June 2023

APPLICATION EMULATOR FOR VEHICLES

Jacob Monsein Kandel

Saeid Farivar Asanjan

Bradley David Stenning

Assaf Kremer

Matias N. Fernandez Bustos

See next page for additional authors

Inventor(s)
Jacob Monsein Kandel, Saeid Farivar Asanjan, Bradley David Stenning, Assaf Kremer, Matias N. Fernandez Bustos, Gustavo Balestraci, and Arzu Ozdogan

APPLICATION EMULATOR FOR VEHICLES

ABSTRACT

The infotainment system (e.g., 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, train, a plane, a boat, a helicopter, a personal transport vehicle, etc.) may execute an application (e.g., a media application, a messaging application, a navigation application, etc.) originally developed for a different type of computing device, such as a smartphone, a tablet, a smartwatch, etc. To run the application, the infotainment system may load the files of the application into an emulator module (e.g., a virtual environment simulating the computing device for which the application was originally designed). The infotainment system may modify display values and other parameters within the emulator module to match the form factor and specifications of a display in the vehicle. In this way, the techniques may offer seamless integration as well as an expanded application ecosystem, significantly improving the user experience.

DESCRIPTION

FIG. 1 below is a conceptual diagram illustrating a vehicle 10 in accordance with techniques of this disclosure. Examples of vehicle 10 may include an automobile, a motorcycle, a bus, a recreational vehicle (RV), a semi-trailer truck, a tractor or other type of farm equipment, train, a plane, a boat, a helicopter, a personal transport vehicle, etc. As shown in FIG. 1, vehicle 10 includes an infotainment system 100 (e.g., a head unit). As further shown in FIG. 1, infotainment system 100 includes one or more processors 102, one or more displays 104 (“display 104”), and one or more storage devices 106 (“storage device 106”).

