October 2021

OFF-PROCESS GUI RENDERING

Roberto Perez

Babak Bostan

Jorge Pereira

OFF-PROCESS GUI RENDERING

ABSTRACT

An application may offload rendering of a graphical user interface (GUI) to an underlying operating system executing at a head unit (e.g., an infotainment system) of a vehicle (e.g., automobile, motorcycle, a bus, a recreational vehicle (RV), a semi-trailer truck, a tractor or other type of farm equipment, a train, a plane, a helicopter, etc.) or a computing device (e.g., a cellular phone, a smartphone, a desktop computer, a laptop computer, a tablet computer, a portable gaming device, a watch, etc.). The application may invoke functions of an application library that allow the application to generate GUI metadata including a GUI template (e.g., that specifies a predetermined layout) and content for populating the GUI template. The operating system, which may expose an interface (e.g., defined by the application library) for interacting with a display of the head unit or some other display in communication with the head unit, may receive, from the application, the GUI metadata and a token identifying the application. The operating system may then use the GUI metadata and the token to render a corresponding GUI for the application. In this way, the OS may render GUIs for applications in accordance with GUI templates, enabling a consistent look and feel that may, as one example, satisfy safety guidelines.

DESCRIPTION

FIG. 1 below is a conceptual diagram illustrating a head unit 100 (e.g., an infotainment system) of a vehicle (e.g., an automobile, a motorcycle, a bus, a recreational vehicle (RV), a semi-trailer truck, a tractor or other type of farm equipment, a train, a plane, a helicopter, etc.) that includes an operating system 102 (“OS 102”) configured to render a graphical user interface

(GUI) on behalf of applications 104A-104N (collectively, “applications 104”) executing at head unit 100. As shown in FIG. 1, head unit 100 may include one or more processors 106, a presence-sensitive display 108, and one or more storage devices 110. As shown in FIG. 1, storage devices 110 includes OS 102, applications 104, and an application library 112.

