



(51) International Patent Classification:

G06F 3/01 (2006.01) G06F 9/54 (2006.01)
G06F 9/451 (2018.01) G09G 5/14 (2006.01)
G06F 9/48 (2006.01)

(21) International Application Number:

PCT/US2023/035591

(22) International Filing Date:

20 October 2023 (20.10.2023)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

63/380,406 21 October 2022 (21.10.2022) US
18/068,102 19 December 2022 (19.12.2022) US

(71) Applicant: **META PLATFORMS TECHNOLOGIES, LLC** [US/US]; 1 Meta Way, Menlo Park, California 94025 (US).

(72) Inventors: **FORBES JR., Ronald Omega**; 1 Meta Way, Menlo Park, California 94025 (US). **FURTWANGLER, Nathan**; 1 Meta Way, Menlo Park, California 94025 (US). **LUTHER, Matthew**; 1 Meta Way, Menlo Park, California 94025 (US). **CUERVO LAFFAYE, Eduardo Alberto**; 1 Meta Way, Menlo Park, California 94025 (US).

(74) Agent: **COLBY, Steven** et al.; Rimon Law, 420 West Main Street, Suite 101B, Boise, Idaho 83702 (US).

(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, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, MG, MK, MN, MU, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, CV, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, 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, ME, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

(54) Title: APPLICATION MULTITASKING IN A THREE-DIMENSIONAL ENVIRONMENT

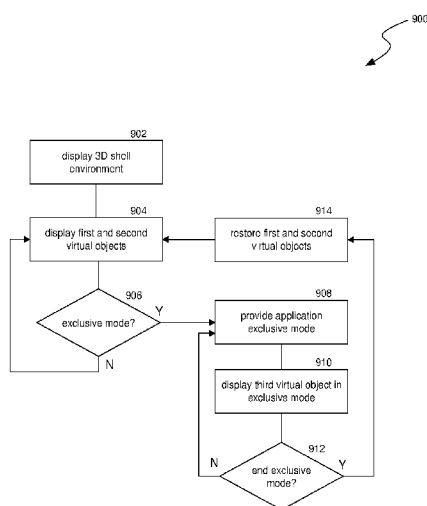


FIG. 9

(57) Abstract: Aspects of the present disclosure are directed to application multitasking in a shell with a three-dimensional environment. Implementations immerse a user in a three-dimensional environment via an artificial reality system, such as a display environment for a system shell. The system shell can execute applications (e.g., system shell applications, remoted applications, etc.). An executing application can correspond to a displayed virtual object (e.g., panel). The system shell can concurrently execute two, three, or more applications and the three-dimensional environment can concurrently display two, three, or more corresponding virtual objects that display contents for the executing applications. Implementations of a mode manager can manage a mode for the three-dimensional environment/system shell. Example modes include cooperative mode and exclusive mode. Implementations of cooperative mode permit concurrent display of multiple virtual objects from different applications while the exclusive mode permits display of virtual objects only from the executing application entering exclusive mode.

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))*

1 APPLICATION MULTITASKING IN A
2 THREE-DIMENSIONAL ENVIRONMENT

3 TECHNICAL FIELD

4 The present disclosure is directed to application multitasking in a shell with a
5 three-dimensional environment.

6 BACKGROUND

7 Artificial reality systems have grown in popularity and this trend is expected to
8 accelerate. Immersive artificial reality environments can provide unique experiences
9 and support virtual social interactions among users. However, user interactions with
10 executing applications in a three-dimensional environment often follow rigid
11 conventions. For example, XR systems often omit concurrent execution and display
12 of multiple executing applications. In addition, switching between multiple applications
13 often involves a manual user workflow with several steps, such as terminating
14 execution of a current application before executing a new application.

15 SUMMARY

16 According to an aspect, there is provided a method for application multitasking
17 in a shell with a three-dimensional artificial reality, XR, environment, the method
18 comprising: displaying the XR environment comprising a first virtual object that
19 displays content for a first application executing in the shell, and a second virtual object
20 that displays content for a second application executing in the shell; triggering an
21 exclusive mode for the first application, the exclusive mode preventing applications,
22 other than the first application, from displaying world-locked content, wherein, during
23 the exclusive mode, the first and second virtual objects are removed from display, and
24 the second application is transitioned to a background state; redisplaying, in response
25 to termination of the exclusive mode, the first virtual object and the second virtual
26 object in the XR environment by: transitioning the second application from the
27 background state to executing in the shell; and displaying content for the first
28 application executing in the shell at the first virtual object and content for the second
29 application executing in the shell at the second virtual object.

30 In one embodiment, when displayed or redisplayed, the first virtual object and
31 the second virtual object are displayed side-by-side in a predefined near-field region
32 of the XR environment relative to the user.

33 In one embodiment, the near-field region of the XR environment comprises a
34 plurality of nodes positioned side-by-side and, when displayed, each of the first virtual

1 object and second virtual object is assigned to one of the nodes.

2 In one embodiment, in exclusive mode, content for the first application is
3 dynamically moved to a third virtual object displayed in the exclusive region of the XR
4 environment.

5 In one embodiment, the exclusive region is closer in proximity to the user than
6 the near-field region in the XR environment.

7 In one embodiment, the triggered exclusive mode for the first application
8 provides the first application use of at least a designated exclusive region of the XR
9 environment, and, in exclusive mode, content for the first application is dynamically
10 moved to a third virtual object displayed in the exclusive region of the XR environment.

11 In one embodiment, the designated exclusive region of the XR environment is
12 closer in proximity to the user than the near-field region in the XR environment.

13 In one embodiment, the first virtual object, second virtual object, and third virtual
14 object comprise two-dimensional panels.

15 In one embodiment: the XR environment displayed to the user comprises a
16 shell XR environment, in exclusive mode, the content for the first application comprises
17 a first application XR environment, in exclusive mode, the shell XR environment is
18 replaced with the first application XR environment, and in response to terminating the
19 exclusive mode, the first application XR environment is replaced with the shell XR
20 environment.

21 In one embodiment, the method further comprises: storing, in response to
22 triggering the exclusive mode for the first application, state information for the second
23 executing application and the second virtual object, wherein the redisplaying the
24 second virtual object in the XR environment comprises restoring the second virtual
25 object using the stored state information.

26 In one embodiment, the stored state information comprises a content state for
27 the second virtual object and an execution state for the second executing application.

28 In one embodiment, redisplaying the second virtual object in the XR
29 environment comprises restoring, using the stored state information, content displayed
30 at the second virtual object and an execution state for the second application.

31 In one embodiment, said displaying the XR environment comprises displaying
32 the XR environment to a user via an XR device; said triggering the exclusive mode is
33 in response to a request; and the exclusive mode provides the first application use of
34 at least a designated exclusive region of the XR environment.

1 According to another aspect, there is provided a computer-readable storage
2 medium storing instructions that, when executed by a computing system, cause the
3 computing system to perform the method set out above.

4 According to a further embodiment, there is provided a computing system for
5 application multitasking in a shell with a three-dimensional artificial reality, XR,
6 environment, the computing system comprising: one or more processors; and one or
7 more memories storing instructions that, when executed by the one or more
8 processors, cause the computing system to perform the method set out above.

9 It will be appreciated that any features described herein as being suitable for
10 incorporation into one or more aspects or embodiments of the present disclosure are
11 intended to be generalizable across any and all aspects and embodiments of the
12 present disclosure. Other aspects of the present disclosure can be understood by
13 those skilled in the art in light of the description, the claims, and the drawings of the
14 present disclosure. The foregoing general description and the following detailed
15 description are exemplary and explanatory only and are not restrictive of the claims.

16 BRIEF DESCRIPTION OF THE DRAWINGS

17 Figure 1 is a block diagram illustrating an overview of devices on which some
18 implementations of the present technology can operate.

19 Figure 2A is a wire diagram illustrating a virtual reality headset which can be
20 used in some implementations of the present technology.

21 Figure 2B is a wire diagram illustrating a mixed reality headset which can be
22 used in some implementations of the present technology.

23 Figure 2C is a wire diagram illustrating controllers which, in some
24 implementations, a user can hold in one or both hands to interact with an artificial
25 reality environment.

26 Figure 3 is a block diagram illustrating an overview of an environment in which
27 some implementations of the present technology can operate.

28 Figure 4 is a block diagram illustrating components which, in some
29 implementations, can be used in a system employing the disclosed technology.

30 Figures 5 is a diagram of a three-dimensional environment for a system shell
31 with concurrently executing applications.

32 Figures 6A and 6B are diagrams of three-dimensional environments for a
33 system shell with multiple virtual object displays.

34 Figure 7A is a diagram of a three-dimensional environment with a theater

1 display.

2 Figure 7B is a diagram of a three-dimensional environment illustrating multiple
3 display locations.

4 Figure 8 is a diagram of a shared three-dimensional environment with a theater
5 display.

6 Figure 9 is a flow diagram illustrating a process used in some implementations
7 of the present technology for application multitasking in a shell with a
8 three-dimensional environment.

9 The techniques introduced here may be better understood by referring to the
10 following Detailed Description in conjunction with the accompanying drawings, in
11 which like reference numerals indicate identical or functionally similar elements.

12 DETAILED DESCRIPTION

13 Aspects of the present disclosure are directed to application multitasking in a
14 shell with a three-dimensional environment. Implementations immerse a user in a
15 three-dimensional environment via an artificial reality system. The three-dimensional
16 environment can be the display environment for a system shell. Implementations of
17 the system shell execute applications (e.g., system shell applications and applications
18 remote from the system shell). For example, an executing application can correspond
19 to a two-dimensional virtual object (e.g., panel) displayed in the three-dimensional
20 environment. In some implementations, the system shell can concurrently execute
21 two, three, or more applications and the three-dimensional environment can
22 concurrently display two, three, or more corresponding virtual objects that display
23 contents for the executing applications. The concurrently displayed virtual objects can
24 be side-by-side panel displays in some implementations. Example applications
25 include web browsers, music players, video players, social media applications,
26 messaging or other communication applications, third-party applications,
27 streaming/casting applications, a content library application, or any other suitable
28 application.

29 Virtual object displays in the three-dimensional environment can comprise
30 display components for executing applications. For example, the content for an
31 executing application is displayed in its corresponding virtual object. Virtual objects
32 can display a webpage for a browser application, an interface for playing music or a
33 video, messages with other users, or any other suitable content. The user can interact
34 with executing applications using their virtual object displays, such as via hand-held

1 controller input (e.g., sensed motion, button presses, etc.), sensed user motion (e.g.,
2 virtual hands), or any other suitable user input.

