US011721058B2

(12) **United States Patent**
Bonansea

(10) **Patent No.: US 11,721,058 B2**
(45) **Date of Patent: *Aug. 8, 2023**

(54) **CUSTOMIZABLE ANIMATIONS FOR TEXT MESSAGES**

(71) Applicant: **EMONSTER INC**, Seattle, WA (US)

(72) Inventor: **Enrique Bonansea**, Tokyo (JP)

(73) Assignee: **EMONSTER INC.**, Seattle, WA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **18/069,104**

(22) Filed: **Dec. 20, 2022**

(65) **Prior Publication Data**

US 2023/0119376 A1 Apr. 20, 2023

Related U.S. Application Data

(63) Continuation of application No. 17/751,509, filed on May 23, 2022, now Pat. No. 11,532,114, which is a continuation of application No. 17/210,404, filed on Mar. 23, 2021, now Pat. No. 11,341,707, which is a continuation of application No. 15/723,068, filed on Oct. 2, 2017, now Pat. No. 10,957,088, which is a continuation of application No. 14/811,426, filed on Jul. 28, 2015, now Pat. No. 9,779,532.

(60) Provisional application No. 62/031,752, filed on Jul. 31, 2014.

(51) **Int. Cl.**

G06T 13/80	(2011.01)
H04L 51/10	(2022.01)
H04W 4/14	(2009.01)

(52) **U.S. Cl.**

CPC ***G06T 13/80*** (2013.01); ***H04L 51/10*** (2013.01); ***H04W 4/14*** (2013.01)

(58) **Field of Classification Search**

CPC G06T 13/40; G06T 13/00; G06T 13/80; G06T 13/20; A63F 13/10; H04W 4/14; H04W 4/12; H04W 88/184; H04M 1/72552; H04M 3/42382; H04L 51/18

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

6,453,294	B1 *	9/2002	Dutta	G10L 21/06 704/235
7,216,092	B1	5/2007	Weber et al.	
8,639,214	B1 *	1/2014	Fujisaki	G06Q 20/306 455/406
8,832,201	B2 *	9/2014	Wall	H04L 51/046 709/200
8,949,461	B2 *	2/2015	Steele	H04L 67/565 709/246
9,405,821	B1	8/2016	Bessis	
9,779,532	B2 *	10/2017	Bonansea	H04L 51/10
10,957,088	B2 *	3/2021	Bonansea	H04L 51/10
11,341,707	B2 *	5/2022	Bonansea	H04W 4/14
11,532,114	B2 *	12/2022	Bonansea	H04W 4/14

(Continued)

OTHER PUBLICATIONS

Unicode 7.0.0 Standard, https://www.unicode.org/versions/Unicode7.0.0/ (last visited Jul. 29, 2014).

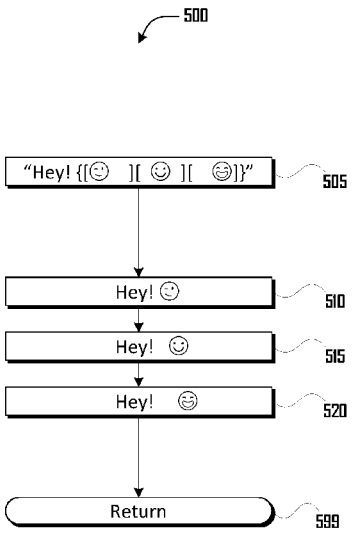
Primary Examiner — Todd Buttram

(74) *Attorney, Agent, or Firm* — AEON Law, PLLC; Adam L. K. Philipp; David V. H. Cohen

(57) **ABSTRACT**

A method and system for transforming simple user input into customizable animated images for use in text-messaging applications.

20 Claims, 7 Drawing Sheets



(56) **References Cited**

U.S. PATENT DOCUMENTS

2003/0011643 A1* 1/2003 Nishihata H04N 21/4312 715/810
2003/0110450 A1* 6/2003 Sakai G06F 40/109 715/256
2004/0179037 A1* 9/2004 Blattner H04L 51/00 715/751
2004/0199906 A1 10/2004 McKnight et al.
2006/0019636 A1 1/2006 Guglielmi et al.
2006/0199575 A1* 9/2006 Moore H04M 3/493 455/418
2008/0182566 A1* 7/2008 Camp, Jr. H04L 51/18 455/418
2010/0162138 A1 6/2010 Pascal et al.
2011/0246890 A1* 10/2011 Mellamphy H04L 51/58 715/719
2011/0296324 A1* 12/2011 Goossens G06F 3/04845 715/763
2012/0001924 A1 1/2012 Richardson
2012/0101805 A1* 4/2012 Barbosa G06F 40/237 704/9
2013/0086008 A1* 4/2013 Souza H04L 12/6418 707/E17.122
2014/0372541 A1 12/2014 Feghali
2015/0287403 A1 10/2015 Zaslansky et al.
2015/0302628 A1* 10/2015 Ye G06F 3/0483 715/721
2016/0035123 A1* 2/2016 Bonansea H04W 4/14 345/473
2018/0082461 A1* 3/2018 Bonansea H04L 51/10
2021/0209826 A1* 7/2021 Bonansea H04L 51/10
2022/0284653 A1* 9/2022 Bonansea H04L 51/10

