

DYNAMIC GRAPHIC CODE IMPLEMENTATION METHOD AND APPARATUS

5 TECHNICAL FIELD

The specification relates to the field of Internet technologies, and in particular, to a dynamic graphic code implementation method and apparatus.

TECHNICAL BACKGROUND

A graphic code such as a bar code or a QR code is an information carrier, and is a
10 graphic identifier generated by processing, such as coding, information according to a certain rule.

To ensure that private information carried in the graphic code is not stolen, it is necessary to provide a more reliable and secure graphic code implementation scheme.

SUMMARY OF INVENTION

15 In view of the above, the specification provides a dynamic graphic code implementation method and apparatus.

Specifically, the specification is achieved by the following technical solutions:

A dynamic graphic code implementation method, comprising:

invoking a distance sensor to detect a distance between a device and a blocking object
20 when displaying a graphic code; and

if the distance is within a set distance range, stopping refreshing the graphic code.

A dynamic graphic code implementation apparatus, comprising:

a distance detecting unit configured to invoke a distance sensor to detect a distance
between a device and a blocking object when displaying a graphic code; and

a refresh stopping unit configured to stop refreshing the graphic code if the distance is within a set distance range.

A dynamic graphic code implementation apparatus, comprising:

a processor; and

5 a memory configured to store a machine executable instruction;

wherein by reading and executing the machine executable instruction stored in the memory corresponding to a dynamic graphic code implementation control logic, the processor is caused to:

10 invoke a distance sensor to detect a distance between a device and a blocking object when displaying a graphic code; and

if the distance is within a set distance range, stop refreshing the graphic code.

As can be seen from the above description, this embodiment can invoke a distance sensor to detect a distance between a device and a blocking object when displaying a graphic code, and stop refreshing the graphic code if the distance is within a set distance
15 range, which avoids the security risk caused by the case that a code scanning device scans two graphic codes before and after refreshing, thereby improving the reliability and security of the dynamic graphic code.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a schematic diagram showing a dynamic graphic code implementation
20 process according to an example embodiment of the specification.

FIG. 2 is a schematic structural diagram showing a dynamic graphic code implementation apparatus according to an example embodiment of the specification.

FIG. 3 is a block diagram showing a dynamic graphic code implementation apparatus according to an example embodiment of the specification.

25

DETAILED DESCRIPTION OF THE EMBODIMENTS

Example embodiments will be described in detail herein, examples of which are illustrated in the accompanying drawings. If the drawings are involved in the following description, the same numeral in different drawings refers to the same or similar elements, unless otherwise indicated. The implementations described in the following example embodiments do not represent all implementations consistent with the specification. Instead, the implementations are merely examples of apparatuses and methods consistent with what are described in detail in the appended claims and some aspects of the specification.

The terms used in the specification are merely intended to describe particular embodiments, and are not intended to limit the specification. The singular forms "a" and "the" used in the specification and the appended claims are also intended to include plural forms, unless other meanings are clearly indicated in the context. It should also be understood that the term "and/or" as used herein refers to and encompasses any or all possible combinations of one or more associated listed items.

It should be understood that although the terms, first, second, third, etc. may be used in the specification to describe various information, such information should not be limited to these terms. These terms are only used to distinguish the same type of information from each other. For example, the first information may also be referred to as the second information without departing from the scope of the specification. Similarly, the second information may also be referred to as the first information. Depending on the context, the word "if" as used herein may be interpreted as "at the time of" or "when" or "in response to determination."

In this embodiment, in addition to the specified information, a dynamic graphic code generally carries a time factor, such as a timestamp of the time when the dynamic graphic

code is generated. A graphic code generating device can generate graphic codes carrying different time factors for the specified information with the change of time, and a graphic code analyzing device can determine whether the graphic code is reliable according to the time factor, thereby improving the security of the graphic code and reducing the risk of information leakage.

FIG. 1 is a schematic diagram showing a dynamic graphic code implementation process according to an example embodiment of the specification.

Referring to FIG. 1, the specification provides a dynamic graphic code implementation method, which can be applied to a client that displays a graphic code. The client is generally installed in an electronic device such as a mobile phone or a tablet computer. The dynamic graphic code implementation method can include the following steps:

Step 102: a distance sensor is invoked to detect the distance between a device and a blocking object when a graphic code is displayed.

In this embodiment, the display of the graphic code can be triggered by a user. For example, in offline electronic payment, the user opens a payment graphic code, and in riding a bus, the user opens a riding graphic code. The payment graphic code and the riding graphic code are generally dynamic graphic codes, which can be updated over time. In this embodiment, the distance sensor of the device can be invoked to detect the distance between the device and the blocking object when the graphic code is displayed. For example, the device is a mobile phone, and the distance sensor is generally located in the vicinity of a phone receiver. The positions of distance sensors of different devices may also be different, and details will not be described in this embodiment.