3 In some implementations, a near-field (e.g., predefined region or volume of the
4 three-dimensional environment) can comprise a set of nodes that can each host a
5 virtual object in cooperative mode. The near-field can comprise the surface of a
6 cylinder that surrounds the user (or a portion of the cylindrical surface) such that that
7 the near-field is a first distance from the user. In some implementations, the nodes
8 can be side-by-side on the near-field so that when the nodes host virtual objects, they
9 are displayed side-by-side. For example, the near-field can display two, three, or more
10 side-by-side two-dimensional virtual objects (e.g., panels) at a first distance from the
11 user, and each virtual object can display contents for a concurrently executing
12 application. In some implementations, when a given application begins executing
13 (e.g., via user input, or any other suitable trigger) or a background application is
14 restored, the given application can be assigned to one of the unassigned nodes of the
15 near-field. A user can also select a virtual object that corresponds to an executing
16 application (e.g., grab a virtual object via user input, such as a pinch) and position the
17 virtual object over an unassigned node at the near-field to assign the virtual
18 object/executing application to the node.

19 Implementations of the system shell can store the state of executing
20 applications and the shell state of their corresponding two-dimensional virtual objects.
21 For example, when an application is transitioned to background status, such as when
22 an exclusive mode is entered, or the system shell is transitioned to inactive (e.g., when
23 a user initiates an XR application that triggers display of a new XR environment, ends
24 a user session with the XR system, etc.), the application state of executing
25 application(s) and the shell state (e.g., position, size, etc.) of their corresponding two-
26 dimensional virtual object(s) can be stored. The state storage responsibilities can be
27 divided among the system shell and the individual applications. For example, the shell
28 state stored by the system shell can include: size and position (e.g., within the three-
29 dimensional environment) of the corresponding two-dimensional virtual object(s),
30 application specific display parameters (e.g., tab orientation, relevant
31 websites/webpages, etc.), and any other suitable state parameter(s) that initialize an
32 application. State stored by the individual applications can include: specific state of
33 content displayed at the time the application is closed (e.g., open tab, scroll position,
34 open webpage, open messages in a communication application, etc.), and any other

1 suitable application state parameter.

2 When the application(s) are restored (e.g., exclusive mode is terminated, the
3 system shell returns to an active state, etc.), implementations of the system shell can
4 restore/initialize display of virtual objects for the application(s) according to the system
5 shell's stored shell state (e.g., position and size of the virtual object, initialization
6 parameters for the application, etc.) and the application itself can restore the content
7 displayed at the virtual object(s) according to the application's stored state. In
8 combination, this state save and restoration workflow provides the user a seamless
9 experience. For example, when an executing application is transitioned into the
10 background having a given state (e.g., virtual object position and size, content
11 displayed in the virtual object, etc.), the application can be restored in the same given
12 state (without any additional interaction with or management by the user) upon
13 transitioning into active execution again. Similarly, when a system shell is transitioned
14 to inactive (e.g., its three-dimensional virtual environment is no longer displayed), a
15 given state for each of its executing applications can be stored. Upon reactivation of
16 the system shell (e.g., redisplay of the three-dimensional environment), each
17 application can be restored to its given state to provide a seamless restoration of the
18 state for the entire system shell.

19 Implementations of a mode manager can manage a mode for the
20 three-dimensional environment/system shell. Example modes include cooperative
21 mode and exclusive mode. Implementations of cooperative mode permit concurrent
22 execution of multiple applications and display of multiple virtual objects, such as
23 multiple panels in a near-field that display the contents of multiple executing
24 applications. In some examples, the mode manager can initiate exclusive mode for
25 an executing application, for example according to user input, based on a request from
26 the application, or in any other suitable manner. Implementations of the exclusive
27 mode permit display of virtual objects only from the executing application entering
28 exclusive mode, for example at a location in the three-dimensional environment closer
29 to the user than the near-field region, or at any other suitable location. When exclusive
30 mode is initiated, executing applications other than the application entering exclusive
31 mode are transitioned to the background and their states are stored. In addition, the
32 virtual object displays for these applications are no longer displayed until exclusive
33 mode is exited. In some implementations, when a given application enters exclusive
34 mode, applications other than the given application are prevented from displayed

1 world-locked content (e.g., content locked to a real-world location while displayed in
2 the XR environment).

3 Implementations of the three-dimensional environment include a theater screen
4 at a far-field region or volume. For example, the far-field region can be a cylindrical
5 surface (or portion of a cylindrical surface) that is further from the user than the near-
6 field. The theater screen can display content for any suitable application assigned to
7 the theater screen. In some implementations, the theater screen can be active in
8 cooperative mode or in exclusive mode. For example, the theater screen can display
9 a video while a user interacts with other executing application(s) via their
10 corresponding virtual object(s). In some implementations, a theater screen can be
11 shared among multiple users. For example, a video can be cast to a theater screen
12 that is simultaneously displayed to multiple users via multiple XR systems. In this
13 example, each user can be immersed in a three-dimensional environment (e.g., that
14 corresponds to a system shell), and the theater screen can be displayed in the far-
15 field region for each user. In other examples, the theater screen can simultaneously
16 display video to the multiple users in any other suitable manner.

17 Embodiments of the disclosed technology may include or be implemented in
18 conjunction with an artificial reality system. Artificial reality or extra reality (XR) is a
19 form of reality that has been adjusted in some manner before presentation to a user,
20 which may include, e.g., virtual reality (VR), augmented reality (AR), mixed reality
21 (MR), hybrid reality, or some combination and/or derivatives thereof. Artificial reality
22 content may include completely generated content or generated content combined
23 with captured content (e.g., real-world photographs). The artificial reality content may
24 include video, audio, haptic feedback, or some combination thereof, any of which may
25 be presented in a single channel or in multiple channels (such as stereo video that
26 produces a three-dimensional effect to the viewer). Additionally, in some
27 embodiments, artificial reality may be associated with applications, products,
28 accessories, services, or some combination thereof, that are, e.g., used to create
29 content in an artificial reality and/or used in (e.g., perform activities in) an artificial
30 reality. The artificial reality system that provides the artificial reality content may be
31 implemented on various platforms, including a head-mounted display (HMD)
32 connected to a host computer system, a standalone HMD, a mobile device or
33 computing system, a "cave" environment or other projection system, or any other
34 hardware platform capable of providing artificial reality content to one or more viewers.

1 "Virtual reality" or "VR," as used herein, refers to an immersive experience
2 where a user's visual input is controlled by a computing system. "Augmented reality"
3 or "AR" refers to systems where a user views images of the real world after they have
4 passed through a computing system. For example, a tablet with a camera on the back
5 can capture images of the real world and then display the images on the screen on
6 the opposite side of the tablet from the camera. The tablet can process and adjust or
7 "augment" the images as they pass through the system, such as by adding virtual
8 objects. "Mixed reality" or "MR" refers to systems where light entering a user's eye is
9 partially generated by a computing system and partially composes light reflected off
10 objects in the real world. For example, a MR headset could be shaped as a pair of
11 glasses with a pass-through display, which allows light from the real world to pass
12 through a waveguide that simultaneously emits light from a projector in the MR
13 headset, allowing the MR headset to present virtual objects intermixed with the real
14 objects the user can see. "Artificial reality," "extra reality," or "XR," as used herein,
15 refers to any of VR, AR, MR, or any combination or hybrid thereof.

16 The shell component of conventional XR systems fails to support multi-
17 application functionality. For example, XR systems often omit concurrent execution
18 and display of multiple executing applications. In addition, switching between multiple
19 applications often involves a manual user workflow with several steps, such as
20 terminating execution of a current application before executing a new application.

21 Implementations provide a shell XR environment and system shell that
22 executes applications (e.g., system shell applications and applications remote from
23 the system shell). For example, an executing application can correspond to a
24 two-dimensional virtual object displayed in the XR environment. The system shell can
25 concurrently execute two, three, or more applications and the XR environment can
26 concurrently display two, three, or more corresponding virtual objects that display
27 contents for the executing applications. The concurrently displayed virtual objects can
28 be side-by-side panel displays in some implementations.

29 In addition, implementations of the system shell can store the state of executing
30 applications and the shell state of their corresponding two-dimensional virtual objects.
31 For example, when an executing application is transitioned to background status, the
32 application state of the executing application and the shell state of its corresponding
33 two-dimensional virtual object can be stored. When the application transitions back to
34 active execution, implementations of the system shell can restore/initialize display of

1 virtual object for the application according to the system shell's stored shell state and
2 the application can restore the content displayed at the virtual object according to the
3 application's stored state. Accordingly, implementations provide mechanisms for
4 concurrent execution of applications in the shell environment and efficient switching
5 between executing applications.

6 Several implementations are discussed below in more detail in reference to the
7 figures. Figure 1 is a block diagram illustrating an overview of devices on which some
8 implementations of the disclosed technology can operate. The devices can comprise
9 hardware components of a computing system 100 that provide application multitasking
10 in a shell with a three-dimensional environment. In various implementations,
11 computing system 100 can include a single computing device 103 or multiple
12 computing devices (e.g., computing device 101, computing device 102, and computing
13 device 103) that communicate over wired or wireless channels to distribute processing
14 and share input data. In some implementations, computing system 100 can include a
15 stand-alone headset capable of providing a computer created or augmented
16 experience for a user without the need for external processing or sensors. In other
17 implementations, computing system 100 can include multiple computing devices such
18 as a headset and a core processing component (such as a console, mobile device, or
19 server system) where some processing operations are performed on the headset and
20 others are offloaded to the core processing component. Example headsets are
21 described below in relation to Figures 2A and 2B. In some implementations, position
22 and environment data can be gathered only by sensors incorporated in the headset
23 device, while in other implementations one or more of the non-headset computing
24 devices can include sensor components that can track environment or position data.

25 Computing system 100 can include one or more processor(s) 110 (e.g., central
26 processing units (CPUs), graphical processing units (GPUs), holographic processing
27 units (HPUs), etc.) Processors 110 can be a single processing unit or multiple
28 processing units in a device or distributed across multiple devices (e.g., distributed
29 across two or more of the computing devices 101-103).

30 Computing system 100 can include one or more input devices 120 that provide
31 input to the processors 110, notifying them of actions. The actions can be mediated
32 by a hardware controller that interprets the signals received from the input device and
33 communicates the information to the processors 110 using a communication protocol.
34 Each input device 120 can include, for example, a mouse, a keyboard, a touchscreen,

1 a touchpad, a wearable input device (e.g., a haptics glove, a bracelet, a ring, an
2 earring, a necklace, a watch, etc.), a camera (or other light-based input device, e.g.,
3 an infrared sensor), a microphone, or other user input devices.

