the program according to a first rendering if the program is configured in the enabled state, and displaying said icon according to a second rendering different from the first rendering if the program is configured in the disabled state.

20 The process may further comprise the following features taken separately or combined whenever it is technically possible:

- rendering the icon using a first set of colors if the current state is the enabled state, and rendering the icon is rendered using a second set of colors if the current state is the disabled state, wherein the second set of colors is a subset of the first set;
- 25 • altering a default shape of the icon into a first altered shape if the current state is one of the disabled and enabled states, and altering a default shape of the icon into a second altered shape or leaving said default shape unchanged, if the current state is the other state;
- compositing the icon with a first indicator overlapping the icon, if the current state
30 is one of the disabled and enabled states, and compositing the icon with a second indicator overlapping the icon, wherein the second indicator is different from the first indicator, or leaving the icon free of indicator, if the current state is the other of the two states;
- configuring the program into the enabled state whenever detecting that the
35 program has been started using the icon;

- displaying a context menu whenever detecting that the icon was pressed on or tapped on, said menu showing a first item for directly starting the application and another item for changing the state of the program;
- 5 • detecting a user event indicating that the icon has been moved on the screen according to a first predetermined motion pattern, and changing the current state of the program in response to detecting the user event;
- using as a predetermined motion pattern a motion of the icon into a predetermined area of the screen;
- 10 • detecting that the icon has been dragged and dropped in the predetermined area as user event;
- displaying a target indicating the location of the predetermined area on the screen
- displaying the target whenever detecting that the icon has been pressed on during a predetermined duration, and erasing the target from the screen in response to detecting the user event;
- 15 • detecting a second user event ordering to stop the program after it has been started, and configuring the application into the disabled state in response to detecting the second user event;
- terminating, ending and/or killing any process of the program being executed whenever the program is configured into the disabled state;
- 20 • saving the state the program has been configured into a non-volatile memory of the terminal;
- configuring the program so that metadata of the program stored in a memory of the terminal is still accessible when said program is in the disabled state.

25 The invention also proposes a computer program product comprising code instructions for execution of the method for controlling the execution of a program by a terminal.

The invention also proposes a terminal comprising a screen, a memory storing a computer program, at least one processor for executing the program, said processor being configured to configure alternately the program into an enabled state, wherein at least one process of the program is allowed to be executed by the processor, and a disabled state, 30 wherein execution of any process of the program by the processor is forbidden, wherein the terminal is characterized in that the processor is further configured to display on the screen an icon for starting the program according to a first rendering if the program is configured in the enabled state, and display said icon according to a second rendering different from the first rendering if the program is configured in the disabled state.

35

DESCRIPTION OF FIGURES

Further features, goals and advantages will be highlighted in the following description which is not limitative and must be read in relation with the following figures.

- 5 • Figure 1 represents schematically a terminal according to an embodiment of the invention.
- Figure 2 represents steps of a method for controlling execution of a computer program by a terminal, according to an embodiment of the invention.
- Figures 2 and 3 represent steps of a method for controlling execution of a computer
10 program by a terminal according to an embodiment of the invention.
- Figures 4a to 4d represent different states of a terminal while the steps of figure 3 are carried out.
- Figure 5 represent other steps of a method for controlling execution of a computer program by a terminal according to an embodiment of the invention.
- 15 • Figures 6a to 6c represent different states of a terminal while the steps of figure 5 are carried out.

On all figures, similar elements have identical references.

DETAILED DESCRIPTION OF AN EMBODIMENT OF THE INVENTION

20 Referring to **figure 1**, a terminal comprises at least one processor 10, a screen 12, a user input interface 14 and a memory 16.

The memory 16 comprises a volatile memory unit 18 (such as RAM) and a non-volatile memory unit 20 (such as flash memory, HDD memory, SSD memory or a combination thereof).

25 The user input interface 14 typically is configured to capture user events: it may comprise a keyboard, a touchscreen 12, a mouse, or a combination thereof.

An operating system is stored in the non-volatile memory unit 20, as well as at least one computer program referred to as an “application” in the following.

The processor 10 is configured to execute the operating system and the application.