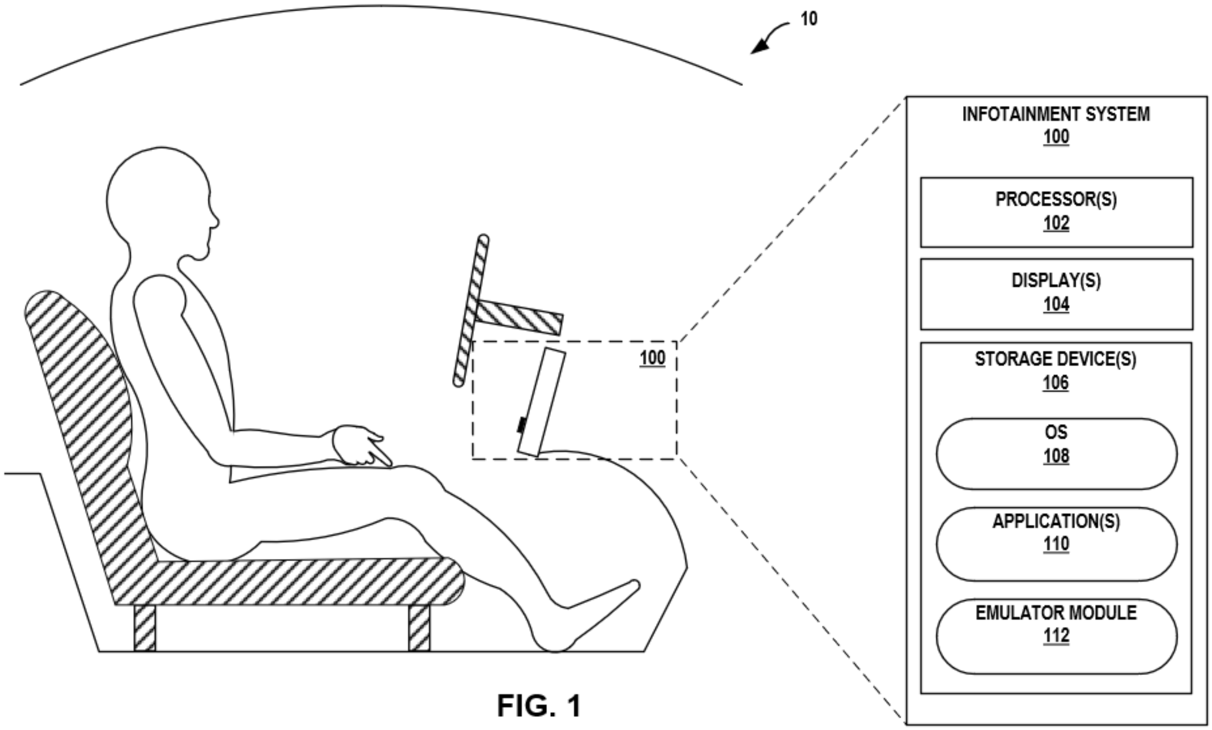

FIG. 1

Infotainment system 100 of vehicle 10 may operate to assist, inform, entertain, or otherwise provide for interactions with one or more occupants of vehicle 10. Infotainment system 100 may represent an integrated infotainment system that provides a user interface (UI), such as a voice user interface (VUI), a graphical user interface (GUI), etc. In general, infotainment system 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), a seating system (for controlling a position of a driver and/or passenger seat), etc.

Processors 102 may implement functionality and/or execute instructions associated with infotainment system 100. Examples of processors 102 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. In some examples, processors

102 may represent a system on a chip (SoC) that includes an integrated circuit for implementing one or more of the above referenced examples of processors 102, along with supporting memory and/or storage, and possibly various interfaces, modems, etc., as a single package.

Display 104 of infotainment system 100 may be a presence-sensitive display that functions as an input device and as an output device. For example, display 104 may function as an input device using a presence-sensitive input component, such as a resistive touchscreen, a surface acoustic wave touchscreen, a pressure-sensitive screen, an acoustic pulse recognition touchscreen, or another presence-sensitive display technology. Additionally, display 104 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, active-matrix organic light-emitting diode (AMOLED) display, etc.

Storage devices 106 of infotainment system 100 may include one or more computer-readable storage media. For example, storage devices 106 may be configured for long-term, as well as short-term storage of information, such as instructions, data, or other information used by infotainment system 100. In some examples, storage devices 106 may include non-volatile storage elements. Examples of such non-volatile storage elements include magnetic hard discs, optical discs, solid state discs, etc. 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 106 may include an operating system 108 (“OS 108”) that provides an execution environment for one or more applications, such as application 110A-110N (collectively, “application 110”). OS 108 may represent a multi-threaded operating system or a single-threaded operating system with which application 110 may interface to access hardware of infotainment system 100. OS 108 may include a kernel that facilitates access to the underlying hardware of infotainment system 100, where kernel may present a number of different interfaces (e.g., application programming interfaces – APIs) that application 110 may invoke to access the underlying hardware of infotainment system 100. Examples of application 110 may include a media application, a messaging application, a navigation application, a video conferencing application, a communication application, etc.

In general, different computing devices, such as smartphones, laptops, vehicles, etc., have different form factors, specifications, and interfaces. Consequently, the design and development of applications may require adjustment to ensure a satisfactory user experience for each computing device. For instance, if a developer of application 110 designs application 110 for a smartphone but does not adjust application 110 for execution by infotainment system 100 of vehicle 10, several issues may arise. As one example, infotainment system 100 may undesirably stretch application 110 to fill display 104, causing distortion and pixelation, or maintain the original size of application 110 and leave large areas of display 104 unused. In the former case, the GUI of application 110 may look unattractive and/or force the driver to reach (e.g., by leaning) for some UI elements, which can be dangerous. In the latter case, the UI elements may be inappropriately small (e.g., because smartphone displays are usually smaller than vehicle displays), requiring more precision (and therefore attention from the driver) to interact with.