* cited by examiner

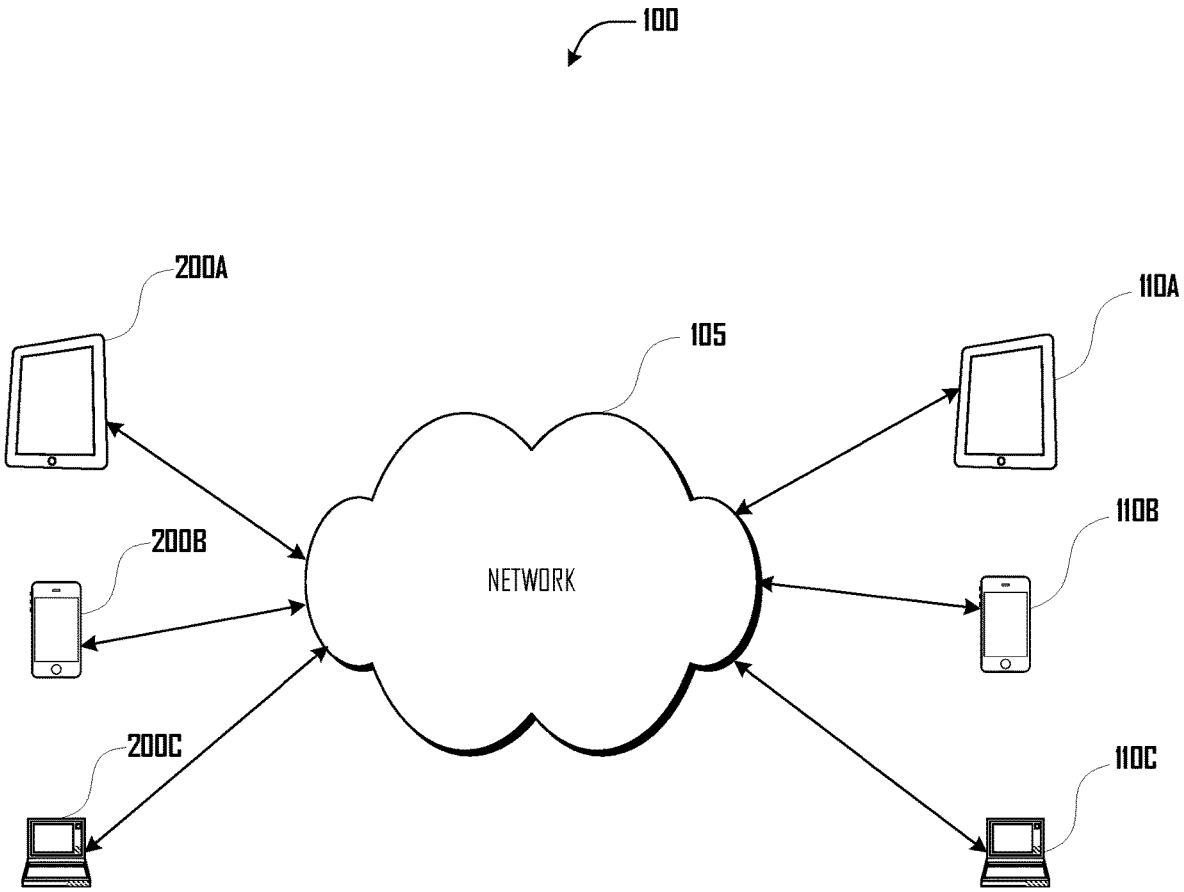


Fig.1

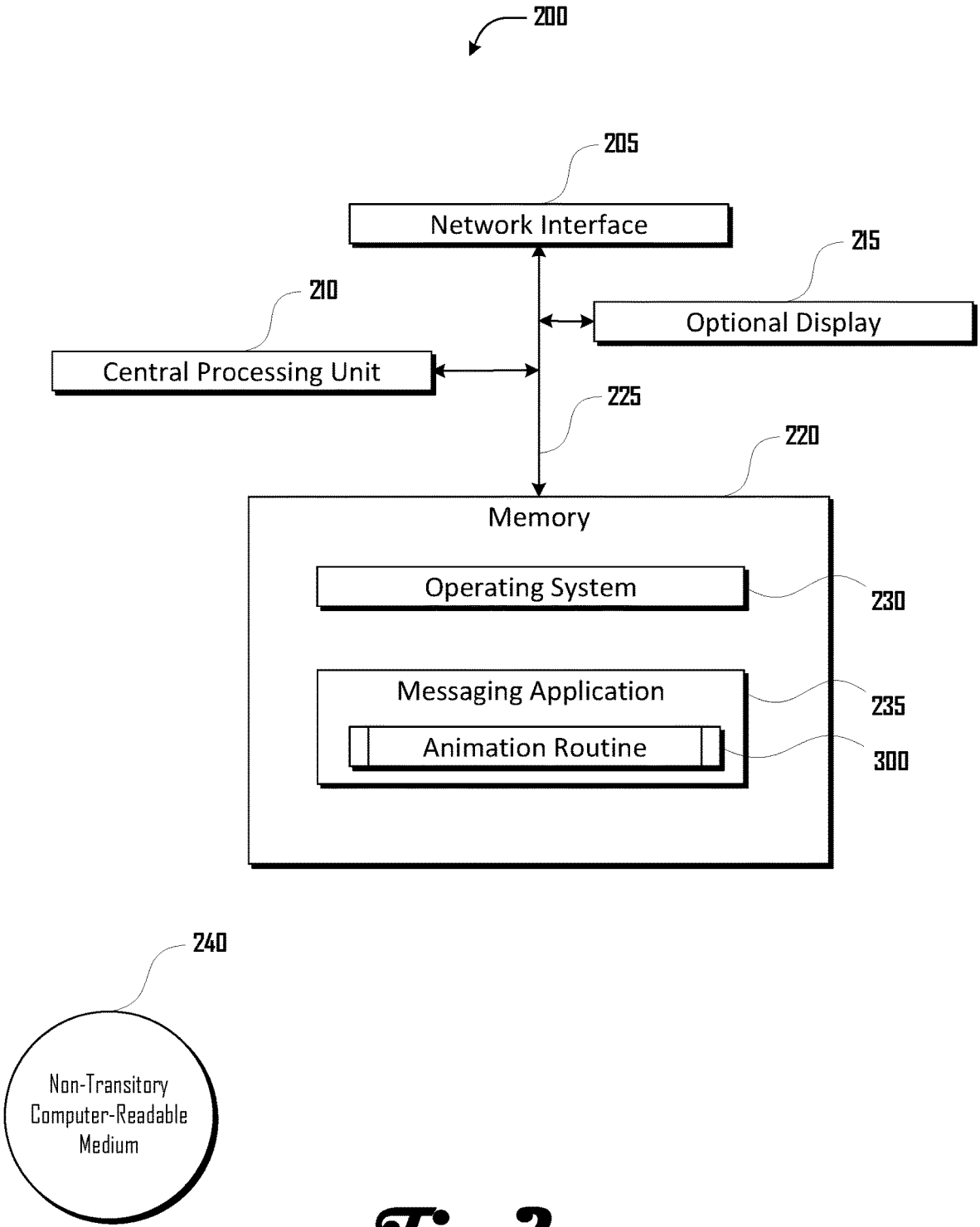

200
205
Network Interface
215
Optional Display
210
Central Processing Unit
225
220
Memory
Operating System
230
Messaging Application
235