30 The operating system is configured to control the execution of the application by the processor 10.

The operating system comprises instructions for displaying a graphical user interface (GUI) on the screen 12 of the terminal 1. This GUI typically comprises a “home screen” menu, an application menu and a settings menu known of the prior art and discussed in the
35 introduction.

The application is associated with graphical data for displaying on the screen 12 an icon G (illustrated on figures 4a-4d and 6a-6c) for starting the application.

The operating system is further configured to start the application once a particular user event involving the icon G is detected. This user event will be described later on.

5 The application comprises at least one foreground process comprising code instructions for displaying a GUI specific to this application and for communicating with the user input interface 14 through the operating system.

10 The application further comprises at least one background process. As indicated before, the background process may be executed by the processor secretly, even though the user has closed the application.

The application may be configured by the operating system into two different states:

- an enabled state, wherein at least one process of the program is allowed to be executed by the operating system, and
 - a disabled state, wherein execution of any process of the program by the operating system is forbidden.
- 15

More particularly, the background process shall not be executed by the terminal 1 if the application is disabled.

In practice, a flag may identify the current state of an application: in the disabled state, the flag is set to a first value (say 0) and in the enabled state the flag is set to a second value different from the first value (say 1).

20

This flag can be saved in the non-volatile memory unit and updated whenever the state changes.

The operating system may secretly start any background process of an enabled application, i.e. without the user's explicit demand.

25 An enabled application may be automatically started when the operation system boots up. For instance, Windows operating system starts automatically an application at boot if said application is linked to a "startup" folder, has a service registered in the OS, or is in somewhere in the Windows Registry. Android operating system also offers such a feature by automatically invoking an enabled application once the device boot is completed.

30

The operating system can also start and/or run a process of an enabled program secretly whenever the user unlocks the terminal, i.e. after successful authentication of the user (using a PIN code, a gesture drawn on screen, etc.).

35 On the contrary, the operation system refrains from starting and running any process of a disabled application, including background processes.

In an embodiment, the operating system is authorized to access (read) metadata stored in the memory 16 (typically the non-volatile memory 20) that is specifically associated with a disabled program.

This metadata may include assets such as image data (especially icons), audio data, video data and/or text data (such as the program's manifest file).

Disabling an enabled application

The application is first supposed to be enabled, i.e. its state is supposed to be set to the value "enabled".

Referring to **figure 2**, the operating system displays the icon G of the application on the screen 12 according to a first rendering R1 in a step 100 (see **figure 4a**).

This first rendering R1 is for instance a default rendering of the graphical data using a predetermined set of colors.

This first rendering R1 can also comprise a first shape alteration of the default shape of the icon.

This first rendering R1 can comprise a first composition of the icon with a first indicator overlapping the icon.

This step 100 occurs when the user accesses a menu containing the icon G, such as the applications menu or the home screen H.

In a step 102, the operating system detects a predetermined user event indicating that the icon G has been moved on the screen 12 according to a first predetermined motion pattern.

In an embodiment the detection step 102 comprises the sub-steps illustrated on **figure 3** and the terminal 1 behaves as illustrated on figures **4a-4d**.

In this embodiment, the motion pattern is a motion of the icon G into a predetermined area A of the screen 12, for example a rectangular top part of the screen 12 of predetermined size.

In a sub-step 200, the operating system detects that the user presses on the icon G for a predetermined duration.

In response to said detection 200, the operating system displays 202 a target T indicating the location of the predetermined area A (see **figure 4a**).

Besides, the operation system checks the current state of the application and displays as target T a piece of text corresponding to the action of switching the state of the application from the current state into the other state. For instance, the word "disable" is displayed in the target T (see **figure 4b**).

Whenever the operation system detects that 204 the icon G has been dragged then dropped in the predetermined area A of the screen 12, disabling 104 or enabling 118 the application is triggered.

5 The target T is erased from the screen 12 when the user “drops” the icon G, i.e. when the operating system detects that the user stops pressing the icon G after dragging it (see **figures 4c and 4d**). Now turning back to **figure 2**, the operating system reconfigures 104 the applications in the “disabled” state in response to detecting 102 the user event. This reconfiguration makes sure that the operation system refrains from starting and running any background process of the application.