In accordance with techniques of this disclosure, OS 108 may use an emulator module 112 to execute and display application 110. When a driver of vehicle 10 wants to run application 110, OS 108 may load the files of application 110 into emulator module 112. Additionally, OS 108 may modify display values and other parameters within emulator module 112 to match the form factor and specifications of display 104. In this way, the techniques may offer seamless integration (e.g., by allowing a driver to use application 110 within vehicle 10 without requiring developers to create separate versions of application 110 for different car infotainment systems) as well as an expanded application ecosystem (e.g., by giving infotainment system 100 access to a much larger range of applications, as emulator module 112 may execute potentially any application for any type of computing device). Notably, the techniques described here may apply to single-display environments as well as multi-display environments, such as when two or more displays come together to create the appearance of a larger display.

Emulator module 112 may represent a virtual environment that simulates the software conditions (and optionally the hardware conditions) of a different computing device. For example, emulator module 112 may simulate the software conditions of a smartphone. As described above, when a driver of vehicle 10 wants to run application 110, OS 108 may load the files of application 110 into emulator module 112. The files may include important application data, including code, resources, assets, manifest files, etc.

Regarding display modifications, OS 108 may provide display values for display 104 and/or other parameters (e.g., resolution, aspect ratio, color depth, etc.) to emulator module 112. In turn, emulator module 112 may adjust the size, resolution, etc., of application 110 with respect to the screen size of display 104. In some examples, the adjustments by emulator module 112 may be pre-determined (e.g., by the original equipment manufacturer (OEM) of vehicle 10). In some examples, the driver may provide inputs to adjust the size, resolution, display density, etc., of application 110.

Thus, emulator module 112 may represent an adaptive environment for application 110. For example, without requiring the developer of application 110 to write any code, emulator module 112 may alter the size of application 110 in real-time based on predefined rules or conditions (e.g., in accordance with the preferences of the OEM). In some examples, emulator module 112 may automatically adjust how application 110 appears based on the specifications of display 104, user input, the states or modes of vehicle 10 (e.g., when parked versus driving), etc.

In some examples, emulator module 112 may cause application 110 to enter an immersive mode. When in the immersive mode, the GUI of application 110 may be enlarged (which may or may not be fullscreen). To use immersive mode, the driver may provide an input, such as by performing a specific gesture on display 104 (e.g., pinching out, double tapping, long pressing, etc.), pressing a specific button or icon within the GUI, giving a voice command (e.g., “enter immersive mode”), and so on. In any case, immersive mode may advantageously provide more space to display content, which may be particularly beneficial for applications with rich visual content, like media players. In this way, immersive mode may facilitate a more engaging, enjoyable user experience.

Emulator module 112 may include one or more components in the GUI to improve the user experience. For example, emulator module 112 may include a toolbar that includes a back button. The back button may allow the driver to return to the previous screen or step of application 110, making navigation of application 110 simpler and more intuitive. Having key functions like 'back' easily accessible in the toolbar may reduce the need for complex gestures or sequences of interactions, decreasing driver distraction. The location and orientation of the toolbar may be customizable (e.g., by the OEM, by the driver, etc.).

Emulator module 112 may enable customization of the size, style, placement, etc., of the GUI for application 110 to facilitate a user-friendly and safer driving environment. For example, Emulator module 112 may cause display 104 to display UI elements of application 110 within easy reach of the driver without the driver having to stretch or lean, which may be distracting or even dangerous while driving. Additionally, emulator module 112 may enlarge UI elements to make the UI elements easier to see and interact with, reducing the amount of time a driver diverts their eyes from the road and/or removes their hands from the wheel.