In this embodiment, after the graphic code is displayed, a code scanning operation is generally performed, and the blocking object is generally a code scanning device, such as a barcode scanner and a terminal device having a code scanning function. Certainly, the

blocking object may also be other objects, such as a user's finger or other objects, and is not particularly limited in this embodiment.

Step 104: If the distance is within a set distance range, refreshing the graphic code is stopped.

5 Based on step 102, when the graphic code is displayed, if it is detected that the distance between the device and the blocking object is within the set distance range, it is indicated that the probability that the code scanning operation is currently being performed is high, and the refreshing of the graphic code can be stopped, so as to avoid the security risk caused by the refreshing of the graphic code in the code scanning process, for
10 example, the code scanning device scans two graphic codes before and after refreshing, and then two payment requests are initiated.

The distance range is generally a range of distance from the device when the code scanning device can successfully scan the graphic code, which can be preset by a developer, for example, 0 cm-5 cm, and 0 cm-10 cm, etc.

15 In this embodiment, in a client refresh mode, stopping refreshing the graphic code may include: not generating a new graphic code when the refresh time of the graphic code arrives. The client refresh mode refers to the generation of graphic code by a client.

For example, the graphic code is a payment QR code, it is assumed that the refresh period of the payment QR code is 1 minute, the blocking object is a code scanning device.

20 After the payment QR code is displayed, if the refresh mode is the client refresh mode, the client can generate a new payment QR code according to the user's account information and time factor every minute, and refresh the currently displayed payment QR code with the newly generated payment QR code. It is also assumed that the client displays the payment QR code at 16:22:02, the next refresh time of the payment QR code is 16:23:02.

25 If it is detected at 16:22:58 that the distance between the mobile phone and the code

scanning device is within the set distance range, the client does not generate a new payment QR code at 16:23:02.

In this embodiment, in a server refresh mode, stopping refreshing the graphic code may include: not sending a refresh request to the server when the refresh time of the graphic code arrives. The server refresh mode refers to the generation of graphic code by the server. For example, when the graphic code is displayed or refreshed, the client sends a request to the server, and the server generates a graphic code and then sends it to the client.

For example, the graphic code is a payment QR code, the refresh period of the payment QR code is 1 minute, and the blocking object is a code scanning device. After the payment QR code is displayed, if the refresh mode is the server refresh mode, the client can send a refresh request to the server every minute, and the server generates a new payment QR code according to the user's account information and the time factor, and sends the newly generated QR code to the client for the client to refresh. It is also assumed that the client displays the payment QR code at 16:22:02, the next refresh time of the payment QR code is 16:23:02. If it is detected at 16:22:58 that the distance between the mobile phone and the code scanning device is within the set distance range, the client does not send a refresh request to the server at 16:23:02.

As can be seen from the above description, this embodiment can invoke a distance sensor to detect the distance between a device and a blocking object when displaying a graphic code, and stop refreshing the graphic code if the distance is within a set distance range, which avoids the security risk caused by the case that the code scanning device scans two graphic codes before and after refreshing, thereby improving the reliability and security of the dynamic graphic code.

Based on the embodiment as shown in FIG. 1, the refreshing of the graphic code can be restored if it is detected that the distance between the device and the blocking object is not within the distance range after stopping refreshing the graphic code.

For example, the graphic code is a payment QR code, the refresh period of the payment QR code is 1 minute, and the blocking object is a code scanning device. It is assumed that the set distance range is 0 cm-5 cm, and the client displays the payment QR code at 16:22:02. If it is detected at 16:22:58 that the distance between the mobile phone and the code scanning device is 4.5 cm, the refreshing of the payment QR code can be stopped, for example, at the next refresh time 16:23:02, the client does not generate a new payment QR code. It is also assumed that at 16:23:15, the distance between the mobile phone and the code scanning device is detected to be 15 cm, which is not within the distance range. If the payment QR code is still displayed at the moment, the refresh of the payment QR code can be restored.

In an example, the implementation for restoring the refreshing of the payment QR code may be: refreshing the payment QR code when it is detected that the distance between the mobile phone and the code scanning device is not within the distance range, for example, a new payment QR code is generated according to the current timestamp and the user account for refreshing. Still taking the above example as an example, the payment QR code can be refreshed at 16:23:15. After the payment QR code is refreshed, the next refresh time can be based on the time 16:22:02 when the payment QR code is displayed, for example, 16:24:02, or based on the refresh time 16:23:15 after the restoring, for example, 16:24:15, which is not particularly limited in this embodiment.

In another example, the implementation for restoring the refresh of the payment QR code may also be: refreshing the payment QR code when the next refresh time arrives. Still taking the above example as an example, at 16:23:15, the distance between the

mobile phone and the code scanning device is detected to be 15 cm, and the refreshing of the payment QR code can be made at the next refresh time 16:24:02 of the payment QR code.