10 The operating system further checks if any process of the application is currently run by the processor 10. If yes, the operating systems terminates, ends and/or kills any running process of the application in a step 106. The space allocated in the volatile memory by these running processes is also freed.

15 Besides, the flag associated with the application is set 108 to the “disabled” value and saved 110 in the non-volatile memory of the terminal 1. As a consequence, the application will still be disabled even if the terminal 1 is turned off then turned on again and/or if the operation system reboots for some reason.

In a step 112, the operating system displays the icon G of the application according to a second rendering R2 different from the first rendering R1 (see **figure 4d**).

20 For instance, the second rendering R2 uses a limited set of colors, which is a subset of the set of colors used for the first rendering R1. In an embodiment, the icon G is displayed according to the second rendering in monochrome or in grayscale.

The second rendering R2 can also be a second shape alteration of the icon wherein the second shape alteration is different from the first shape alteration of rendering R1.

25 It should be understood that one of the two renderings R1 et R2 may not perform any alteration of the icon while the other rendering does.

Alternatively, this first rendering R1 can be a composition of the icon with a second indicator overlapping the icon that is different from the first indicator using in rendering R1.

30 Reconfiguration 104, termination 106, update 108 and displaying step 112 may be carried out in any order after detecting 102 the user event indicating that the icon G was moved according to the predetermined motion pattern.

Enabling a disabled application

Referring to **figure 5**, the application is switched from the “disabled” state to the “enabled” state according to the following steps.

We suppose that the icon G is displayed 114 according to the second rendering R2 (see **figure 6a**).

5 In step 116, the operation system detects that the user clicked on or tapped on the icon G of the disabled application.

In response to this detection 116, the operation also reconfigures 118 the application into the “enabled” state. Once the configuration 118 is complete, background processes are allowed to be run by the processor 10.

10 Besides, the operating system automatically starts 120 the application.

This automatic startup of the application is advantageous in that the user needs not access any settings menu in order to re-enable the application before starting it. In other words, enabling and launching an application are merged into one single user action.

15 At startup, a foreground process of the application is executed by the processor 10, which displays the GUI M of the application on the screen 12 (see **figure 6b**).

Besides, the flag associated with the application is set 122 to the “enabled” value and saved 122 in the non-volatile memory of the terminal 1. As a consequence, the application will still be enabled even if the terminal 1 is turned off then turned on again and/or if the operation system reboots for some reason.

20 Besides, the operating system displays the icon G of the application on the screen 12 according to a first rendering R1 in the step 100 (see **figure 6c**) in response to this detection 116.

25 This change may not be immediately visible on the screen 12 if the GUI M of the application actually covers the icon G. But this change will be visible on the screen once the user exits the application, for instance by clicking on the cross X illustrated on **figure 6b**.

In another embodiment, the application is not directly started whenever the user clicks on or taps on the icon G. The operation system rather displays a context menu whenever detecting that the icon G was pressed on or tapped on.

30 The context menu shows a first graphical item for directly starting the application, and a second graphical item for changing the state of the program.

If the operating system detects that the user clicks on or taps on the second item, the operation system reconfigures the application into the “enabled” state but the application is not directly started. From this point on, the operation system is allowed to start a
35 background process of the application secretly.

If the operating system detects that the user clicks on or taps on the first item, the operation system starts the application and temporarily allows processes of said application to be run until the user exits the application.

5 If the operating system detects that the user clicks or taps outside the context menu, which does not displayed full screen 12, the operating systems closes the context menu.

The operating system may also enable a disabled application whenever that the icon G has been moved on the screen 12 according to a second predetermined motion pattern.

10 This second motion pattern may be identical to the first motion pattern; in this case, all steps 200, 202, 204, 206 can be performed as well for enabling an application (but text “enable” is displayed in target T instead of text “disable”).

The method described upwards for controlling the execution of an application in the terminal 1 can be performed for each application installed in the terminal 1. Each application has its own state, its own icon, and can therefore be enabled and disabled independently of the other applications.

15 In an embodiment, a set of applications may be disabled or enabled by performing the drag-and-drop procedure of figure 3 on a folder containing icons of each application in said set of applications, rather than on a single icon G.

20 Another context in which the icon G of a program may be displayed is whenever a user of the terminal 1 is invited to select a program among a predetermined set of programs in response to a particular action.