4 Processors 110 can be coupled to other hardware devices, for example, with
5 the use of an internal or external bus, such as a PCI bus, SCSI bus, or wireless
6 connection. The processors 110 can communicate with a hardware controller for
7 devices, such as for a display 130. Display 130 can be used to display text and
8 graphics. In some implementations, display 130 includes the input device as part of
9 the display, such as when the input device is a touchscreen or is equipped with an eye
10 direction monitoring system. In some implementations, the display is separate from
11 the input device. Examples of display devices are: an LCD display screen, an LED
12 display screen, a projected, holographic, or augmented reality display (such as a
13 heads-up display device or a head-mounted device), and so on. Other I/O devices
14 140 can also be coupled to the processor, such as a network chip or card, video chip
15 or card, audio chip or card, USB, firewire or other external device, camera, printer,
16 speakers, CD-ROM drive, DVD drive, disk drive, etc.

17 In some implementations, input from the I/O devices 140, such as cameras,
18 depth sensors, IMU sensor, GPS units, LiDAR or other time-of-flights sensors, etc.
19 can be used by the computing system 100 to identify and map the physical
20 environment of the user while tracking the user's location within that environment. This
21 simultaneous localization and mapping (SLAM) system can generate maps (e.g.,
22 topologies, grids, etc.) for an area (which may be a room, building, outdoor space,
23 etc.) and/or obtain maps previously generated by computing system 100 or another
24 computing system that had mapped the area. The SLAM system can track the user
25 within the area based on factors such as GPS data, matching identified objects and
26 structures to mapped objects and structures, monitoring acceleration and other
27 position changes, etc.

28 Computing system 100 can include a communication device capable of
29 communicating wirelessly or wire-based with other local computing devices or a
30 network node. The communication device can communicate with another device or a
31 server through a network using, for example, TCP/IP protocols. Computing system
32 100 can utilize the communication device to distribute operations across multiple
33 network devices.

34 The processors 110 can have access to a memory 150, which can be contained

1 on one of the computing devices of computing system 100 or can be distributed across
2 of the multiple computing devices of computing system 100 or other external devices.
3 A memory includes one or more hardware devices for volatile or non-volatile storage,
4 and can include both read-only and writable memory. For example, a memory can
5 include one or more of random access memory (RAM), various caches, CPU registers,
6 read-only memory (ROM), and writable non-volatile memory, such as flash memory,
7 hard drives, floppy disks, CDs, DVDs, magnetic storage devices, tape drives, and so
8 forth. A memory is not a propagating signal divorced from underlying hardware; a
9 memory is thus non-transitory. Memory 150 can include program memory 160 that
10 stores programs and software, such as an operating system 162, application manager
11 164, and other application programs 166. Memory 150 can also include data memory
12 170 that can include, e.g., virtual object data, application data, environment or
13 application state data, configuration data, settings, user options or preferences, etc.,
14 which can be provided to the program memory 160 or any element of the computing
15 system 100.

16 Some implementations can be operational with numerous other computing
17 system environments or configurations. Examples of computing systems,
18 environments, and/or configurations that may be suitable for use with the technology
19 include, but are not limited to, XR headsets, personal computers, server computers,
20 handheld or laptop devices, cellular telephones, wearable electronics, gaming
21 consoles, tablet devices, multiprocessor systems, microprocessor-based systems,
22 set-top boxes, programmable consumer electronics, network PCs, minicomputers,
23 mainframe computers, distributed computing environments that include any of the
24 above systems or devices, or the like.

25 Figure 2A is a wire diagram of a virtual reality head-mounted display (HMD)
26 200, in accordance with some embodiments. The HMD 200 includes a front rigid body
27 205 and a band 210. The front rigid body 205 includes one or more electronic display
28 elements of an electronic display 245, an inertial motion unit (IMU) 215, one or more
29 position sensors 220, locators 225, and one or more compute units 230. The position
30 sensors 220, the IMU 215, and compute units 230 may be internal to the HMD 200
31 and may not be visible to the user. In various implementations, the IMU 215, position
32 sensors 220, and locators 225 can track movement and location of the HMD 200 in
33 the real world and in an artificial reality environment in three degrees of freedom
34 (3DoF) or six degrees of freedom (6DoF). For example, the locators 225 can emit

1 infrared light beams which create light points on real objects around the HMD 200. As
2 another example, the IMU 215 can include e.g., one or more accelerometers,
3 gyroscopes, magnetometers, other non-camera-based position, force, or orientation
4 sensors, or combinations thereof. One or more cameras (not shown) integrated with
5 the HMD 200 can detect the light points. Compute units 230 in the HMD 200 can use
6 the detected light points to extrapolate position and movement of the HMD 200 as well
7 as to identify the shape and position of the real objects surrounding the HMD 200.

8 The electronic display 245 can be integrated with the front rigid body 205 and
9 can provide image light to a user as dictated by the compute units 230. In various
10 embodiments, the electronic display 245 can be a single electronic display or multiple
11 electronic displays (e.g., a display for each user eye). Examples of the electronic
12 display 245 include: a liquid crystal display (LCD), an organic light-emitting diode
13 (OLED) display, an active-matrix organic light-emitting diode display (AMOLED), a
14 display including one or more quantum dot light-emitting diode (QOLED) sub-pixels, a
15 projector unit (e.g., microLED, LASER, etc.), some other display, or some combination
16 thereof.

17 In some implementations, the HMD 200 can be coupled to a core processing
18 component such as a personal computer (PC) (not shown) and/or one or more
19 external sensors (not shown). The external sensors can monitor the HMD 200 (e.g.,
20 via light emitted from the HMD 200) which the PC can use, in combination with output
21 from the IMU 215 and position sensors 220, to determine the location and movement
22 of the HMD 200.

23 Figure 2B is a wire diagram of a mixed reality HMD system 250 which includes
24 a mixed reality HMD 252 and a core processing component 254. The mixed reality
25 HMD 252 and the core processing component 254 can communicate via a wireless
26 connection (e.g., a 60 GHz link) as indicated by link 256. In other implementations,
27 the mixed reality system 250 includes a headset only, without an external compute
28 device or includes other wired or wireless connections between the mixed reality HMD
29 252 and the core processing component 254. The mixed reality HMD 252 includes a
30 pass-through display 258 and a frame 260. The frame 260 can house various
31 electronic components (not shown) such as light projectors (e.g., LASERs, LEDs,
32 etc.), cameras, eye-tracking sensors, MEMS components, networking components,
33 etc.

34 The projectors can be coupled to the pass-through display 258, e.g., via optical

1 elements, to display media to a user. The optical elements can include one or more
2 waveguide assemblies, reflectors, lenses, mirrors, collimators, gratings, etc., for
3 directing light from the projectors to a user's eye. Image data can be transmitted from
4 the core processing component 254 via link 256 to HMD 252. Controllers in the HMD
5 252 can convert the image data into light pulses from the projectors, which can be
6 transmitted via the optical elements as output light to the user's eye. The output light
7 can mix with light that passes through the display 258, allowing the output light to
8 present virtual objects that appear as if they exist in the real world.

9 Similarly to the HMD 200, the HMD system 250 can also include motion and
10 position tracking units, cameras, light sources, etc., which allow the HMD system 250
11 to, e.g., track itself in 3DoF or 6DoF, track portions of the user (e.g., hands, feet, head,
12 or other body parts), map virtual objects to appear as stationary as the HMD 252
13 moves, and have virtual objects react to gestures and other real-world objects.

14 Figure 2C illustrates controllers 270 (including controller 276A and 276B),
15 which, in some implementations, a user can hold in one or both hands to interact with
16 an artificial reality environment presented by the HMD 200 and/or HMD 250. The
17 controllers 270 can be in communication with the HMDs, either directly or via an
18 external device (e.g., core processing component 254). The controllers can have their
19 own IMU units, position sensors, and/or can emit further light points. The HMD 200
20 or 250, external sensors, or sensors in the controllers can track these controller light
21 points to determine the controller positions and/or orientations (e.g., to track the
22 controllers in 3DoF or 6DoF). The compute units 230 in the HMD 200 or the core
23 processing component 254 can use this tracking, in combination with IMU and position
24 output, to monitor hand positions and motions of the user. The controllers can also
25 include various buttons (e.g., buttons 272A-F) and/or joysticks (e.g., joysticks 274A-
26 B), which a user can actuate to provide input and interact with objects.

27 In various implementations, the HMD 200 or 250 can also include additional
28 subsystems, such as an eye tracking unit, an audio system, various network
29 components, etc., to monitor indications of user interactions and intentions. For
30 example, in some implementations, instead of or in addition to controllers, one or more
31 cameras included in the HMD 200 or 250, or from external cameras, can monitor the
32 positions and poses of the user's hands to determine gestures and other hand and
33 body motions. As another example, one or more light sources can illuminate either or
34 both of the user's eyes and the HMD 200 or 250 can use eye-facing cameras to

1 capture a reflection of this light to determine eye position (e.g., based on set of
2 reflections around the user's cornea), modeling the user's eye and determining a gaze
3 direction.

4 Figure 3 is a block diagram illustrating an overview of an environment 300 in
5 which some implementations of the disclosed technology can operate. Environment
6 300 can include one or more client computing devices 305A-D, examples of which can
7 include computing system 100. In some implementations, some of the client
8 computing devices (e.g., client computing device 305B) can be the HMD 200 or the
9 HMD system 250. Client computing devices 305 can operate in a networked
10 environment using logical connections through network 330 to one or more remote
11 computers, such as a server computing device.

12 In some implementations, server 310 can be an edge server which receives
13 client requests and coordinates fulfillment of those requests through other servers,
14 such as servers 320A-C. Server computing devices 310 and 320 can comprise
15 computing systems, such as computing system 100. Though each server computing
16 device 310 and 320 is displayed logically as a single server, server computing devices
17 can each be a distributed computing environment encompassing multiple computing
18 devices located at the same or at geographically disparate physical locations.

19 Client computing devices 305 and server computing devices 310 and 320 can
20 each act as a server or client to other server/client device(s). Server 310 can connect
21 to a database 315. Servers 320A-C can each connect to a corresponding database
22 325A-C. As discussed above, each server 310 or 320 can correspond to a group of
23 servers, and each of these servers can share a database or can have their own
24 database. Though databases 315 and 325 are displayed logically as single units,
25 databases 315 and 325 can each be a distributed computing environment
26 encompassing multiple computing devices, can be located within their corresponding
27 server, or can be located at the same or at geographically disparate physical locations.