Animation Routine
300
240
Non-Transitory Computer-Readable Medium

Fig.2

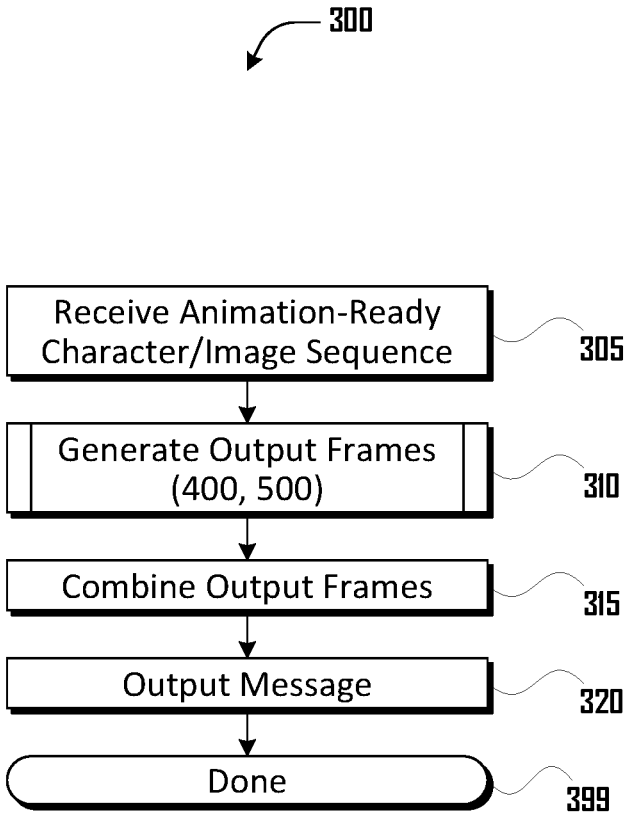

300
Receive Animation-Ready Character/Image Sequence
305
Generate Output Frames (400, 500)
310
Combine Output Frames
315
Output Message
320
Done
399