25 Such an action is for example clicking on a “share” button printed on a web page or on the GUI of another program that is being displayed on the screen of the terminal 1. The predetermined set of programs comprises in this case all programs installed in the terminal 1 and implementing a function of sharing content over a network such as the Internet (mailing applications, social network clients and so on).

In response to clicking on the “share” button, the operating system displays on the screen all icons of said predetermined programs as a list.

Each icon is rendered using rendering R1 or R2, depending on the current state of each program.

30 Whenever the user clicks on or taps on an icon that is displayed according to rendering R2 (meaning that the corresponding application is in the “disabled” state), all steps 116, 118, 120 are performed.

35 A disabled application may also be enabled using other triggers that detection 116 involving the icon G, for instance whenever detecting that the user did a predetermined

gesture on the touchscreen (i.e drew a particular pattern), said predetermined gesture being associated with the program.

Besides, it should be understood that all steps described above that involve the icon G of the program may be performed for all short-cut icons of said program scattered in
5 different menus of the operating system (or the same menu).

More particularly, disabling or enabling a program triggers a change of rendering R1 or R2 of all icons of said program used by the user as short-cuts.

CLAIMS

1. Method for controlling the execution of a program by a terminal (1) comprising a screen (12), wherein the method comprises the steps of:

- 5
- configuring the program into:
 - an enabled state, wherein at least one process of the program is allowed to be executed by the terminal (1), and
 - a disabled state, wherein execution of any process of the program by the terminal (1) is forbidden,

10 wherein the method further comprises the steps of:

- displaying (100) on the screen (12) an icon (G) for starting the program according to a first rendering if the program is configured in the enabled state,
- displaying (114) said icon (G) according to a second rendering different from the first rendering if the program is configured in the disabled state.

15

2. Method as claimed in the preceding claim, comprising the steps of:

- rendering the icon (G) using a first set of colors if the current state is the enabled state,
- rendering the icon (G) is rendered using a second set of colors if the current state is the disabled state, wherein the second set of colors is a subset of the first set.

20

3. Method as claimed in any of the preceding claims, comprising:

- altering a default shape of the icon into a first altered shape if the current state is one of the disabled and enabled states,
- altering a default shape of the icon into a second altered shape or leaving said default shape unchanged, if the current state is the other state.

25

4. Method as claimed in any of the preceding claims, comprising:

- compositing the icon with a first indicator overlapping the icon, if the current state is one of the disabled and enabled states,
- compositing the icon with a second indicator overlapping the icon, wherein the second indicator is different from the first indicator, or leaving the icon free of indicator, if the current state is the other of the two states

30

35

5. Method as claimed in any of the preceding claims, comprising the steps of configuring the program into the enabled state whenever detecting that the program has been started using the icon (G).
- 5 6. Method as claimed in any of the preceding claims, comprising the steps of displaying a context menu whenever detecting that the icon (G) was pressed on or tapped on, said menu showing a first item for directly starting the application and another item for changing the state of the program.
- 10 7. Method as claims in any of the preceding claims, further comprising the steps of detecting a user event indicating that the icon (G) has been moved on the screen (12) according to a predetermined motion pattern, and changing the current state of the program in response to detecting the user event.
- 15 8. Method as claimed in the preceding claim, wherein the predetermined motion pattern is a motion of the icon (G) into a predetermined area (A) of the screen (12).
9. Method as claimed in the preceding claim, wherein the user event indicates that the icon (G) has been dragged and dropped in the predetermined area (A).
- 20 10. Method as claimed in any of claims 8 and 9, further comprising displaying a target (T) indicating the location of the predetermined area (A) on the screen (12).
11. Method as claimed in the preceding claim, further comprising:
- 25
 - displaying (202) the target (T) whenever detecting that the icon (G) has been pressed on during a predetermined duration,
 - erasing (206) the target from the screen (12) in response to detecting the user event.
- 30 12. Method as claimed in any of the preceding claims, further comprising:
- detecting a second user event ordering to stop the program after it has been started,
 - configuring the application into the disabled state in response to detecting the second user event.