28 Network 330 can be a local area network (LAN), a wide area network (WAN),
29 a mesh network, a hybrid network, or other wired or wireless networks. Network 330
30 may be the Internet or some other public or private network. Client computing devices
31 305 can be connected to network 330 through a network interface, such as by wired
32 or wireless communication. While the connections between server 310 and servers
33 320 are shown as separate connections, these connections can be any kind of local,
34 wide area, wired, or wireless network, including network 330 or a separate public or

1 private network.

2 Figure 4 is a block diagram illustrating components 400 which, in some
3 implementations, can be used in a system employing the disclosed technology.
4 Components 400 can be included in one device of computing system 100 or can be
5 distributed across multiple of the devices of computing system 100. The components
6 400 include hardware 410, mediator 420, and specialized components 430. As
7 discussed above, a system implementing the disclosed technology can use various
8 hardware including processing units 412, working memory 414, input and output
9 devices 416 (e.g., cameras, displays, IMU units, network connections, etc.), and
10 storage memory 418. In various implementations, storage memory 418 can be one or
11 more of: local devices, interfaces to remote storage devices, or combinations thereof.
12 For example, storage memory 418 can be one or more hard drives or flash drives
13 accessible through a system bus or can be a cloud storage provider (such as in
14 storage 315 or 325) or other network storage accessible via one or more
15 communications networks. In various implementations, components 400 can be
16 implemented in a client computing device such as client computing devices 305 or on
17 a server computing device, such as server computing device 310 or 320.

18 Mediator 420 can include components which mediate resources between
19 hardware 410 and specialized components 430. For example, mediator 420 can
20 include an operating system, services, drivers, a basic input output system (BIOS),
21 controller circuits, or other hardware or software systems.

22 Specialized components 430 can include software or hardware configured to
23 perform operations for controlling an indicator by combining signals from multiple
24 sensors. Specialized components 430 can include system shell 434, application state
25 manager 436, environment mode manager 438, and components and APIs which can
26 be used for providing user interfaces, transferring data, and controlling the specialized
27 components, such as interfaces 432. In some implementations, components 400 can
28 be in a computing system that is distributed across multiple computing devices or can
29 be an interface to a server-based application executing one or more of specialized
30 components 430. Although depicted as separate components, specialized
31 components 430 may be logical or other nonphysical differentiations of functions
32 and/or may be submodules or code-blocks of one or more applications.

33 System shell 434 can manage the software components of an XR system, such
34 as system display components, software functionality for system display components,

1 and interactions with launched applications. For example, system shell 434 can be a
2 shell environment in which applications are executed, such as shell applications or
3 applications remote from the shell. Example applications include web browsers,
4 music players, video players, social media applications, messaging web browsers,
5 music players, video players, social media applications, messaging or other
6 communication applications, third-party applications, streaming/casting applications,
7 a content library application, or any other suitable application.

8 Implementations of system shell 434 can interact with the applications
9 executing at the XR system to manage visual displays for the applications, such as
10 two-dimensional virtual object displays (e.g., panels). System shell 434 can manage
11 positioning and sizing of the virtual object displays and allocate portions of display
12 area/volume to the executing applications. Implementations of system shell 434 can
13 also restore a state for backgrounded/suspended application(s)/virtual object(s) using
14 the saved state(s) managed by state manager 436. In some implementations, system
15 shell 434 can comprise several units of executing software in combination. Additional
16 details on system shell 434 are provided below in relation to Figures 5, 6, 7, 8, and
17 blocks 902, 904, 906, 908, 910, and 912 of Figure 9.

18 Application state manager 436 can save a shell state for applications, such as
19 positioning and location for the virtual object display component of the applications.
20 For example, when an application is transitioned to background status, such as when
21 an exclusive mode is entered, or the system shell is transitioned to inactive (e.g., when
22 a user initiates an XR application that triggers display of a new XR environment), the
23 application state of executing application(s) and the shell state (e.g., position, size,
24 etc.) of their corresponding two-dimensional virtual object(s) can be stored. The state
25 storage responsibilities can be divided among application state manager 436 and the
26 individual applications.

27 For example, application state manager 436 can store the shell state for an
28 application, including: size and position (e.g., within the three-dimensional
29 environment) of the corresponding two-dimensional virtual object(s), application
30 specific display parameters (e.g., tab orientation, relevant websites/webpages, etc.),
31 and any other suitable state parameter(s) that initialize an application. State stored by
32 the individual applications can include: specific state of content displayed at the time
33 the application is closed (e.g., open tab, scroll position, open webpage, open
34 messages in a communication application, etc.), and any other suitable application

1 state parameter. When system shell 434 restores an application with a saved state,
2 the application's shell state stored by application state manager 436 is used to restore
3 the state of the application's virtual object (e.g., position and size of the virtual object,
4 initialization parameters for the application, etc.) and the application itself can restore
5 the content displayed by the restored virtual object according to the application's
6 stored state.

7 Environment mode manager 438 can manage a mode for the three-dimensional
8 environment/system shell 434. Example modes include cooperative mode and
9 exclusive mode. Implementations of cooperative mode permit concurrent execution
10 of multiple applications and display of multiple virtual objects, such as multiple panels
11 in a near-field relative to the user that display the contents of multiple executing
12 applications. In some examples, mode manager 438 can initiate exclusive mode for
13 an executing application, for example according to user input, based on a request from
14 the application, or in any other suitable manner. Implementations of the exclusive
15 mode permit display of virtual objects only from the executing application entering
16 exclusive mode. When exclusive mode is initiated, executing applications other than
17 the application entering exclusive mode are transitioned to the background and their
18 state(s) are stored. In addition, the virtual object displays for these applications are
19 no longer displayed until exclusive mode is exited. Additional details on environment
20 mode manager 438 are provided below in relation to blocks 906, 908, and 910 of
21 Figure 9.

22 Implementations of the system shell and the applications that execute at the
23 system shell can manage a lifecycle for applications. Example lifecycle stages for an
24 application running at the system shell can include: inactive, minimized, unfocused,
25 focused, and exclusive. An application at the exclusive stage can be operating in
26 exclusive mode at the system shell/in the three-dimensional environment of the shell.
27 An application in the focused stage can be one of several concurrently executing
28 applications operating in cooperative mode that has user input priority. For example,
29 a concurrently executing application can be focused when the user is interacting with
30 the application's virtual object display component or the last user interaction was with
31 the application's virtual object display component. In other examples, an application
32 can gain focus when it is launched, restored, or in any other suitable manner.

33 Application(s) in the unfocused stage can be one, two, or more concurrently
34 executing application(s) operating in cooperative mode that do not have user input

1 priority. For example, when three applications are concurrently executing in a system
2 shell, two can be unfocused (e.g., without user input priority) and one can be focused.
3 In some implementations, unfocused applications continue to execute at the system
4 shell (e.g., are not backgrounded or suspended) and their content continues to be
5 displayed at their corresponding virtual object displays.

6 Application(s) in the minimized stage can be backgrounded or suspended
7 applications that comprise a stored state (e.g., shell state and application state) so
8 that the application(s) can be restored (e.g., to a state at the time the application was
9 minimized). In some implementations, minimized applications are not executing at the
10 system shell and do not comprise a corresponding virtual object display. Applications
11 in an inactive state are applications that are not executing and do not comprise a
12 stored state. When launched, inactive applications create a new application session
13 (rather than restoring to a stored state).

14 In some implementations, during cooperative mode, when an application is
15 launched, restored from a stored state, or selected by a user (e.g., via user input) the
16 application is put into focus. For example, the application will gain focus (e.g., input
17 priority) and a previous application that had focus will lose focus. In some
18 implementations, a first application can launch a second application, and the launched
19 second application gains focus while the first application loses focus. In some
20 implementations, the system shell can automatically minimize one or more defocused
21 applications to conserve system resources. Because shell (e.g., virtual object position
22 and/or size) and application state are saved during minimization, the user does not
23 risk loss of state when the system shell minimizes a defocused application.

24 Figures 5 is a diagram of a three-dimensional environment for a system shell
25 with concurrently executing applications. Environment 500 includes virtual objects
26 502, 504, and 506, virtual hand 508, and menu 510. In some implementations,
27 environment 500 can be a three-dimensional environment for a system shell.
28 Environment 500 (and the system shell) can be operating in cooperative mode. For
29 example, virtual objects 502, 504, and 506 can be two-dimensional virtual objects
30 (e.g., panels) assigned to nodes at the near-field that each correspond to a different
31 executing application. Virtual hand 508 can be a user presence in environment 500
32 controlled by tracked user movement (e.g., hand movement, controller movement,
33 etc.). In environment 500, virtual hand 508 is interacting with virtual object 502.
34 Accordingly virtual object 502 and its corresponding executing application are in focus

(e.g., have user input priority) while virtual objects 504 and 506 and their corresponding executing applications are unfocused (e.g., executing with display components, lack of user input priority).

Menu 510 can comprise a listing of the focused application, unfocused applications, minimized applications, applications available for launch, recently used applications, any other suitable listing of applications, or any combination thereof. In some implementations, menu 510 can be used to launch applications, restore a minimized application to executing/unfocused status, enter exclusive mode for an application, or for any other suitable purpose.

Figures 6A and 6B are diagrams of three-dimensional environments for a system shell with multiple virtual object displays. Environment 600 of Figure 6A includes field 602, virtual objects 604, 606, and 608, exclusive field 610, and user 612. When environment 600 and the system shell are in cooperative mode, multiple applications can be concurrently running that each correspond to one or virtual objects 604, 606, and 608. Field 602 can be a cylindrical surface a distance away from user 612 (e.g., 2 meters, 4 meters, etc.) with node points assignable to virtual objects, such as a near-field. For example, virtual objects 604, 606, and 608 are each assigned to a node of field 602.

Environment 620 of Figure 6B includes virtual objects 622, 624, and 626. Environment 620 can be similar to environment 500 of Figure 5, where concurrently executing applications can each correspond to one of virtual objects 622, 624, and 626. For example, virtual objects 622, 624, and 626 can each display the content for an executing application.