Fig.3

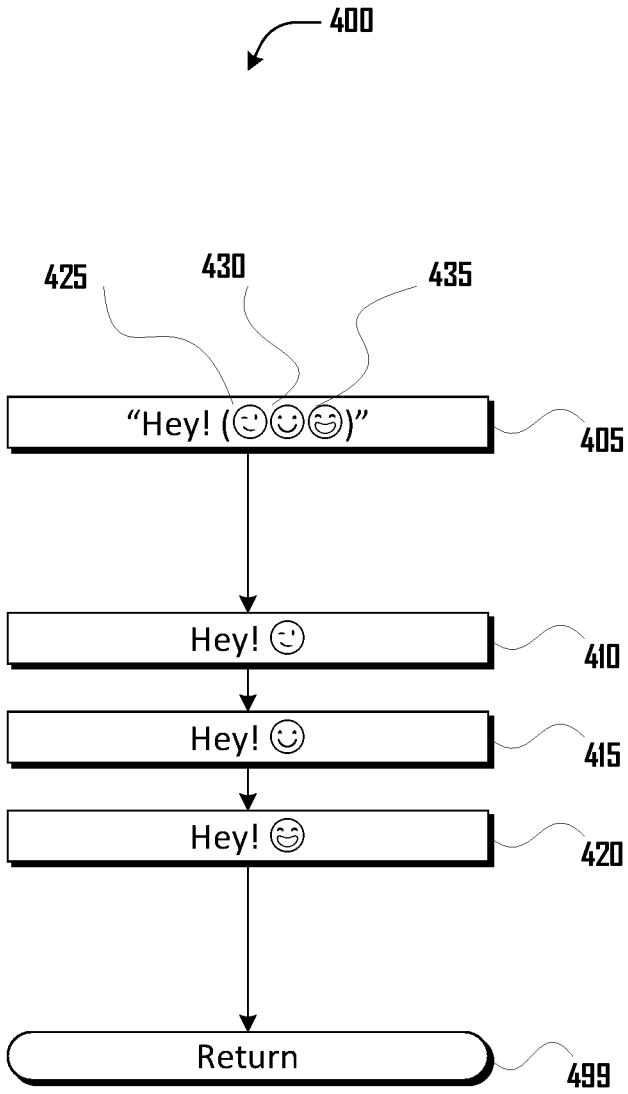


Fig. 4

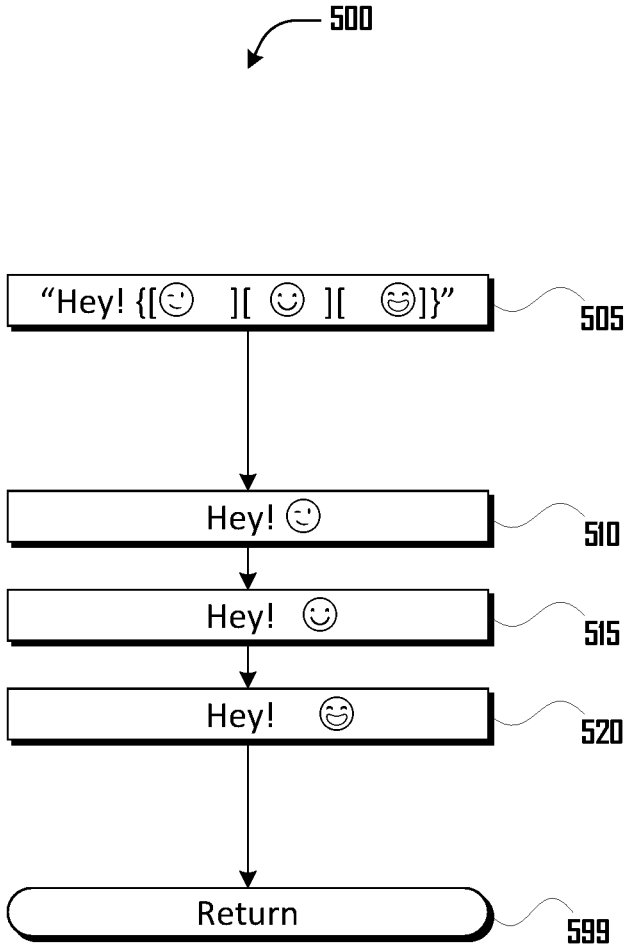


Fig.5

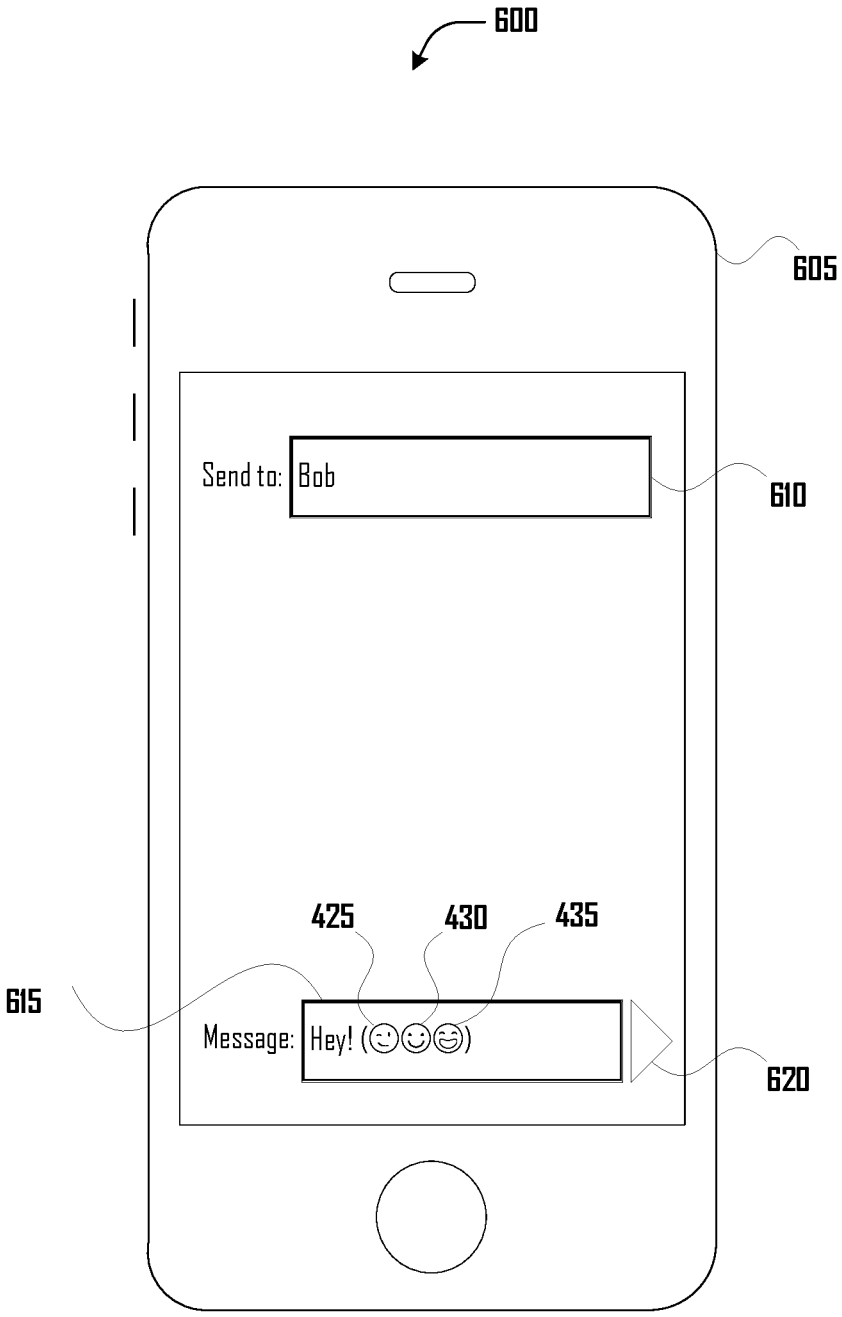


Fig.6

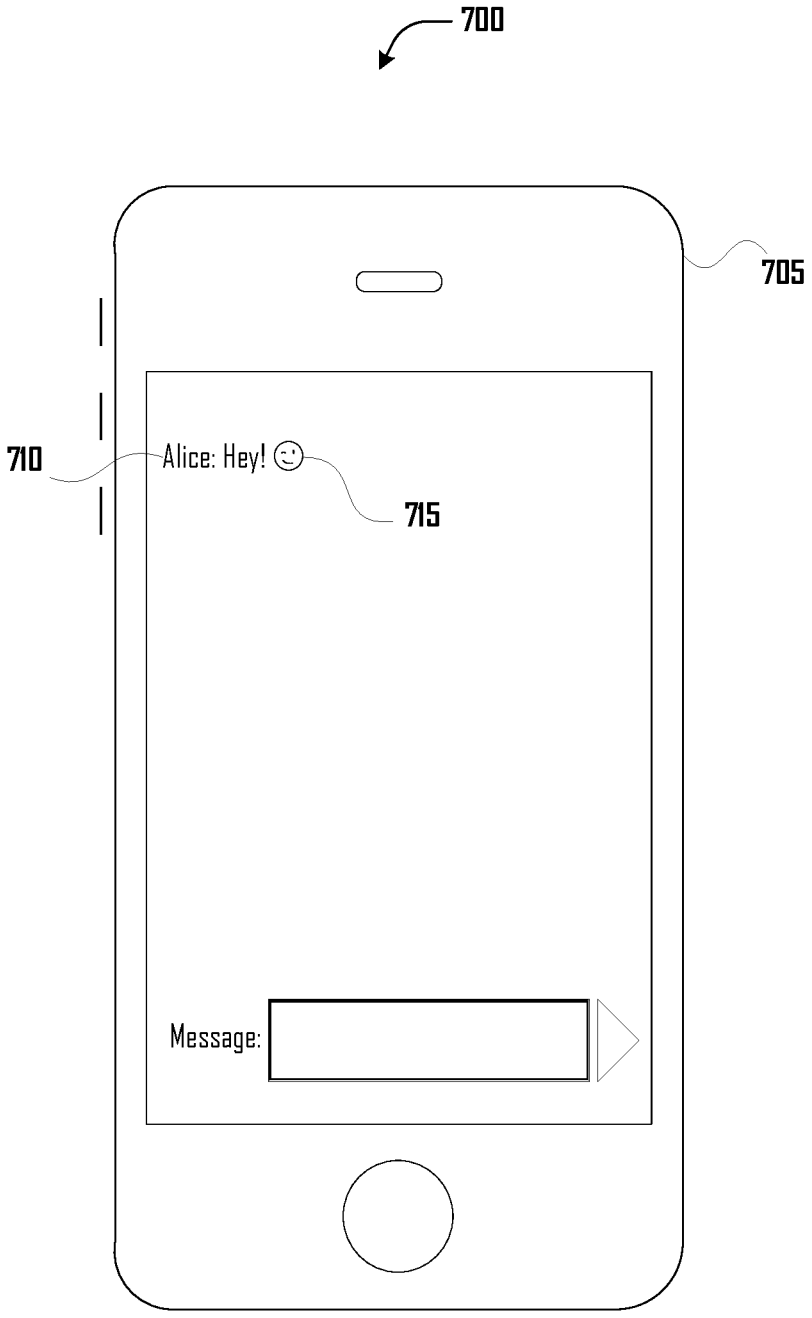


Fig. 7

CUSTOMIZABLE ANIMATIONS FOR TEXT MESSAGES

CROSS-REFERENCE TO RELATED APPLICATIONS

This application claims priority to non-provisional U.S. patent application Ser. No. 17/751,509, filed May 23, 2022 and issued as U.S. Pat. No. 11,532,114 on Dec. 20, 2022; to non-provisional U.S. patent application Ser. No. 17/210,404, filed Mar. 23, 2021 and issued as U.S. Pat. No. 11,341,707 on May 24, 2022; to non-provisional U.S. patent application Ser. No. 15/723,068, filed Oct. 2, 2017 and issued as U.S. Pat. No. 10,957,088 on Mar. 23, 2021; and to non-provisional U.S. patent application Ser. No. 14/811,426, filed Jul. 28, 2015 and issued as U.S. Pat. No. 9,779,532 on Oct. 3, 2017; and claims the benefit of provisional U.S. Patent Application No. 62/031,752, filed Jul. 31, 2014. The above-cited applications are hereby incorporated by reference, in their entireties, for all purposes. The Application Data Sheet filed herewith forms a part of the present application, and all priority documents to which it refers are incorporated by reference herein in their entirety.

FIELD

This disclosure is directed to the field of communication software, particularly text-messaging applications on mobile devices.

BACKGROUND