13. Method as claimed in any of the preceding claims, further comprising terminating, ending and/or killing (106) any process of the program being executed whenever the program is configured into the disabled state.

5 14. Method as claimed in any of the preceding claims, further comprising saving (110, 124) the state the program has been configured into a non-volatile memory of the terminal (1).

15. Method as claimed in any of the preceding claims, wherein metadata of the program that is stored in a memory (20) of the terminal is still accessible when said program is in
10 the disabled state.

16. Computer program product comprising code instructions for execution of the method as claimed in any of the preceding claims, when this program product is executed by at least one processor (10).

15

17. Terminal (1) comprising:

- a screen (12),
- a memory (16) storing a computer program,
- at least one processor (10) for executing the program, said processor (10) being
20 configured to configure alternately the program into:
 - an enabled state, wherein at least one process of the program is allowed to be executed by the processor (10),
 - a disabled state, wherein execution of any process of the program by the processor (10) is forbidden,

25 wherein the terminal (1) is characterized in that the processor is further configured to:

- display on the screen (12) an icon (G) for starting the program according to a first rendering if the program is configured in the enabled state,
- display said icon (G) according to a second rendering different from the first rendering if the program is configured in the disabled state.

30

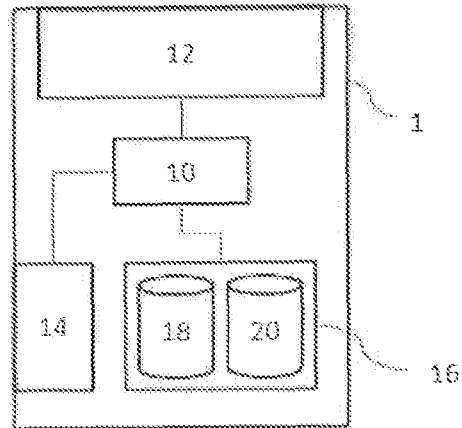


FIG. 1

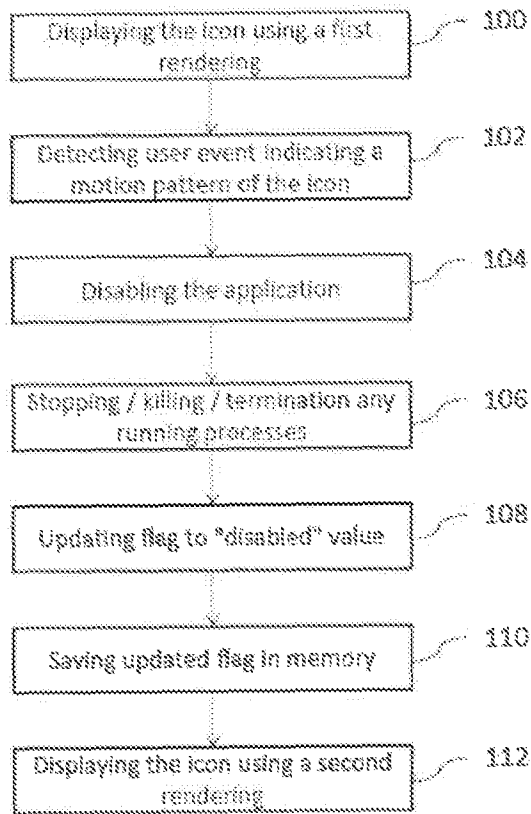


FIG. 2

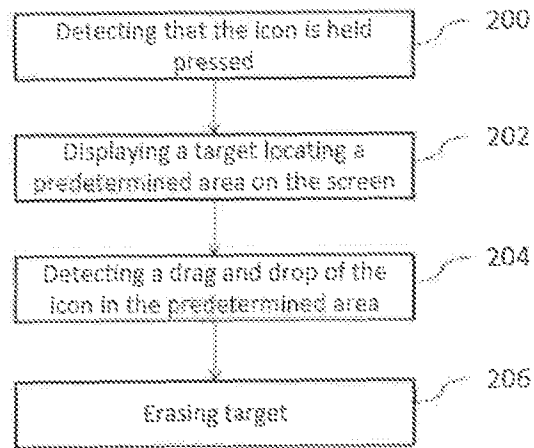


FIG. 3

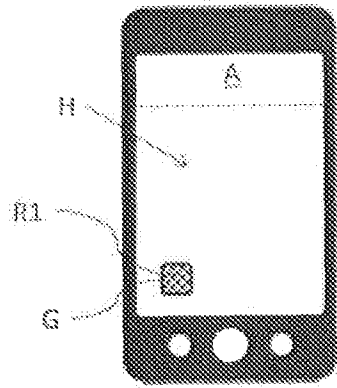


FIG. 4a

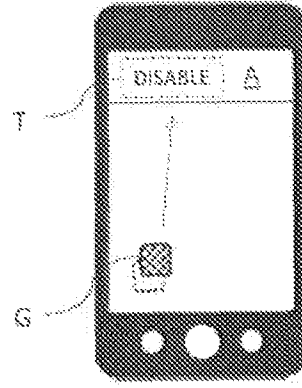


FIG. 4b

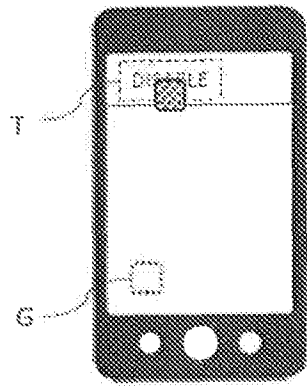


FIG. 4c

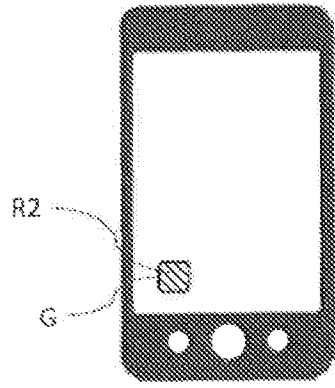


FIG. 4d

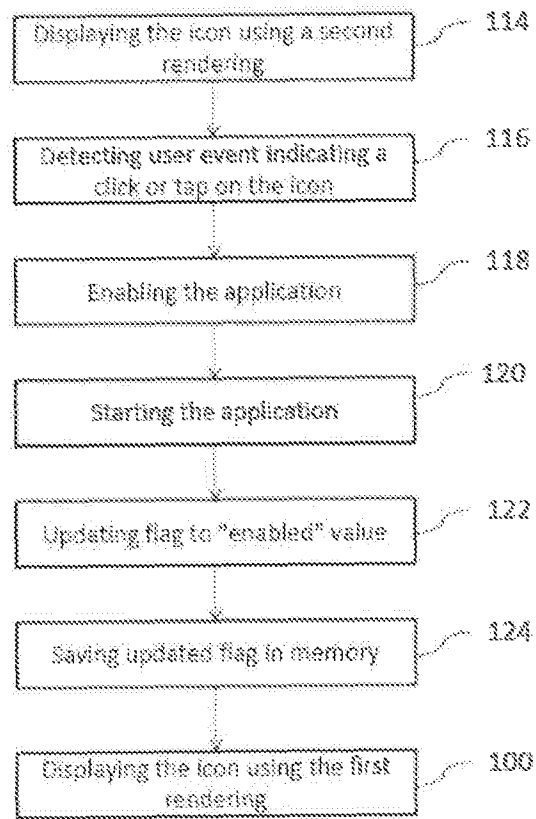


FIG. 5

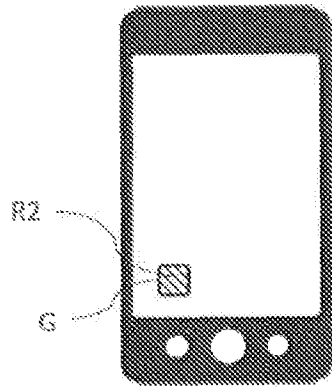


FIG. 6a

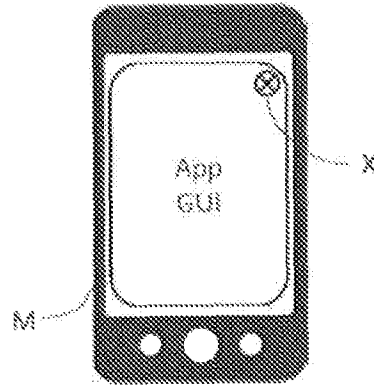


FIG. 6b

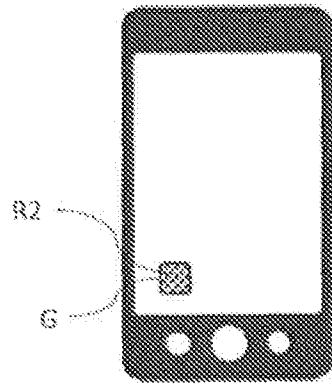


FIG. 6c

INTERNATIONAL SEARCH REPORT

International application No PCT/IB2016/001078