Certainly, the refreshing of the payment QR code can be restored in other ways, which
5 is not particularly limited in this embodiment.

Based on the embodiment as shown in FIG. 1, after the refreshing of the graphic code is stopped, the refreshing of the graphic code can be restored when a time duration since the refreshing the graphic code is stopped reaches a set time duration. The set time duration is generally a time duration required for successful scanning, and can also be set
10 by a developer, such as 5 s and 10 s.

For example, the graphic code is a payment QR code, the refresh period of the payment QR code is 1 minute, and the blocking object is a code scanning device. It is assumed that the time duration since the refreshing is stopped is 10 s, and the client displays the payment QR code at 16:22:02. If the refreshing of the payment QR code is
15 stopped at 16:22:58, the refreshing of the payment QR code can be restored at 16:23:08. For the implementation for restoring the refreshing of the payment QR code, reference can be made to the foregoing description, and details are not described in this embodiment again.

Corresponding to the embodiments of the foregoing dynamic graphic code
20 implementation method, the specification also provides embodiments of a dynamic graphic code implementation apparatus.

The embodiments of the dynamic graphic code implementation apparatus of the specification can be applied to an electronic device. The apparatus embodiments can be implemented by software, or may be implemented by hardware or a combination of
25 hardware and software. Taking the software implementation as an example, a logical

apparatus is formed by reading a corresponding computer program instruction in a non-volatile memory into a memory by a processor of the electronic device in which the apparatus is located. From the hardware aspect, FIG. 2 shows a hardware structure diagram of the electronic device in which the dynamic graphic code implementation
5 apparatus of the specification is located. In addition to the processor, the memory, a network interface, and the non-volatile memory as shown in FIG. 2, the electronic device in which the apparatus in the embodiment is located can also include other hardware according to actual functions of the electronic device, and details are not described herein.

FIG. 3 is a block diagram showing a dynamic graphic code implementation apparatus
10 according to an example embodiment of the specification.

Referring to FIG. 3, the dynamic graphic code implementation apparatus 200 can include a distance detecting unit 201, a refresh stopping unit 202, a first restoring unit 203, and a second restoring unit 204.

The distance detecting unit 201 is configured to invoke a distance sensor to detect the
15 distance between a device and a blocking object when displaying a graphic code;

The refresh stopping unit 202 is configured to stop refreshing the graphic code if the distance is within a set distance range.

The first restoring unit 203 is configured to restore the refreshing of the graphic code if it is detected that the distance between the device and the blocking object is not within the
20 distance range after stopping refreshing the graphic code.

The second restoring unit 204 is configured to restore the refreshing of the graphic code when a time duration since the refreshing of the graphic code is stopped reaches a set time duration.

Optionally, the restoring the refreshing of the graphic code by the first restoring unit
25 203 or the second restoring unit 204 includes:

the graphic code is refreshed when the next refresh time arrives.

Optionally, the refresh stopping unit 202 is configured to not send a refresh request to a server when the refresh time of the graphic code arrives in a server refresh mode.

For details of the implementation process of the functions and effects of units in the foregoing apparatus, reference can be made to the implementation process of the corresponding steps in the foregoing method, and details will not be described herein again.

Since the apparatus embodiments basically correspond to the method embodiments, reference can be made to the partial description of the method embodiments for relevant contents. The apparatus embodiments described above are merely illustrative, where the units described as separate components may or may not be physically separated, and the components displayed as units may or may not be physical units, that is, may be located at the same position, or may also be distributed to multiple network units. Some or all of the modules can be selected according to actual requirements to achieve the objective of the scheme of the specification. A person of ordinary skill in the art can understand and implement the above without involving any inventive effort.

The systems, apparatuses, modules or units illustrated in the above embodiments may be implemented by computer chips or entities, or implemented by a product having a certain function. A typical implementation device is a computer, and the specific form of the computer may be a personal computer, a laptop computer, a cellular phone, a camera phone, a smart phone, a personal digital assistant, a media player, a navigation device, an email transceiver, a game console, a tablet computer, a wearable device, or any combination of these devices.

Corresponding to the foregoing embodiment of the dynamic graphic code implementation method as shown in FIG. 1, the specification also provides a dynamic

graphic code implementation apparatus, including: a processor and a memory configured to store a machine executable instruction. The processor and the memory are generally connected to each other by an internal bus. In other possible implementations, the device may also include an external interface to communicate with other devices or components.

5 In this embodiment, by reading and executing the machine executable instruction stored in the memory corresponding to the dynamic graphic code implementation control logic, the processor is caused to:

 invoke a distance sensor to detect the distance between a device and a blocking object when displaying a graphic code; and

10 if the distance is within a set distance range, stop refreshing the graphic code.