Text messaging is a method of communicating short messages between various types of devices. Text messages are transmitted over the Short Message Service (SMS) and Multimedia Messaging Service (MMS) protocols. Messages sent via SMS are limited to 160 characters in length. MMS messages may contain longer strings of text, as well as photo, audio, or video data. Messages sent over both protocols are commonly referred to as "text messages" or "texts."

Because SMS messages are limited to 160 characters in length, users have developed shorthand conventions for expressing common sentiments in such messages, such as "LOL" ("laughing out loud"), "ROFL" ("rolling on the floor laughing"), "BRB" ("be right back"), "NP" ("no problem"), and many others. Although MMS messages do not share the same length limitation, such messages typically are similarly brief due to a variety of factors, including user convention, device interface considerations, historical technological limitations, and bandwidth concerns. Thus, these shorthand conventions are common in text messages sent both by SMS and by MMS.

Some of these common abbreviations are known as emoticons. Emoticons use text characters to depict certain emotions, usually as representations of facial expressions. For example, probably the most common emoticon is a "smiley face," typically drawn as two or three characters: ":)" or ":-)". Other common emoticons include ";)" (a wink), ":D" (an extra-large or open-mouthed smile), and ":P" (tongue sticking out). Emoticons allow text users to express complex emotions using very few characters, conserving available space in text messages and reducing typing requirements.