--

A. CLASSIFICATION OF SUBJECT MATTER INV. G06F9/44 ADD.				
According to International Patent Classification (IPC) or to both national classification and IPC				
B. FIELDS SEARCHED				
Minimum documentation searched (classification system followed by classification symbols) G06F				
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched				
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) EPO-Internal, WPI Data				
C. DOCUMENTS CONSIDERED TO BE RELEVANT				
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.		
X	Anonymous: "How do I uninstall, turn off, and hide applications on my Samsung Galaxy S5 mini?", 1 January 2015 (2015-01-01), XP055310303, Retrieved from the Internet: URL: http://www.samsung.com/us/support/answer/ANS00045673/ [retrieved on 2016-10-13] the whole document ----- -/--	1-17		
<input checked="" type="checkbox"/> Further documents are listed in the continuation of Box C.				
<input checked="" type="checkbox"/> See patent family annex.				
* Special categories of cited documents :				
<table style="width: 100%; border: none;"> <tr> <td style="width: 50%; border: none; vertical-align: top;"> <p>"A" document defining the general state of the art which is not considered to be of particular relevance</p> <p>"E" earlier application or patent but published on or after the international filing date</p> <p>"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)</p> <p>"O" document referring to an oral disclosure, use, exhibition or other means</p> <p>"P" document published prior to the international filing date but later than the priority date claimed</p> </td> <td style="width: 50%; border: none; vertical-align: top;"> <p>"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention</p> <p>"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone</p> <p>"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art</p> <p>"&" document member of the same patent family</p> </td> </tr> </table>			<p>"A" document defining the general state of the art which is not considered to be of particular relevance</p> <p>"E" earlier application or patent but published on or after the international filing date</p> <p>"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)</p> <p>"O" document referring to an oral disclosure, use, exhibition