Returning to environment 600 of Figures 6A, in some implementations environment 600 can transition to exclusive mode, for example in response to user input that transitions one of the executing applications into exclusive mode, when an application requests exclusive mode, or in any other suitable manner. Exclusive field 610 can be a field within environment 600 closer to user 612 than the field 602 in which a virtual object (e.g., two-dimensional panel) can be displayed, such as a virtual object that corresponds to the application running in exclusive mode. In this example, the display of virtual objects 604, 606, and 608 can be selectively removed from environment 600 such that only the virtual object that corresponds the application executing in exclusive mode remains. This virtual object can be displayed at any other suitable location other than exclusive field 610.

1 In some implementation's, an application operating in exclusive mode displays
2 a three-dimensional XR environment to user 612. For example, prior to exclusive
3 mode, the three-dimensional XR environment displayed to user 612 can be a shell XR
4 environment. In exclusive mode, the shell XR environment can be replaced by an
5 application XR environment (e.g., XR environment provided by the application
6 operating in exclusive mode). The content displayed throughout the XR environment
7 can include multiple virtual objects, a sky box, background, lighting, filters, augments,
8 and any other suitable content. In some implementations, the application operating in
9 exclusive mode can be an immersive gaming application, and the content can be
10 immersive gaming content (e.g., a user avatar, other user avatars, three-dimensional
11 virtual objects, two-dimensional panels, different skyboxes and backgrounds, etc.).

12 Implementations of the three-dimensional environment include a theater screen
13 at a far-field region or volume. Figure 7A is a diagram of a three-dimensional
14 environment with a theater display. Environment 700 includes theater object 702,
15 far-field 704, near-field 706, and user 708. Far-field 704 can be a cylindrical surface,
16 portion of a cylindrical surface, or any other suitable field that is further from the user
17 than near-field 706. Theater object 702 can be a virtual object assigned to far-field
18 704 that displays content for an executing application. In some implementations,
19 environment 700 can operate in cooperative mode. For example, theater object 702
20 can display a video while a user interacts with other executing applications via their
21 corresponding virtual objects.

22 Figure 7B is a diagram of a three-dimensional environment illustrating multiple
23 display locations. Diagram 710 illustrates different display locations for near-field
24 displays 712, a far-field display 716, and an exclusive field display 714. Each of
25 near-field displays 712, far-field display 716, and exclusive field display 714 can
26 display content from an executing application, such as in different modes of operation
27 for a system shell. Exclusive field display 714 can be a virtual object that displays
28 content for an application that executes in exclusive mode. Near-field displays 712
29 can be virtual objects that display content for concurrently executing applications while
30 the system shell operates in cooperative mode. In one or both of exclusive mode or
31 cooperative mode, far-field display 716 can be a theater object that displays
32 application content to a user, such as video content.

33 In some implementations, a theater screen can be shared among multiple
34 users. Figure 8 is a diagram of a shared three-dimensional environment in theater

1 mode. Environment 800 includes theater object 802 and users 804. Each of users
2 804 can be displayed a three-dimensional environment by XR systems that comprise
3 theater object 802. For example, a video can be cast to theater object 802 by any
4 participating XR system/system shell, and the video can be simultaneously displayed
5 to users 804 via their XR systems.

6 Those skilled in the art will appreciate that the components illustrated in Figures
7 1-5, 6A, 6B, 7A, 7B, and 8 described above, and in each of the flow diagrams
8 discussed below, may be altered in a variety of ways. For example, the order of the
9 logic may be rearranged, substeps may be performed in parallel, illustrated logic may
10 be omitted, other logic may be included, etc. In some implementations, one or more
11 of the components described above can execute one or more of the processes
12 described below.

13 Figure 9 is a flow diagram illustrating a process used in some implementations
14 of the present technology for application multitasking in a shell with a
15 three-dimensional environment. In some implementations, process 900 can be
16 performed by an XR system. In some implementations, process 900 can be triggered
17 when a three-dimensional environment for a system shell is displayed to a user.

18 At block 902, process 900 can display a three-dimensional environment for a
19 system shell to a user. For example, an XR system can display an immersive
20 three-dimensional environment to a user that includes content (e.g., virtual objects,
21 application content, etc.) for a system shell. In some implementations, the shell XR
22 environment can be displayed to a user via an XR device. In some implementations,
23 a given executing application and/or executing software process can comprise display
24 priority such that the XR device displays content from the priority application/process.
25 In some instances, a shell application can comprise this priority (e.g., in cooperative
26 mode). In other instances, another executing application can comprise this priority
27 (e.g., when the other application is executing in exclusive mode).

28 At block 904, process 900 can display first and second virtual objects in the
29 three-dimensional environment. For example, the system shell and three-dimensional
30 environment can operate in a cooperative mode where multiple concurrent
31 applications are executing. The first virtual object can display content for a first
32 executing application and the second virtual object can display content for a second
33 executing application. In some implementations, the first and second virtual objects
34 are displayed at a first distance from the user, such as at a near-field, while the system

1 shell and three-dimensional environment are in cooperative mode.

2 At block 906, process 900 can detect when an exclusive mode has been
3 initiated for the first application. For example, a user can initiate exclusive mode for
4 the first application, or the first application can request exclusive mode to perform
5 application functionality. In response, the system shell and three-dimensional
6 environment can transition from cooperative mode to exclusive mode. When exclusive
7 mode for an application is initiated, process 900 can progress to block 908. When
8 exclusive mode for an application is not initiated, process 900 can loop back to block
9 904, where the first and second virtual objects can continue to be displayed in the
10 three-dimensional environment until exclusive mode is initiated.

11 At block 908, process 900 can provide the first application exclusive mode
12 operation. For example, in exclusive mode, the first application can display any
13 suitable content to the user via the XR device display component (e.g., head-mounted
14 display). In some implementations, the operation in exclusive mode prevents
15 applications, other than the first application, from displaying world-locked content via
16 the XR device. In some implementations, the operation in exclusive mode provides
17 the first application use of at least a designated exclusive region of the XR environment
18 (e.g., the whole viewable area, a field in front of the user, a specified volume taking
19 most of the viewable area, etc).

20 In some implementations, process 900 can, in exclusive mode, dynamically
21 display a third virtual object in the XR environment (e.g., shell XR environment). For
22 example, the third virtual object can be displayed at a distance from the user that is
23 closer than the first distance, such as in the exclusive region. The third virtual object
24 can be displayed at any other suitable location. The third virtual object can display
25 content for the first application while the application remains in exclusive mode. In
26 some implementations, while in exclusive mode the first virtual object and second
27 virtual object are removed from display in the three-dimensional environment (e.g.,
28 shell XR environment).

29 In some implementations, while operating in exclusive mode, the first
30 application can display, via the XR device, a first application XR environment to the
31 user. For example, prior to exclusive mode (e.g., in cooperative mode), the three-
32 dimensional environment displayed to the user can be a shell XR environment. In
33 exclusive mode, content for the first application can be a first application XR
34 environment, and the XR device can replace the shell XR environment with the first

1 application XR environment. For example, the first application can comprise priority
2 over the shell application relative to the XR device display component when operating
3 in exclusive mode. In some implementations, the first application can comprise an
4 immersive gaming application, and the first application XR environment can comprise
5 immersive gaming content (e.g., avatar(s), virtual object(s), skybox(es),
6 background(s), etc.).

7 At block 910, process 900 can store, in response to triggering the exclusive
8 mode for the first application, state information for the second executing application
9 and the second virtual object. In some implementations, the second executing
10 application is transitioned to a background (e.g., suspended, minimized, etc.) while the
11 first application operates in exclusive mode. The system shell and second application
12 can save the second application's state at the time it is transitioned to the background.
13 For example, the stored state information can include a content state for the second
14 virtual object and an execution state for the second executing application.

15 At block 912, process 900 can detect when exclusive mode is terminated. For
16 example, a user can exit exclusive mode for the first application (e.g., via input that
17 transitions from exclusive mode) and/or the first application can request a transition
18 from exclusive mode (e.g., back to cooperative mode). When it is detected that
19 exclusive mode is terminated, process 900 can progress to block 912. When it is not
20 detected that exclusive mode has terminated, process 900 can loop back to block 908,
21 where the third virtual object can continue to be displayed in exclusive mode until
22 exclusive mode is terminated.

23 At block 914, process 900 can restore the first and second virtual objects. For
24 example, the third virtual object can be removed from display when exclusive mode is
25 terminated and the first and second virtual objects can be redisplayed (e.g., at the first
26 distance from the user). In some implementations, the first virtual object resumes
27 display of the content for the first application while executing in cooperative mode. In
28 some implementations, the second application is transitioned back to executing and
29 its state is restored based on its previously stored state. The second virtual object can
30 display content (e.g., the restored state) of the second application while executing in
31 cooperative mode.

32 In some implementations, in response to terminating the exclusive mode, the
33 XR device replaces the first application XR environment with the shell XR
34 environment. For example, when the content for the first application comprises a first

1 application XR environment, terminating exclusive mode can include returning to the
2 shell XR environment. In this example, the first and second virtual objects can be
3 restored in the shell XR environment.

4 In some implementations, redisplaying the second virtual object in the XR
5 environment comprises restoring, using the stored state information, content displayed
6 at the second virtual object and an execution state for the second application. For
7 example, the second application can be restored to the execution state that it was in
8 at the time that exclusive mode was triggered for the first application. In addition, the
9 content displayed at the second virtual object can be restored to the content it
10 displayed at the time that exclusive mode was triggered for the first application.
11 Process 900 can then progress to block 904, where the first and second virtual objects
12 are displayed in cooperative mode in the three-dimension environment for the system
13 shell.

14 Reference in this specification to "implementations" (e.g., "some
15 implementations," "various implementations," "one implementation," "an
16 implementation," etc.) means that a particular feature, structure, or characteristic
17 described in connection with the implementation is included in at least one
18 implementation of the disclosure. The appearances of these phrases in various places
19 in the specification are not necessarily all referring to the same implementation, nor
20 are separate or alternative implementations mutually exclusive of other
21 implementations. Moreover, various features are described which may be exhibited
22 by some implementations and not by others. Similarly, various requirements are
23 described which may be requirements for some implementations but not for other
24 implementations.

25 As used herein, being above a threshold means that a value for an item under
26 comparison is above a specified other value, that an item under comparison is among
27 a certain specified number of items with the largest value, or that an item under
28 comparison has a value within a specified top percentage value. As used herein,
29 being below a threshold means that a value for an item under comparison is below a
30 specified other value, that an item under comparison is among a certain specified
31 number of items with the smallest value, or that an item under comparison has a value
32 within a specified bottom percentage value. As used herein, being within a threshold
33 means that a value for an item under comparison is between two specified other
34 values, that an item under comparison is among a middle-specified number of items,

1 or that an item under comparison has a value within a middle-specified percentage
2 range. Relative terms, such as high or unimportant, when not otherwise defined, can
3 be understood as assigning a value and determining how that value compares to an
4 established threshold. For example, the phrase "selecting a fast connection" can be
5 understood to mean selecting a connection that has a value assigned corresponding
6 to its connection speed that is above a threshold.