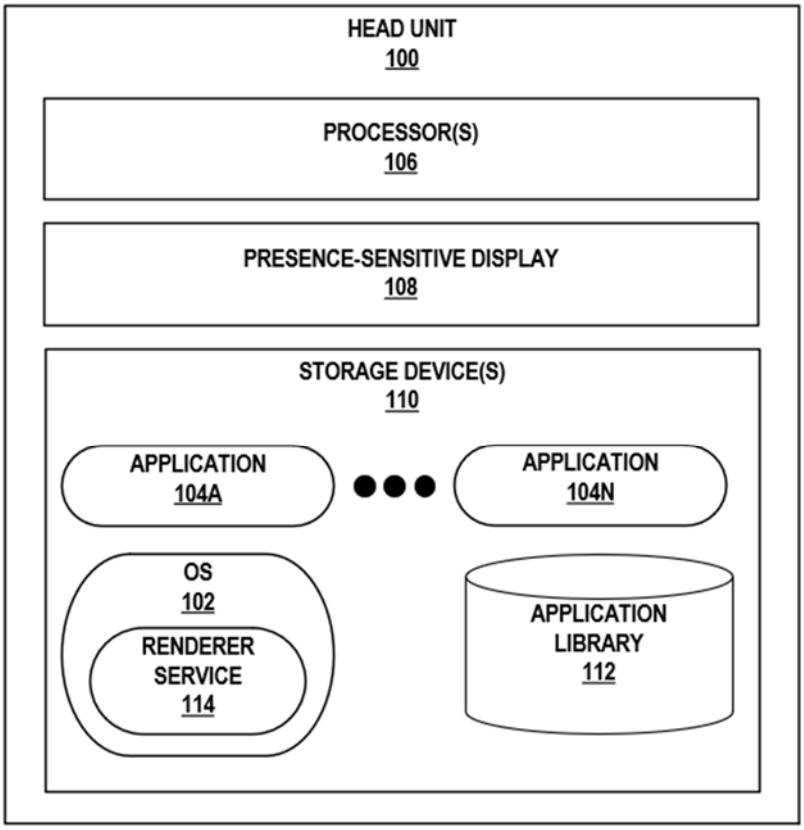

FIG. 1

Head unit 100 may operate to assist, inform, entertain, and/or provide for interactions with one or more occupants of a vehicle. Head unit 100 may represent an integrated head unit that provides a user interface (UI), such as a voice user interface (VUI), a graphical user interface (GUI), etc. In general, head unit 100 may control one or more vehicle systems, such as a heating, ventilation, and air conditioning (HVAC) system, a lighting system (for controlling interior and/or exterior lights), an infotainment system, a seating system (for controlling a position of a driver and/or passenger seat), etc.

Processors 106 may implement functionality and/or execute instructions associated with head unit 100. Examples of processors 106 may include one or more of an application specific integrated circuit (ASIC), a field programmable gate array (FPGA), an application processor, a display controller, an auxiliary processor, a central processing unit (CPU), a graphics processing unit (GPU), one or more sensor hubs, and any other hardware configure to function as a processor, a processing unit, or a processing device.

Presence-sensitive display 108 of head unit 100 may be a presence-sensitive display that functions as an input device and as an output device. For example, presence-sensitive display 108 may function as an input device using a presence-sensitive input component, such as a resistive touchscreen, a surface acoustic wave touchscreen, a capacitive touchscreen, a projective capacitance touchscreen, a pressure sensitive screen, an acoustic pulse recognition touchscreen, or another presence-sensitive display technology. Additionally, presence-sensitive display 108 may function as an output (e.g., display) device using any of one or more display components, such as a liquid crystal display (LCD), dot matrix display, light emitting diode (LED) display, microLED display, organic light-emitting diode (OLED) display, e-ink, active-matrix organic light-emitting diode (AMOLED) display, or similar monochrome or color display capable of outputting visible information to a user of head unit 100.

Storage devices 110 may include one or more computer-readable storage media. For example, storage devices 110 may be configured for long-term, as well as short-term storage of information, such as instructions, data, or other information used by head unit 100. In some examples, storage devices 110 may include non-volatile storage elements. Examples of such non-volatile storage elements include magnetic hard discs, optical discs, solid state discs, and/or the like. In other examples, in place of, or in addition to the non-volatile storage elements,

storage devices 110 may include one or more so-called “temporary” memory devices, meaning that a primary purpose of these devices may not be long-term data storage. For example, the devices may comprise volatile memory devices, meaning that the devices may not maintain stored contents when the devices are not receiving power. Examples of volatile memory devices include random-access memories (RAM), dynamic random-access memories (DRAM), static random-access memories (SRAM), etc.

As shown in FIG. 1, storage devices 110 includes OS 102. OS 102 may provide an execution environment for applications 104. OS 102 may represent a multi-threaded operating system or a single-threaded operating system with which applications 104 may interface to access hardware of head unit 100. OS 102 may include a kernel that facilitates access to the underlying hardware of head unit 100, where kernel may present a number of different interfaces (e.g., application programmer interfaces – APIs) that applications 104 may invoke to access the underlying hardware of head unit 100.

OS 102 of head unit 100 may operate to perform techniques in accordance with this disclosure. However, it should be understood that an operating system of a computing device (e.g., a cellular phone, a smartphone, a desktop computer, a laptop computer, a tablet computer, a portable gaming device, a portable media player, an e-book reader, a watch (including a so-called smartwatch), a gaming controller, etc.) may operate to perform these techniques.

Applications 104 may represent third-party applications that a user of head unit 100 obtains via application store services provided by way of operating system 102. Applications 104 may extend the software functionality of head unit 100, where applications 104 may execute within an execution environment presented by operating system 102. Applications 104 may, as a few examples, provide gaming services (e.g., video games), email services, web browsing

services, texting and/or chat services, web conferencing services, video conferencing services, music services (including streaming music services), video services (including video streaming services), navigation services, word processing services, spreadsheet services, slide and/or presentation services, assistant services, text entry services, or any other service commonly provided by applications.