 Optionally, by reading and executing the machine executable instruction stored in the memory corresponding to the dynamic graphic code implementation control logic, the processor is further caused to:

 restore the refreshing of the graphic code if it is detected that the distance between the
15 device and the blocking object is not within the distance range after stopping refreshing the graphic code.

 Optionally, by reading and executing the machine executable instruction stored in the memory corresponding to the dynamic graphic code implementation control logic, the processor is further caused to:

20 restore the refreshing of the graphic code when a time duration since the refreshing of the graphic code is stopped reaches a set time duration.

 Optionally, when the refreshing of the graphic code is restored, the processor is caused to:

 refresh the graphic code when the next refresh time arrives.

Optionally, when the refreshing of the graphic code is stopped, the processor is caused to:

not send a refresh request to a server when the refresh time of the graphic code arrives in a server refresh mode.

5 Corresponding to the foregoing embodiment of the dynamic graphic code implementation method as shown in FIG. 1, the specification also provides a computer readable storage medium having a computer program stored thereon, and when the program is executed by a processor, the following steps are implemented:

a distance sensor is invoked to detect the distance between a device and a blocking
10 object when displaying a graphic code; and

if the distance is within a set distance range, the refreshing of the graphic code is stopped.

Optionally, the following step is further included:

the refreshing of the graphic code is restored if it is detected that the distance between
15 the device and the blocking object is not within the distance range after stopping refreshing the graphic code.

Optionally, the following step is further included:

the refreshing of the graphic code is restored when a time duration since the refreshing of the graphic code is stopped reaches a set time duration.

20 Optionally, the restoring the refreshing of the graphic code includes:

the graphic code is refreshed when the next refresh time arrives.

Optionally, the stopping the refreshing of the graphic code includes:

a refresh request is not sent to a server when the refresh time of the graphic code arrives in a server refresh mode.

The above describes the specific embodiments of the specification. Other embodiments are within the scope of the appended claims. In some cases, the actions or steps recited in the claims can be performed in a different order than the embodiments and can still achieve the desired results. In addition, the processes depicted in the drawings are
5 not necessarily in a particular order or in a sequential order to achieve the desired results. In some embodiments, multi-task processing and parallel processing are also possible or may be advantageous.

The above are only preferred embodiments of the specification, and are not intended to limit the specification. Any modification, equivalent substitution, improvement, etc. made
10 within the spirit and principle of the specification should be included in the protection scope of the specification.

CLAIMS

[Claim 1] A dynamic graphic code implementation method, comprising:

invoking a distance sensor of a device to detect a distance between the device and a

5 blocking object when displaying a graphic code on the device, wherein refreshing the graphic code is scheduled for a first refresh time;

if the distance between the device and the blocking object is within a preset distance range, stopping refreshing the graphic code scheduled for the first refresh time; and

10 after stopping refreshing the graphic code, if it is detected that the distance between the device and the blocking object is not within the preset distance range, restoring the refreshing of the graphic code when a next refresh time for refreshing the graphic code arrives.

[Claim 2] The method according to claim 1, further comprising:

15 restoring the refreshing of the graphic code when a time duration since the refreshing of the graphic code is stopped reaches a preset time duration.

[Claim 3] The method according to claim 1, wherein the stopping refreshing the graphic code comprises:

20 in a server refresh mode, when the first refresh time of the graphic code arrives, not sending a refresh request to a server.

[Claim 4] A non-transitory computer-readable storage medium having stored therein instructions that, when executed by a processor of the device, cause the device to perform the method of any one of claims 1-3.

25

[Claim 5] An apparatus for dynamic graphic code implementation, comprising:

a processor; and
a memory configured to store a machine executable instruction;
wherein by executing the machine executable instruction stored in the memory, the
processor is caused to:

5 invoke a distance sensor of the apparatus to detect a distance between the apparatus
and a blocking object when displaying a graphic code on the apparatus, wherein refreshing
the graphic code is scheduled for a first refresh time;
 stop refreshing the graphic code scheduled for the first refresh time, if the distance
between the apparatus and the blocking object is within a preset distance range; and
10 after stopping refreshing the graphic code, if it is detected that the distance between
the apparatus and the blocking object is not within the preset distance range, restore the
refreshing of the graphic code when a next refresh time for refreshing the graphic code
arrives.

15 [Claim 6] The apparatus according to claim 5, wherein the processor is further caused to:
 restore the refreshing of the graphic code when a time duration since the refreshing of the
graphic code is stopped reaches a preset time duration.

[Claim 7] The apparatus according to claim 5, wherein the processor is further caused to:
20 in a server refresh mode, not send a refresh request to a server when the first refresh time of
the graphic code arrives.