Another type of symbol used in text messages is known as an emoji. Emojis are essentially pictographic versions of emoticons or other symbols. In order to use an emoji, a user

may type a specific character sequence indicating to his text-messaging software that he wishes to send an associated emoji image, though some text-messaging software also allows users to directly select the desired image rather than typing in a character sequence. The text-messaging software then replaces the specified character sequence with the appropriate image and sends it to the intended recipient. Some such emojis are static images, while others may be animated. The Unicode Standard character set adopted several hundred emojis in version 6.0, and added approximately 250 more in the latest version, 7.0. The current Unicode Standard may be found at http://www.unicode.org/versions/Unicode7.0.0/ (last visited Jul. 29, 2014), and is hereby incorporated by reference in its entirety.

One drawback to emojis, however, is that the images used are drawn from fixed sets of images built in to the users' text-messaging software. Users are thus unable to customize the images they send to express any sentiment beyond those already provided with their software. Moreover, the set of animated emojis may be particularly limited compared to the various thoughts that a user may wish to express with an animated image in a text message, and the animations are typically limited to in-place images that cannot move in relation to the surrounding characters.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** is a network and device diagram illustrating exemplary computing devices configured according to embodiments disclosed in this paper.

FIG. **2** illustrates several components of an exemplary computing device capable of sending and receiving text messages in accordance with the present disclosure.

FIG. **3** illustrates an animation routine which converts user input into a desired animation.

FIG. **4** illustrates an animation subroutine, showing sample user input and the corresponding generated output frames.

FIG. **5** illustrates an animation subroutine, utilizing an alternate syntax to provide additional functionality.

FIG. **6** illustrates an exemplary user interface of a computing device for sending a message in accordance with the present disclosure.

FIG. **7** illustrates an exemplary user interface of a computing device after receiving a message prepared and sent in accordance with the present disclosure.

DESCRIPTION

The presently disclosed method and system presents a simple mechanism for allowing users to use, combine, and animate any arbitrary sequence of characters or images to be sent in a text message. An exemplary embodiment is described in which an emoji coding language (ECL) is used to allow a user to specify various properties, such as sequencing, spacing, and images or character sequences that he wishes to animate. The embodiment then processes the user's ECL input, converts it to a format that typical text-messaging software can render, and sends the message. The recipient then sees the sender's intended animation, regardless of the text-messaging software on the recipient's device and even if the message includes images not otherwise present on the recipient's device.

The phrases "in one embodiment," "in various embodiments," "in some embodiments," and the like are used repeatedly. Such phrases do not necessarily refer to the same

embodiment. The terms "comprising," "having," and "including" are synonymous, unless the context dictates otherwise.

Reference is now made in detail to the description of the embodiments as illustrated in the drawings. While embodiments are described in connection with the drawings and related descriptions, there is no intent to limit the scope to the embodiments disclosed herein. On the contrary, the intent is to cover all alternatives, modifications, and equivalents. In alternate embodiments, additional methods, or combinations of illustrated methods, may be added to or combined without limiting the scope to the embodiments disclosed herein.

FIG. **1** depicts a communications system **100** such as those operated by mobile-phone carriers. Sender devices **200** such as tablet **200**A, cell phone **200**B, and personal computer **200**C connect to network **105** in order to exchange information, including text messages, with other users using receiving devices such as tablet **110**A, cell phone **110**B, and personal computer **110**C. Network **105** could be a private network such as those operated by mobile telephony operators, or a public network such as the Internet. Network **105** could also comprise combinations of various public and private networks.

FIG. **2** depicts several components of an exemplary computing device **200** capable of sending and receiving text messages in accordance with the present disclosure. In various embodiments, computing device **200** may be a mobile phone, tablet, desktop PC, laptop, server, workstation, set-top box, appliance, or other computing device that is capable of performing operations such as those described herein. In some embodiments, computing device **200** may include many more components than those shown in FIG. **2**. However, it is not necessary that all of these generally conventional components be shown in order to disclose an illustrative embodiment.

Computing device **200** includes a bus **225** interconnecting several components including a network interface **205**, a central processing unit **210**, an optional display **215**, and a memory **220**.

Memory **220** generally comprises a random-access memory and permanent non-transitory mass storage device, such as a hard disk drive or solid-state drive. Memory **220** stores program code for an operating system **230** and an exemplary messaging application **235** ("App"). The App comprises program code for animation routine **300** (see FIG. **3**). These and other software components may be loaded into memory **220** using a drive mechanism (not shown) associated with a non-transitory computer-readable medium **240**, such as a floppy disk, tape, DVD/CD-ROM drive, memory card, or the like.

FIG. **3** shows animation routine **300**, by which the App converts user input into the desired animation. The App first receives the user's input, possibly including ECL code, in box **305**. In box **310**, the App processes the user input, as described by animation subroutines **400** and **500**, in order to generate the series of images for animation ("output frames") that the user requested. The App combines the generated output frames into an animated GIF file in box **315**, and then outputs the animated GIF in box **320**. Animation routine **300** then ends in block **399**.

Targets for the output operation of box **320** could include a display on the user's device (e.g., to allow for review before sending), a communications network that can deliver the message to the intended recipient, as with communications network **105**, or any other target that may prove useful. If the output target is a communications network such as network **105**, the animation may be sent as an attachment to an MMS message. By sending the message as an animated GIF attached to an MMS message, any text-messaging program used by the recipient should be able to render the animation, regardless of the type of device (e.g., tablet, cell phone, PC), platform (e.g., iOS or Android), or particular text-messaging software used by the recipient. Techniques for combining several image frames into an animated GIF and for sending an animated GIF as an MMS attachment are well known to those skilled in the art.