As further shown in FIG. 1, storage device 110 includes application library 112. Application library 112 may represent a collection of implementations of behavior that has a well-defined interface by which the behavior is invoked. For instance, application library 112 may store a set of implementation behaviors, such as configuration data, pre-written code, subroutines, values, classes, etc. In some examples, application library 112 represents a collection of non-volatile resources (e.g., configuration data, documentation, help data, message templates, pre-written code and subroutines, classes, values, type specifications, etc.).

In general, third-party developers ("developers") may develop applications 104 for head unit 100 of a vehicle. Applications 104 may need to satisfy various guidelines (relating to, e.g., consistency, predictability, customizability, etc.) to ensure driver safety and a positive user experience. As a result, developing applications 104 for head unit 100 may be inconvenient, time-consuming, and even difficult for developers.

In accordance with techniques of this disclosure, applications 104 may offload GUI rendering to OS 102. Applications 104 may invoke functions of application library 112 that allow applications 104 to generate GUI metadata including, for example, a GUI template (which may, e.g., specify a predetermined layout) and content for populating the GUI template. Via application library 112, applications 104 may register with OS 102 to receive corresponding tokens (e.g., access tokens) identifying applications 104. Applications 104 may provide OS 102

with the GUI metadata and specifications of GUI templates to request rendering of corresponding GUIs. OS 102 may expose an interface (e.g., defined by application library 112) for interacting with display 108 or some other display in communication with head unit 100 by which to receive, from applications 104, the GUI metadata and corresponding tokens. In this way, OS 102 may use the GUI metadata to render the corresponding GUIs for applications 104.

As noted above, applications 104 may invoke functions of application library 112 that allow applications 104 to generate GUI metadata. The GUI metadata may be in the form of a sequence of platform-specific instructions for creating a GUI template or in the form of platform-specific template descriptions for drawing the GUI template that are specific to the platform of head unit 100. In some examples, the GUI metadata may include resources, such as images, text, etc. The GUI metadata may also include user input elements, such as buttons, sliders, etc.

Thus, applications 104 may reference application library 112 to create a GUI template. The GUI template may be a pre-determined layout (in this way ensuring a consistent look and feel) and include standard GUI elements, such as grids, lists, tab bars, etc. Examples of various GUI templates are shown below in FIG. 2. In some examples, applications may invoke application library 112 to provide OS 102 a specification of the GUI templates as part of the request for OS 102 to render corresponding GUIs in accordance with techniques of this disclosure.