7 As used herein, the word "or" refers to any possible permutation of a set of
8 items. For example, the phrase "A, B, or C" refers to at least one of A, B, C, or any
9 combination thereof, such as any of: A; B; C; A and B; A and C; B and C; A, B, and C;
10 or multiple of any item such as A and A; B, B, and C; A, A, B, C, and C; etc.

11 Although the subject matter has been described in language specific to
12 structural features and/or methodological acts, it is to be understood that the subject
13 matter defined in the appended claims is not necessarily limited to the specific features
14 or acts described above. Specific embodiments and implementations have been
15 described herein for purposes of illustration, but various modifications can be made
16 without deviating from the scope of the embodiments and implementations. The
17 specific features and acts described above are disclosed as example forms of
18 implementing the claims that follow. Accordingly, the embodiments and
19 implementations are not limited except as by the appended claims.

20

1 CLAIMS

- 2 1. A method for application multitasking in a shell with a three-dimensional artificial
3 reality, XR, environment, the method comprising:
4 displaying the XR environment comprising a first virtual object that
5 displays content for a first application executing in the shell, and a second
6 virtual object that displays content for a second application executing in the
7 shell;
8 triggering an exclusive mode for the first application, the exclusive mode
9 preventing applications, other than the first application, from displaying world-
10 locked content, wherein, during the exclusive mode, the first and second virtual
11 objects are removed from display, and the second application is transitioned to
12 a background state;
13 redisplaying, in response to termination of the exclusive mode, the first
14 virtual object and the second virtual object in the XR environment by:
15 transitioning the second application from the background state to
16 executing in the shell; and
17 displaying content for the first application executing in the shell at
18 the first virtual object and content for the second application executing in
19 the shell at the second virtual object.
- 20 2. The method of claim 1, wherein when displayed or redisplayed, the first virtual
21 object and the second virtual object are displayed side-by-side in a predefined
22 near-field region of the XR environment relative to the user.
- 23 3. The method of claim 2, wherein the near-field region of the XR environment
24 comprises a plurality of nodes positioned side-by-side and, when displayed,
25 each of the first virtual object and second virtual object is assigned to one of the
26 nodes.
- 27 4. The method of claim 2 or claim 3, wherein, in exclusive mode, content for the
28 first application is dynamically moved to a third virtual object displayed in the
29 exclusive region of the XR environment.
- 30 5. The method of claim 4, wherein the exclusive region is closer in proximity to the
31 user than the near-field region in the XR environment.
- 32 6. The method of claim 2, wherein the triggered exclusive mode for the first
33 application provides the first application use of at least a designated exclusive
34 region of the XR environment, and, in exclusive mode, content for the first

- 1 application is dynamically moved to a third virtual object displayed in the
2 exclusive region of the XR environment.
- 3 7. The method of claim 6, wherein the designated exclusive region of the XR
4 environment is closer in proximity to the user than the near-field region in the
5 XR environment.
- 6 8. The method of any of claims 4 to 7, wherein the first virtual object, second virtual
7 object, and third virtual object comprise two-dimensional panels.
- 8 9. The method of any preceding claim,
9 wherein the XR environment displayed to the user comprises a shell XR
10 environment,
11 wherein, in exclusive mode, the content for the first application
12 comprises a first application XR environment,
13 wherein, in exclusive mode, the shell XR environment is replaced with
14 the first application XR environment, and
15 wherein, in response to terminating the exclusive mode, the first
16 application XR environment is replaced with the shell XR environment.
- 17 10. The method of any preceding claim, further comprising:
18 storing, in response to triggering the exclusive mode for the first
19 application, state information for the second executing application and the
20 second virtual object, wherein the redisplaying the second virtual object in the
21 XR environment comprises restoring the second virtual object using the stored
22 state information.
- 23 11. The method of claim 8 or 9, wherein the stored state information comprises a
24 content state for the second virtual object and an execution state for the second
25 executing application.
- 26 12. The method of claim 9, wherein the redisplaying the second virtual object in the
27 XR environment comprises restoring, using the stored state information,
28 content displayed at the second virtual object and an execution state for the
29 second application.
- 30 13. The method of any preceding claim, wherein:
31 said displaying the XR environment comprises displaying the XR
32 environment to a user via an XR device;
33 said triggering the exclusive mode is in response to a request; and
34 the exclusive mode provides the first application use of at least a

- 1 designated exclusive region of the XR environment.
- 2 14. A computer-readable storage medium storing instructions that, when executed
3 by a computing system, cause the computing system to perform the method of
4 any preceding claim.
- 5 15. A computing system for application multitasking in a shell with a
6 three-dimensional artificial reality, XR, environment, the computing system
7 comprising:
8 one or more processors; and
9 one or more memories storing instructions that, when executed by the
10 one or more processors, cause the computing system to perform the method
11 of any of claims 1 to 13.