FIG. **4** depicts animation subroutine **400**, showing sample user input and the corresponding generated output frames. Block **405** depicts an input sequence of characters and images, consisting of the word "Hey!" followed by a space, an open parenthesis, emoji images **425**, **430**, and **435**, and a close parenthesis. In block **405**, subroutine **400** parses this input and decodes the animation commands that the user has included in the text. Subroutine **400** recognizes the parentheses as indicating a sequence of characters or images that the user wishes to animate. Subroutine **400** then generates a sequence of three output frames as shown in boxes **410**, **415**, and **420**. Each output frame contains the text "Hey!" followed by a space and one of the three designated emoji images, in the order in which the user entered them. In box **499**, subroutine **400** returns to animation routine **300**, where generated output frames **410**, **415**, and **420** are combined and output in boxes **315** and **320**, as described above.

The sequence of characters and images processed in subroutine **400** employs a simple syntax in which each character or image placed between matched pairs of parentheses corresponds to a single output frame. Thus, because block **405** contains three emoji images (and no spaces) between the parentheses, the App generates three sequential output frames **410**, **415**, and **420** in order to produce the resulting animation.

The user may, however, wish to create an animation wherein the selected characters or images move within the resulting output frames. The simple syntax shown in block **405** would not allow for this, because subroutine **400** would transform any additional spacing characters into their own distinct output frames.

FIG. **5** shows animation subroutine **500**, utilizing an alternate syntax employed by the App in order to provide this additional functionality. Block **505** depicts a sequence of characters and images consisting of the word "Hey!" followed by a space, a left curly brace, three matched pairs of square brackets containing distinct sequences of characters and emoji images, and then a right curly brace. When subroutine **500** encounters a matched pair of curly braces containing one or more matched pairs of square brackets, it interprets the contents of each matched pair of square brackets as corresponding to a single desired output frame. Subroutine **500** thus interprets the sequence of characters and images shown in block **505** as a request to generate three distinct output frames corresponding to the contents of each matched pair of square brackets, respectively. Thus, by using spaces as shown in block **505**, the sender can create an animation that appears to move laterally within the resulting output frames, as shown in boxes **510**, **515**, and **520**. Subroutine **500** returns to animation routine **300** at box **599**.

The parentheses shown in box **405** operate simply as a shorthand notation for the more complete syntax used in box **505**. In order to allow the user to specify simple animations with a minimum of required keystrokes, the App interprets parentheses as shown in box **405** (i.e., not contained within a matched pair of curly braces) as equivalent to the syntax shown in box **505**, wherein each character within the paren-

theses in box **405** is used to generate a single output frame. Thus, the input shown in box **405**, "Hey! (☺☺😊)" is functionally equivalent to the input "Hey! {[☺][☺][😊]}" but requires fewer keystrokes to enter.

FIG. **6** depicts an exemplary user interface for sending a message in accordance with the present disclosure. Sender Alice opens the above-described App on her mobile phone **605**. In selection field **610**, Alice specifies that she wishes to send a message to recipient Bob. In input field **615**, Alice enters the message she wishes to animate and send, including text characters and emoji images **425**, **430**, and **435**. Alice then selects button **620**, indicating that she wishes to animate the message according to animation routine **300** and then send it to Bob.

After Alice's message is processed and sent, Bob receives it as shown in FIG. **7**. Bob's mobile phone **705** indicates that it has received an MMS message **710** from Alice. The MMS message includes an attachment consisting of a three-frame animated GIF. In accordance with the normal operation of text-messaging software, the animated GIF is displayed in-line in Bob's interface, showing each frame in sequence. Thus, Bob sees the word "Hey!" followed by a space and then image **715**. Image **715** initially consists of emoji **425**. As Bob's text-messaging software renders subsequent frames of the animated GIF, image **715** appears to change into emoji **430**, and then emoji **435**. Bob thus sees the animation as specified by Alice. Because the message is transmitted in a standardized format, namely an animated GIF attached to an MMS message, Bob sees the animation as intended by Alice even if he has a different type of phone, different text-messaging software, or a different network carrier than her.

Many variations in the above-described details of the App, or features not included in the App, are possible while remaining within the scope of the present disclosure. Some examples include, without limitation:

Syntax

Additional syntax could prove useful for certain types of desired animations. For example, if the sender wished to animate an emoticon, he may wish to include parentheses as part of the animation. For example, the sequence "(^_^)"->"(^_-)" (i.e., a wink) might be an animation a user would wish to send. Because the above-described App interprets parentheses as special characters, however, this type of animation would not be possible. The inclusion of "escape characters," however, such as a backslash or tick mark, could allow a sender to specify, for example, that he wants the character immediately before or after the escape character to be interpreted as part of the message itself and not as part of an animation command. Thus, using a backslash as an escape character, the sender could enter the text string "Hey! {[\(^_^\)][\(^_-\)]}" to achieve the desired message, including an animated emoticon containing parentheses. The use of escape characters in this way is well known to those of ordinary skill in the art.

Syntax additions could also provide further capabilities, such as specifying animation parameters (e.g., time delay between output frames) on a per-message basis. Additional syntax may also be useful for incorporating additional types of animations, as described below, and other uses. The designation and use of specific syntax to enable this type of functionality is well known to those skilled in the art.

Animation Types

Beyond the types of animations described above, further animation capabilities may be desirable. For example, users may wish to incorporate rotation or horizontal or vertical flipping of characters or images into their animations. By specifying certain characters to indicate these commands (e.g., the character "^" could be interpreted as a command to vertically flip the next character or image in subsequent output frames), other embodiments could easily incorporate these types of features. Other uses for such additional commands could include, e.g., specifying different colors or altering the size of the animated characters or images.