FIGS. 2A-2B below are conceptual diagrams illustrating GUIs created by an emulator module of an infotainment system, in accordance with techniques disclosed here. FIG. 2A may correspond to a vehicle display with a long vertical dimension, and FIG. 2B may correspond to a vehicle display with a long horizontal dimension. As shown in FIG. 2A-2B, each of the GUIs may include a toolbar, the orientation of which is customizable. The toolbar may include buttons and other controls, such as a back button. Notwithstanding differences in size, the GUIs created by emulator module 112 may be substantially similar to the GUI of application 110 when displayed by a mobile computing device (e.g., in terms of aspect ratio, GUI elements, etc.).

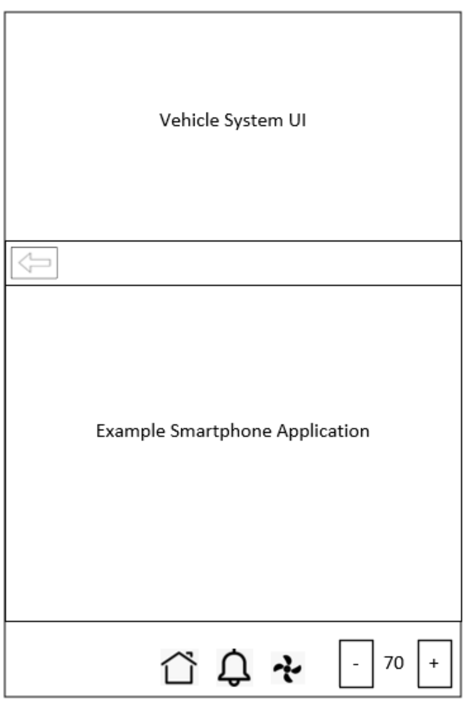

FIG. 2A

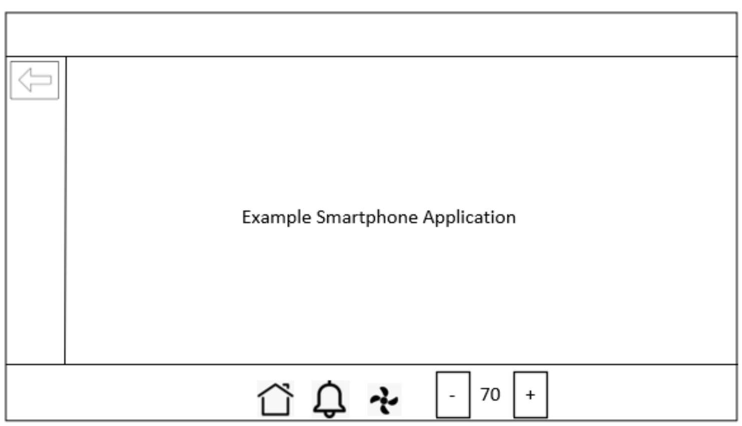

FIG. 2B

The techniques of this disclosure include one or more advantages. For example, infotainment system 100 may expand the application ecosystem for vehicle 10 in a cost-efficient manner because the techniques do not require developers to develop and maintain separate versions of their applications specifically for infotainment system 100. Indeed, developers may use the techniques to make their applications compatible with vehicle 10 without writing any additional code. Additionally, by automatically adjusting the GUI of application 110 (e.g., based on OEM preferences), infotainment system 100 may ensure that application 110 provides a consistent, desirable, and safe user experience while driving. Moreover, infotainment system 100 may enable extensive customization to accommodate not only different infotainment systems, but also different user preferences. Accordingly, the techniques may allow for a safer, more comfortable, and more personalized user experience.

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. 2020/0219469A1. In another example, the techniques of this disclosure may be combined with the techniques described in U.S. Patent Application Publication No. 2020/0307377A1. In yet another example, the techniques of this disclosure may be combined with the techniques described in U.S. Patent Application Publication No. 2019/0356773A1. In yet another example, the techniques of this disclosure may be combined with the techniques described in Mishaal Rahman, "Android 12L changelog: Every new feature for Android tablets," https://blog.esper.io/android-12l-deep-dive/, July, 2022.