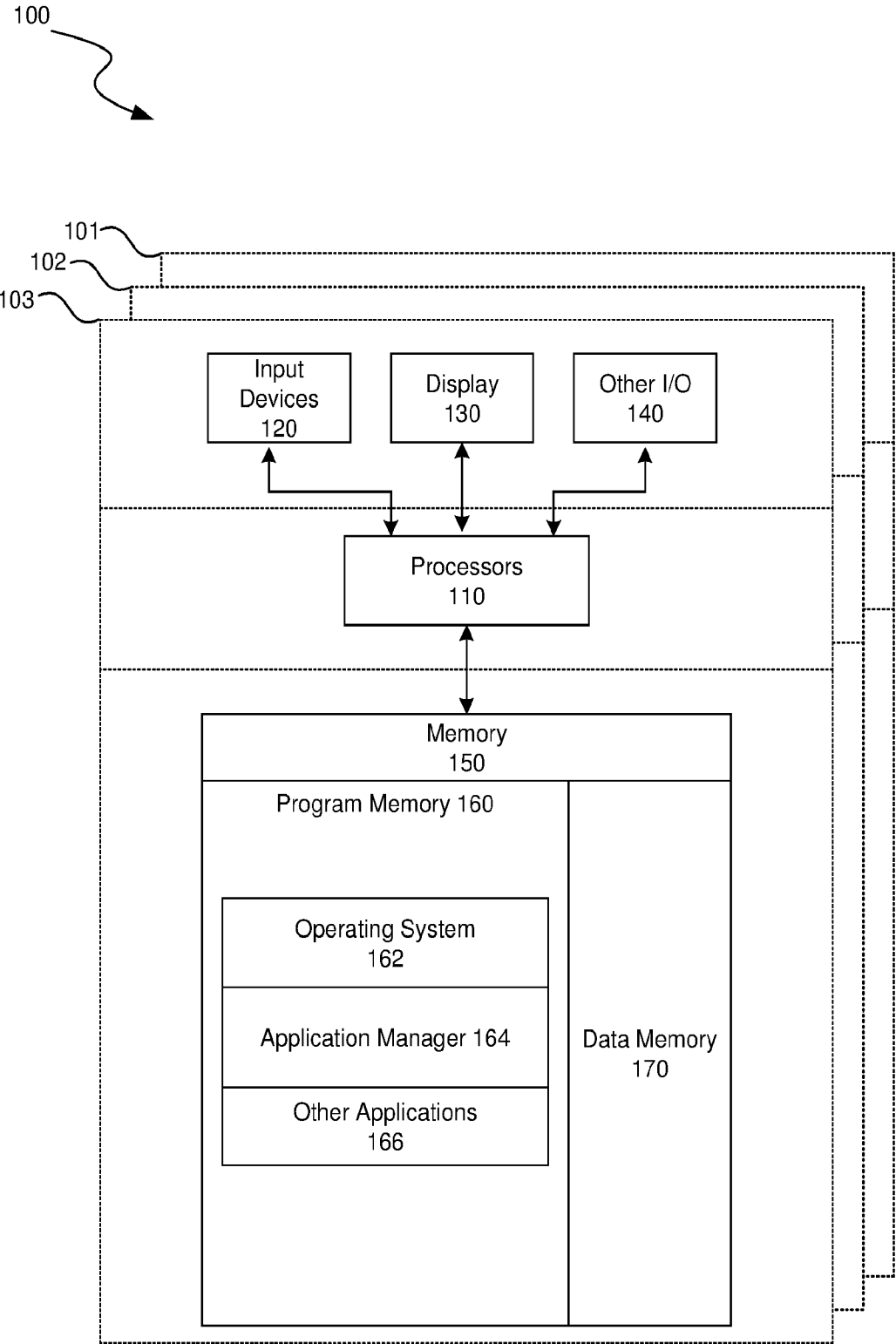


FIG. 1

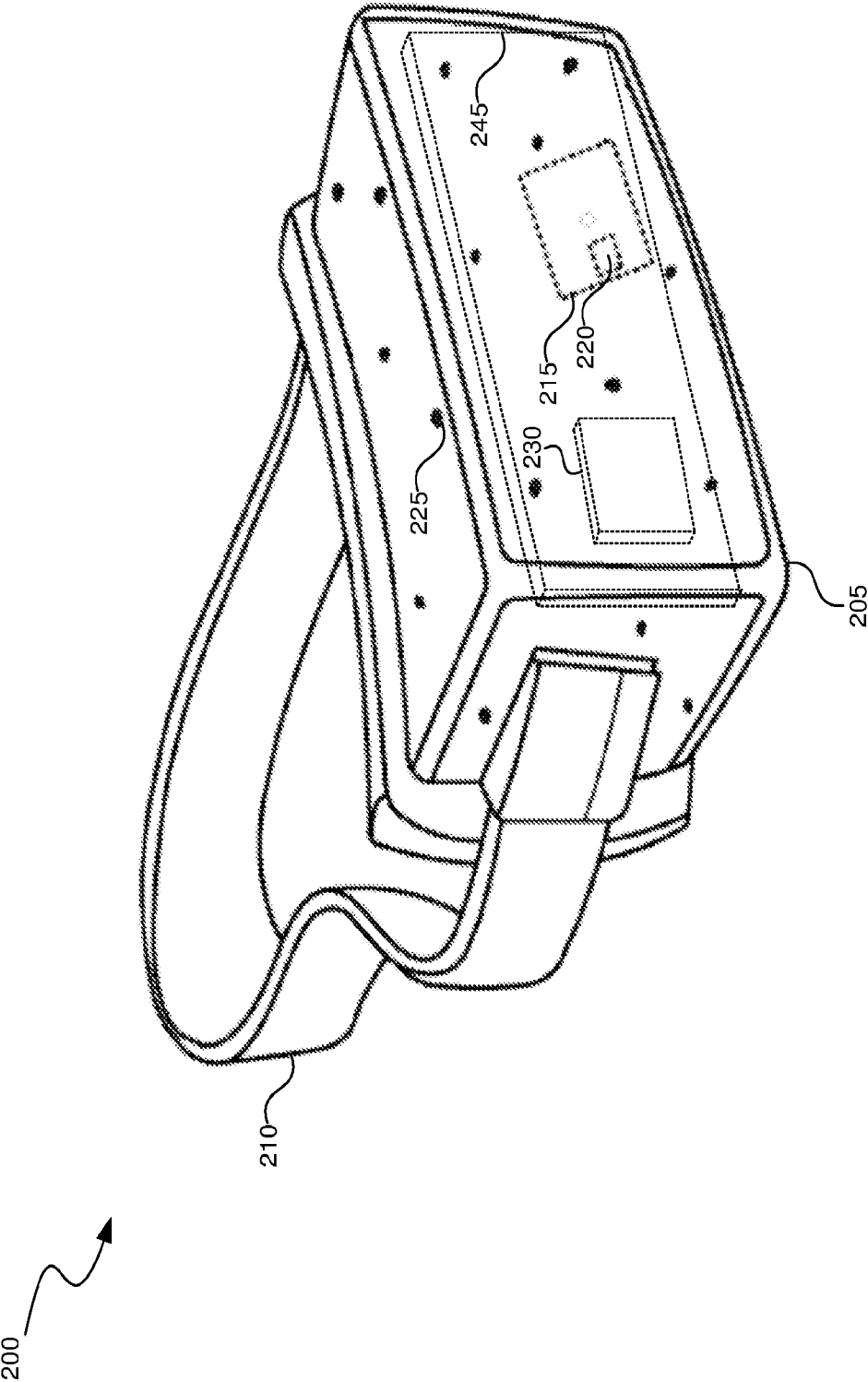


FIG. 2A

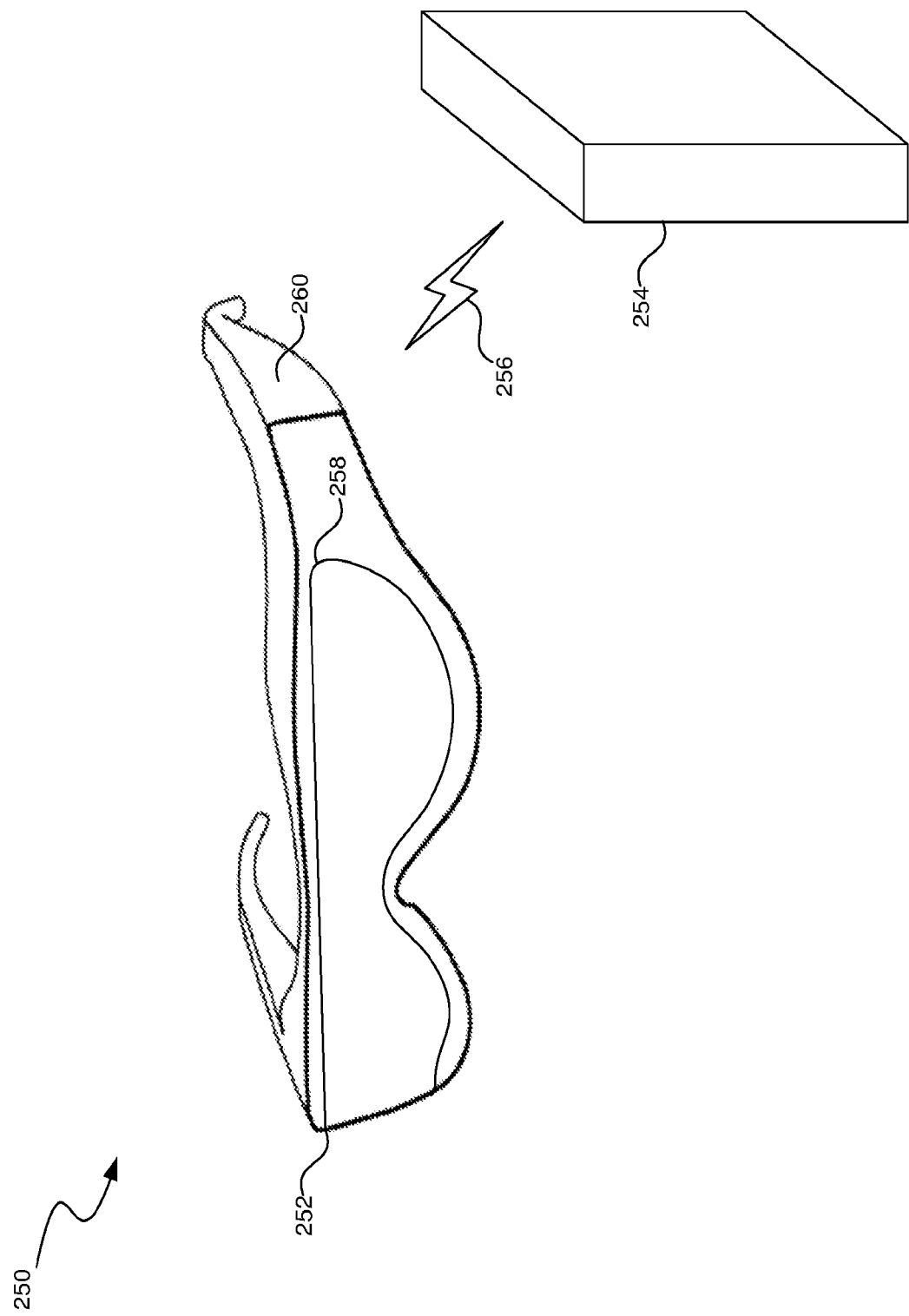


FIG. 2B

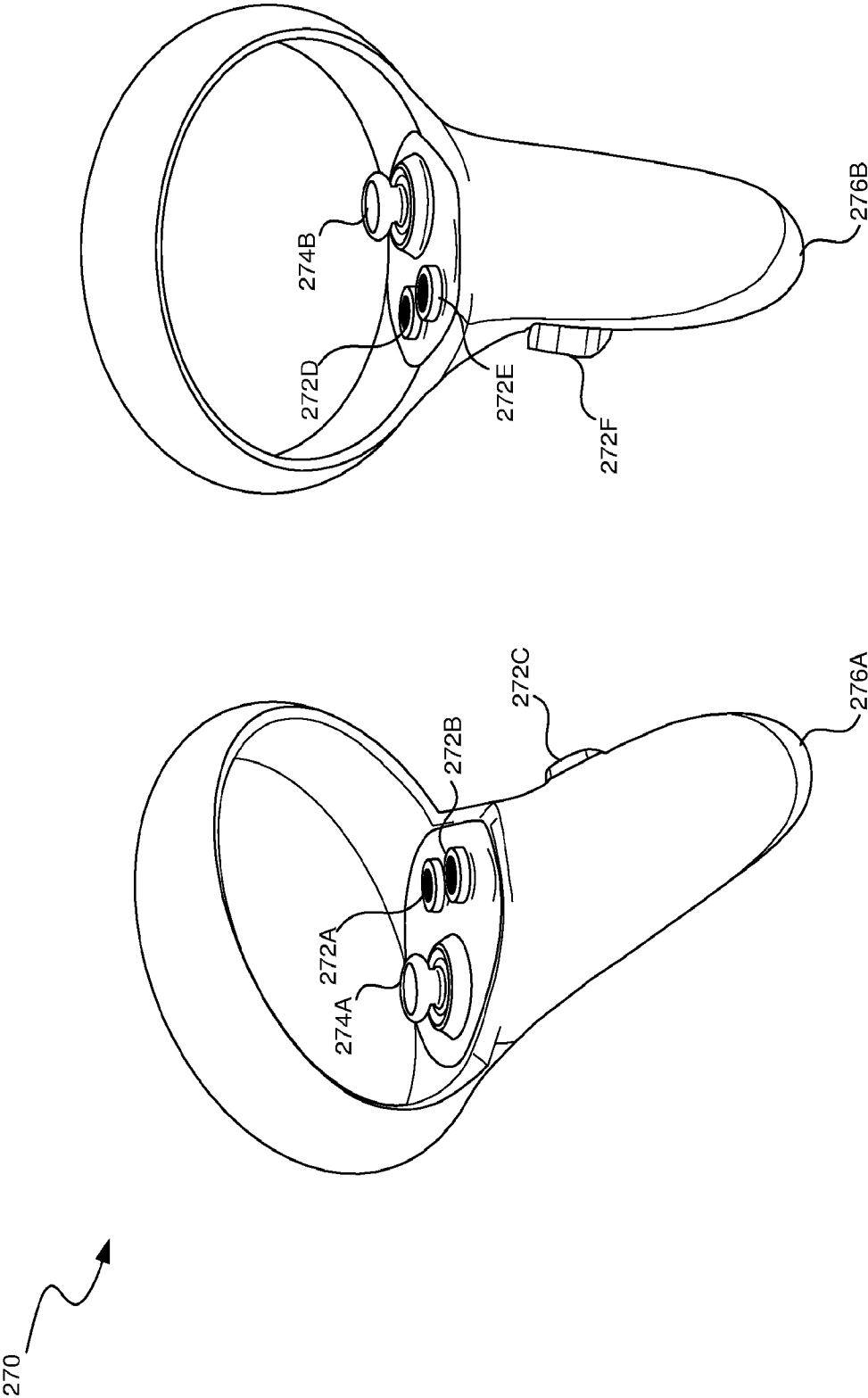