Other embodiments could also implement logic to smoothen the output animation by interpolating between the sender-specified ("source") output frames, automatically generating additional output frames, and inserting the additional output frames between the source output frames as appropriate. Techniques for interpolating between two images are well-known to those skilled in the art.

Animation Parameters

Users may wish to have control over certain parameters governing the animations generated from their input. For example, one sender may want a delay of one second between each output frame, whereas another sender may want a delay of five seconds. Or a sender may wish to generate an animation that replays the output frames once forward and then once backward. Or, if an embodiment includes interpolation capability as described above, certain additional parameters may be associated with that functionality, such as the granularity of the interpolation (i.e., how many interpolated frames to generate for each pair of adjacent source output frames). The present disclosure is intended to encompass all embodiments with such additional features.

Messaging Protocols

Although the messages described in the exemplary embodiments are text messages transmitted via the MMS protocol with animated GIF attachments, the present disclosure is not limited to those technologies. Instead, the scope of the disclosure is intended to include any messaging technology, whether or not currently existing, in which a coding language similar to the presently disclosed ECL could prove useful in allowing senders to create flexible animations out of arbitrary sequences of characters and images. For example, various embodiments could operate in conjunction with instant-messaging software such as Skype, manufactured by Microsoft Corporation of Redmond, Wash., Google+ Hangouts, manufactured by Google, Inc., of Mountain View, Calif., and others, without departing from the scope of the present disclosure.

Application and System Features

Certain embodiments may provide additional application- or system-level features or options. For example, one potential embodiment could show the user the generated animation prior to sending the message, in order to allow for any necessary corrections. In another embodiment, there may be no user-level application at all; instead, animation routine **300** could be implemented as a software module that interacts with third-party text-messaging applications, so that the user can use a text-messaging application of his choice but still enjoy the benefits of the presently disclosed method and system. Animation routine **300** could even be implemented by a network operator at the network level, so that senders could create and send custom animations according to the present disclosure without needing to install any special software on their devices at all. All such embodiments are within the intended scope of the present disclosure.

I claim:

1. A system to animate an instant-messaging message, the system comprising one or more computing devices including one or more processors and memories containing pro-

gram code for execution by the one or more processors, the one or more computing devices configured to:

receive a message to animate and send,

wherein the message to animate and send includes one or more text and/or emoji characters;

receive a selected animation command to apply a type of animation to the message;

generate an animation of the message by applying the commanded type of animation to the message;

display the generated animation of the message; and

receive a selection to send the message to a recipient over a communications network so that the recipient can see the message animated according to the selected animation command.

2. The system of claim **1** wherein the one or more computing devices include a sender device that is a mobile phone, a tablet, a laptop, or a desktop personal computer.

3. The system of claim **2** wherein the program code for execution by the one or more processors is a messaging app or program on the sender device, and wherein the message to animate and send is entered via an input interface of the messaging app or program on the sender device.

4. The system of claim **3** wherein the messaging app or program allows a user to directly select a desired Unicode emoji or emoji image to include in the message.

5. The system of claim **1** wherein receiving the selected animation command to apply a type of animation to the message further includes receiving an indication that a user wishes to animate the message.

6. The system of claim **1** wherein the selected animation command to apply a type of animation to the message is one of a set of provided animation commands.

7. The system of claim **1** wherein the type of animation is one of a set of pre-coded types of animation.

8. The system of claim **1** wherein the selected animation command to apply a type of animation to the message includes altering a size of a character or image of the message.

9. The system of claim **1** wherein the selected animation command to apply a type of animation to the message includes moving a character or image of the message, so that the message moves in relation to other characters on a display.

10. The system of claim **1** wherein the selected animation command to apply a type of animation to the message includes rotating a character or image of the message.

11. The system of claim **1** wherein the selected animation command to apply a type of animation to the message includes changing a color of a character or image of the message.

12. The system of claim **1** wherein generating an animation of the message by applying the commanded type of animation to the message includes generating a series of images of the message processed according to the commanded type of animation.

13. The system of claim **12** wherein displaying the generated animation of the message includes sending the generated animation of the message to a display.

14. The system of claim **1** wherein displaying the generated animation of the message includes showing a user the generated animation of the message prior to sending the message and allowing the user to correct the selected animation command to apply a type of animation to the message.

15. The system of claim **1** wherein receiving a selection to send the message to a recipient over a communications network so that the recipient can see the message animated according to the selected animation command includes receiving a selection to send the message to the recipient as an instant-messaging protocol message.

16. The system of claim **1**, further comprising, in response to receiving the selection to send the message to a recipient, sending the message over a communications network using an instant-messaging protocol.

17. The system of claim **16** wherein the system is configured to send the message in a format that allows the recipient to see the message animated according to the selected animation command.

18. The system of claim **1** wherein the system allows a user to apply a selected animation from a set of pre-coded animations with a minimum of required input.

19. A non-transitory machine-readable medium having stored thereon instructions to animate an instant-messaging message that, when executed by one or more processors in one or more computing devices, configure the computing devices to:

receive a message to animate and send,

wherein the message to animate and send includes one or more text and/or emoji characters;

receive a selected animation command to apply a type of animation to the message;

generate an animation of the message by applying the commanded type of animation to the message;

display the generated animation of the message; and

receive a selection to send the message to a recipient over a communications network so that the recipient can see the message animated according to the selected animation command.

20. A method to allow a user to animate an instant-messaging message, comprising providing machine-readable instructions executable by a computing system to:

receive, from the user, in a memory, a message to animate and send,

wherein the message to animate and send includes one or more text and/or emoji characters;

receive a selected animation command to apply a type of animation to the message;

generate, by a processor, an animation of the message by applying the commanded type of animation to the message;

output the generated animation of the message to a display; and

receive a selection to send the message to a recipient over a communications network connected to a network interface so that the recipient can see the message animated according to the selected animation command.

* * * * *