or other means</p> <p>"P" document published prior to the international filing date but later than the priority date claimed</p>	<p>"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention</p> <p>"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone</p> <p>"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art</p> <p>"&" document member of the same patent family</p>
<p>"A" document defining the general state of the art which is not considered to be of particular relevance</p> <p>"E" earlier application or patent but published on or after the international filing date</p> <p>"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)</p> <p>"O" document referring to an oral disclosure, use, exhibition or other means</p> <p>"P" document published prior to the international filing date but later than the priority date claimed</p>	<p>"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention</p> <p>"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone</p> <p>"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art</p> <p>"&" document member of the same patent family</p>			
Date of the actual completion of the international search		Date of mailing of the international search report		
13 October 2016		21/10/2016		
Name and mailing address of the ISA/ European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016		Authorized officer Kusnierczak, Pawel		

INTERNATIONAL SEARCH REPORT

International application No PCT/IB2016/001078

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	<p>Anonymous: "Android Guide: Managing Your Installed Applications", 3 May 2015 (2015-05-03), XP055310169, Retrieved from the Internet: URL:http://web.archive.org/web/20150503112446/http://www.howtogeek.com/school/basic-android-guide/lesson2/all [retrieved on 2016-10-12] the whole document</p> <p align="center">-----</p>	1-17
A	<p>EP 2 840 486 A1 (SAMSUNG ELECTRONICS CO LTD [KR]) 25 February 2015 (2015-02-25) paragraph [0086] - paragraph [0198]</p> <p align="center">-----</p>	1-17

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/IB2016/001078

Patent document cited in search report	Publication date	Patent family member(s)	Publication date	
EP 2840486	A1	25-02-2015	EP 2840486 A1	25-02-2015
			KR 20150022460 A	04-03-2015
			US 2015056974 A1	26-02-2015