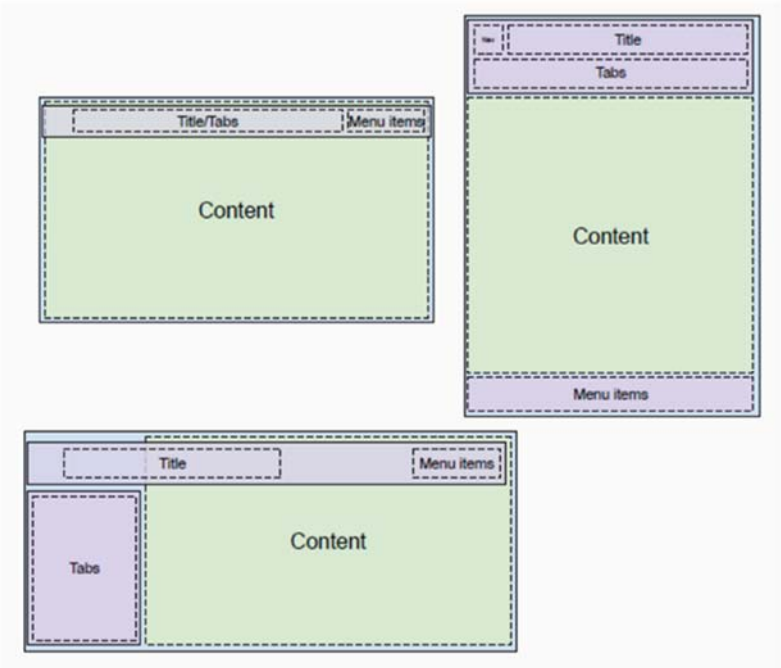

FIG. 2

One or more aspects (e.g., color theme, size, layout, etc.) of the GUI template may be customizable. For example, developers of applications 104 may customize one or more colors of the GUI elements of the GUI template. Examples of various customizations and implementations for the customizations are shown below in FIG. 3. As shown in FIG. 3, aspects that may be customizable include corner radius, text color, text font, border color, background color, etc. As further shown in FIG. 3, in some examples, customizations may be automatically generated or implemented based on prior customizations (referred to herein as “fallback customizations”). For instance, customizations for a toggle element may be automatically generated based on customizations by an original equipment manufacturer (OEM) for a button element. The OEM may modify the automatically generated customizations as desired for any of the elements of the GUI template. For instance, the OEM may replace the toggle element with the fallback customizations with an OEM coded replacement element.

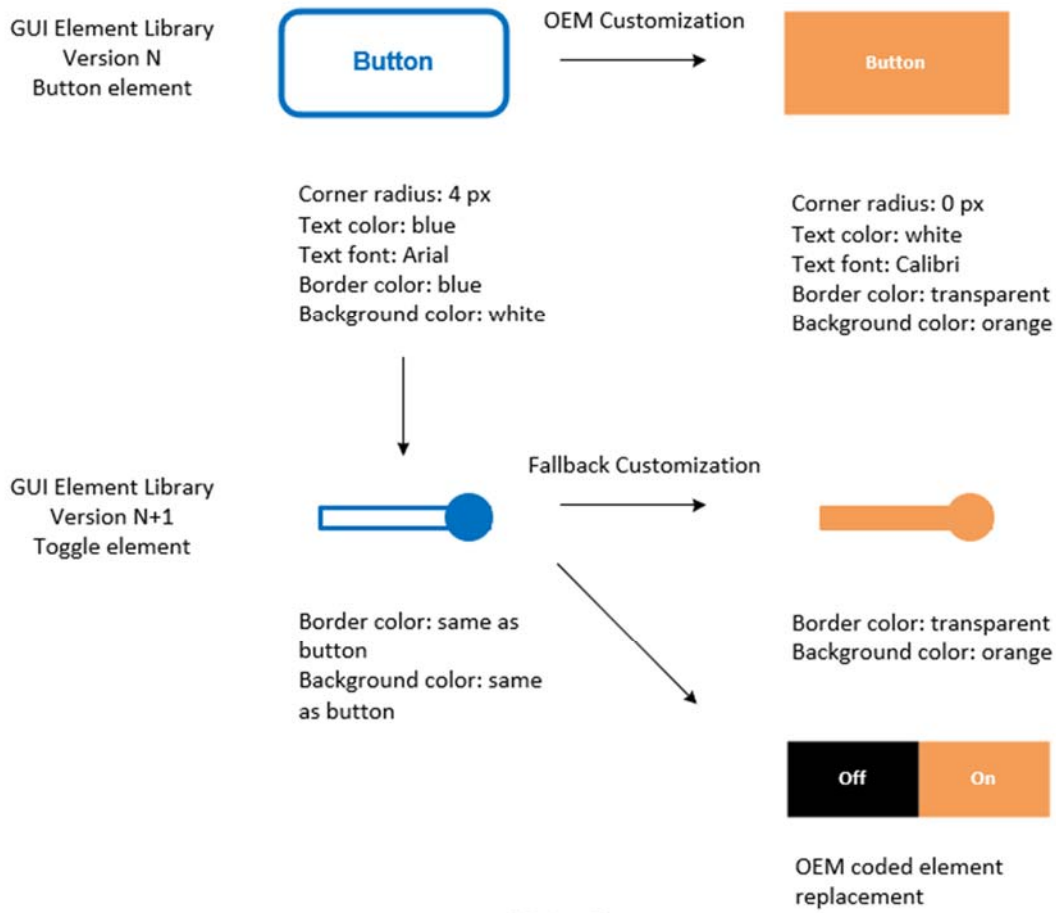

FIG. 3

Application library 112 may define an interface for interacting with OS 102. For instance, application 104 may send and receive messages from OS 102 via application library 112 to transfer the GUI metadata. By interacting with OS 102 via application library 112, applications 104 may cause OS 102 to render and, in some examples, present a GUI corresponding to the GUI metadata.

OS 102 may receive, from applications 104, tokens identifying applications 104. In some examples, the tokens may also control the access, security rights, privileges, etc., of applications 104. Applications 104 may send the tokens to OS 102 with the GUI metadata, thereby enabling OS 102 to determine which GUI has been rendered for which application. In some examples, each of applications 104 may initialize a placeholder GUI configured to be rendered into or

otherwise replaced by the GUI rendered by OS 102 and register a callback on the surface with a token. In accordance with the callbacks, OS 102 may send information to one or more of applications 104 indicating that OS 102 has bound the tokens obtained from applications 104 to an appropriate surface renderer managed by a renderer service 114 provided by OS 102, as shown below in FIG. 4. Renderer service 114 may manage the rendering requirements of applications 104 by creating and maintaining instances of screen renderers for each bounded application activity (sometimes referred to as "template activity"), forwarding incoming requests from template activity to the appropriate screen renderer, clearing or destroying the appropriate screen renderer once the template activity is unbounded, etc.

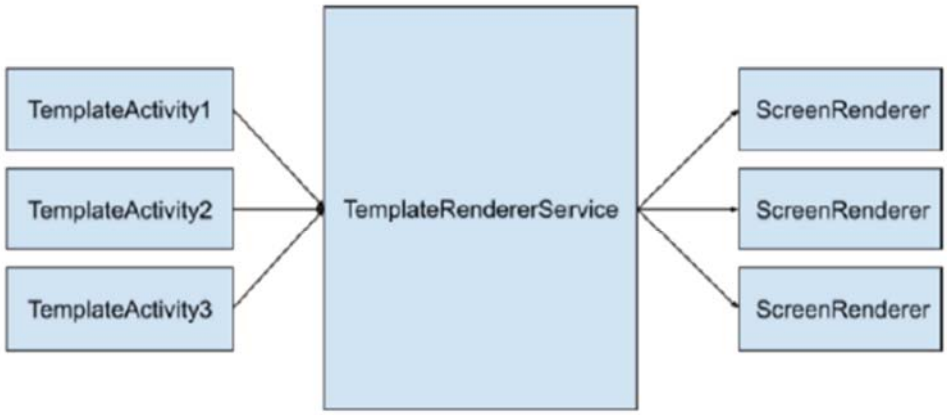

FIG. 4

OS 102 may render an updated GUI. For example, OS 102 may manage interactions (e.g., user interactions) with GUIs of applications 104 and transmit the interactions to applications 104. Responsive to receiving the interactions from OS 102, applications 104 may respond to update the GUIs by sending, to OS 102, updated GUI metadata defining one or more actions, events (e.g., lifecycle events), etc. Renderer service 114 may use the updated GUI

metadata to render the updated GUIs. Renderer service 114 may then replace the current GUIs with the updated GUIs. Replacing the current GUIs may involve clearing or otherwise destroying the current GUIs.

It is noted that the techniques of this disclosure may be combined with any other suitable technique or combination of techniques. As one example, the techniques of this disclosure may be combined with the techniques described in U.S. Patent Application Publication No. 2019/0339918A1. In another example, the techniques of this disclosure may be combined with the techniques described in U.S. Patent Application Publication No. 2015/0193090A1. In yet another example, the techniques of this disclosure may be combined with the techniques described in U.S. Patent Application Publication No. 2013/861614A1. In yet another example, the techniques of this disclosure may be combined with the techniques described in "Using the Android for Cars App Library," Google Android, June 21, 2021.