FIG. 2C

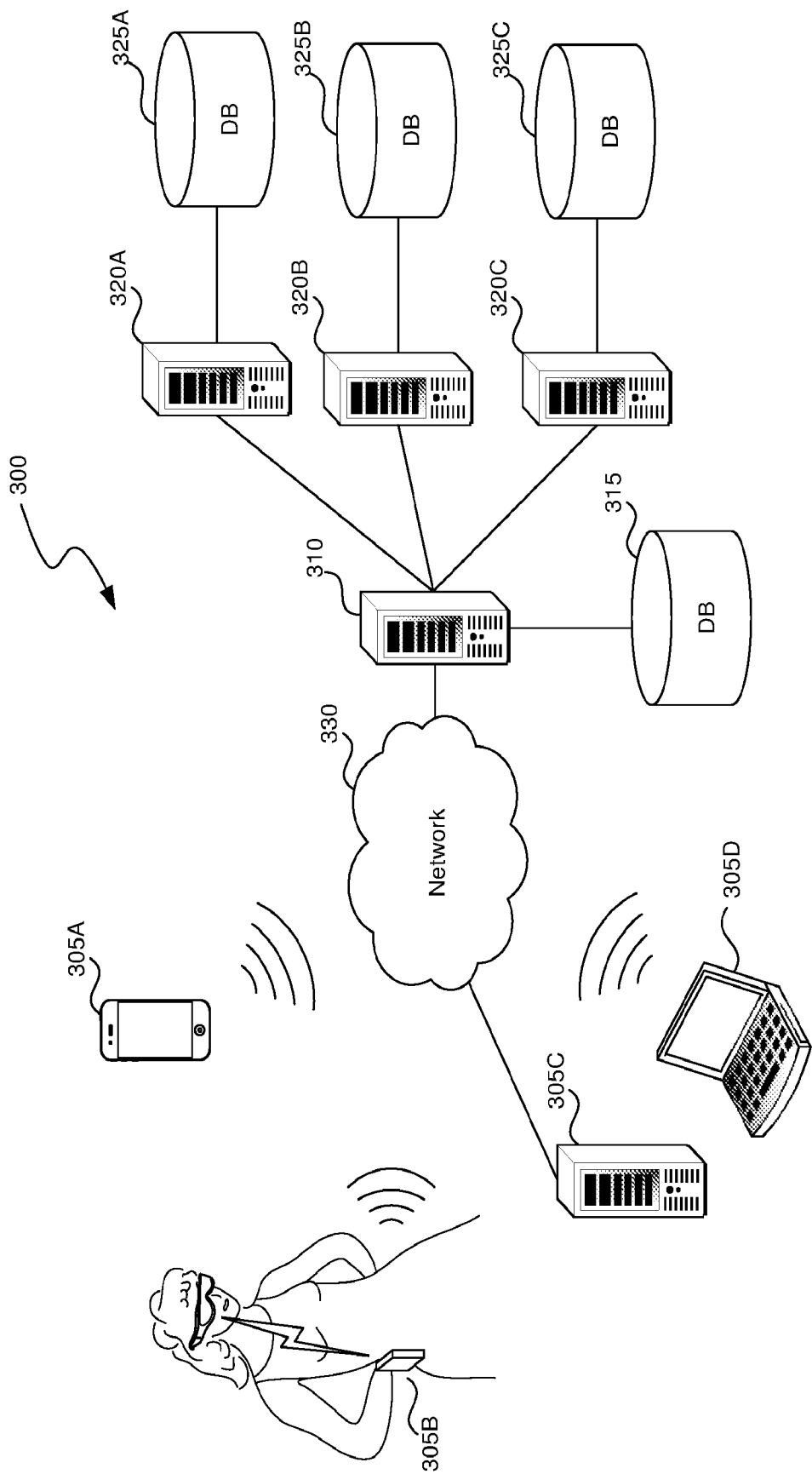


FIG. 3

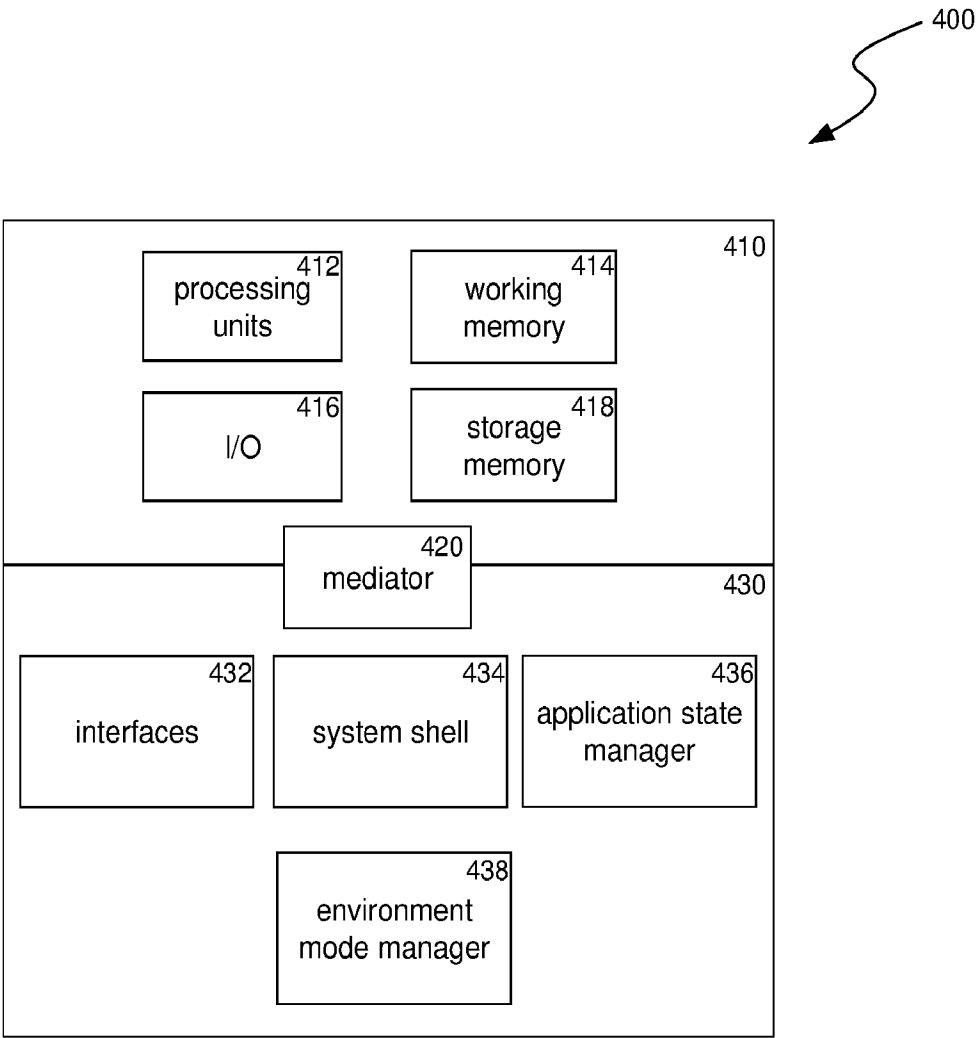


FIG. 4

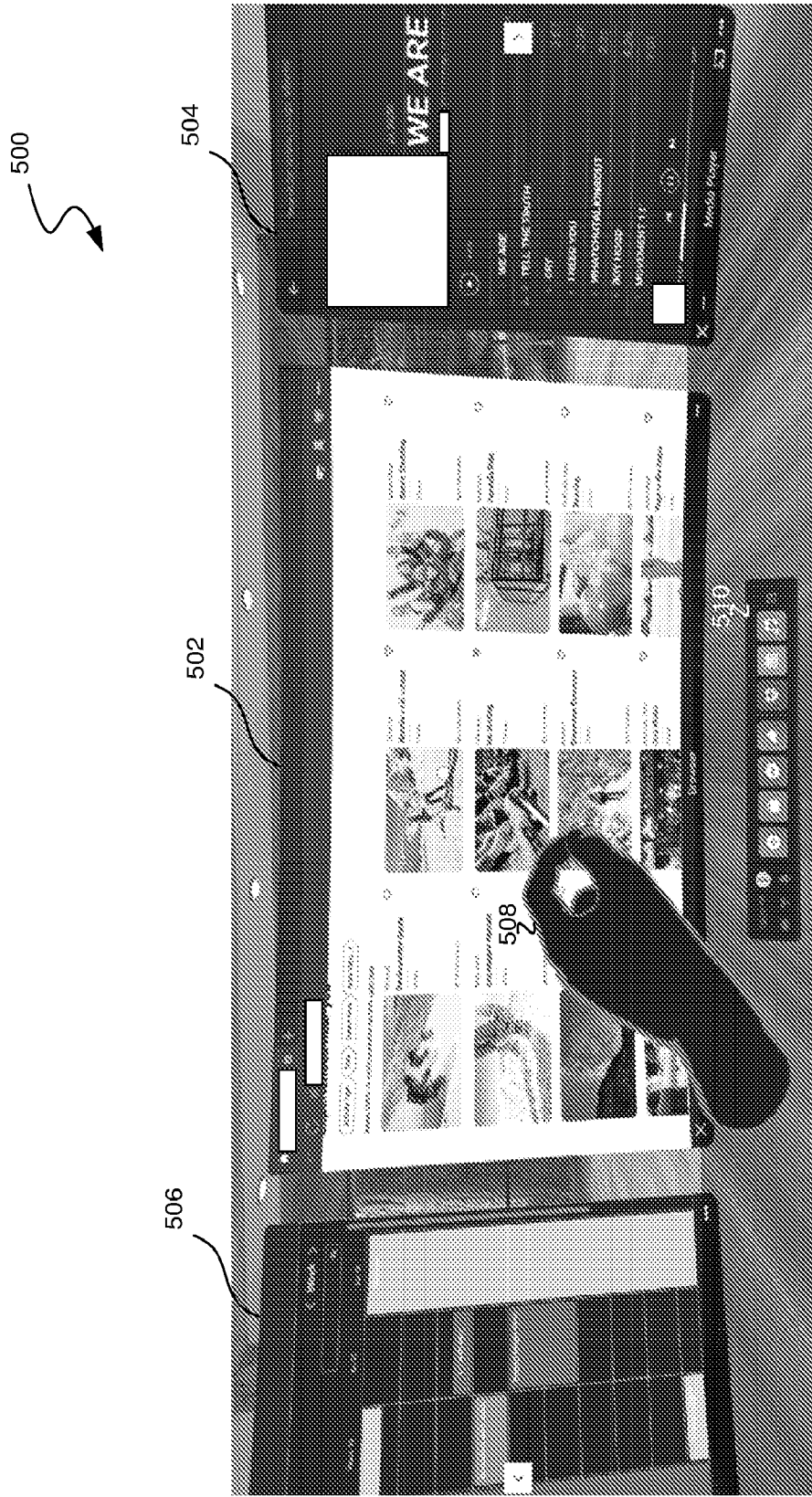


FIG. 5

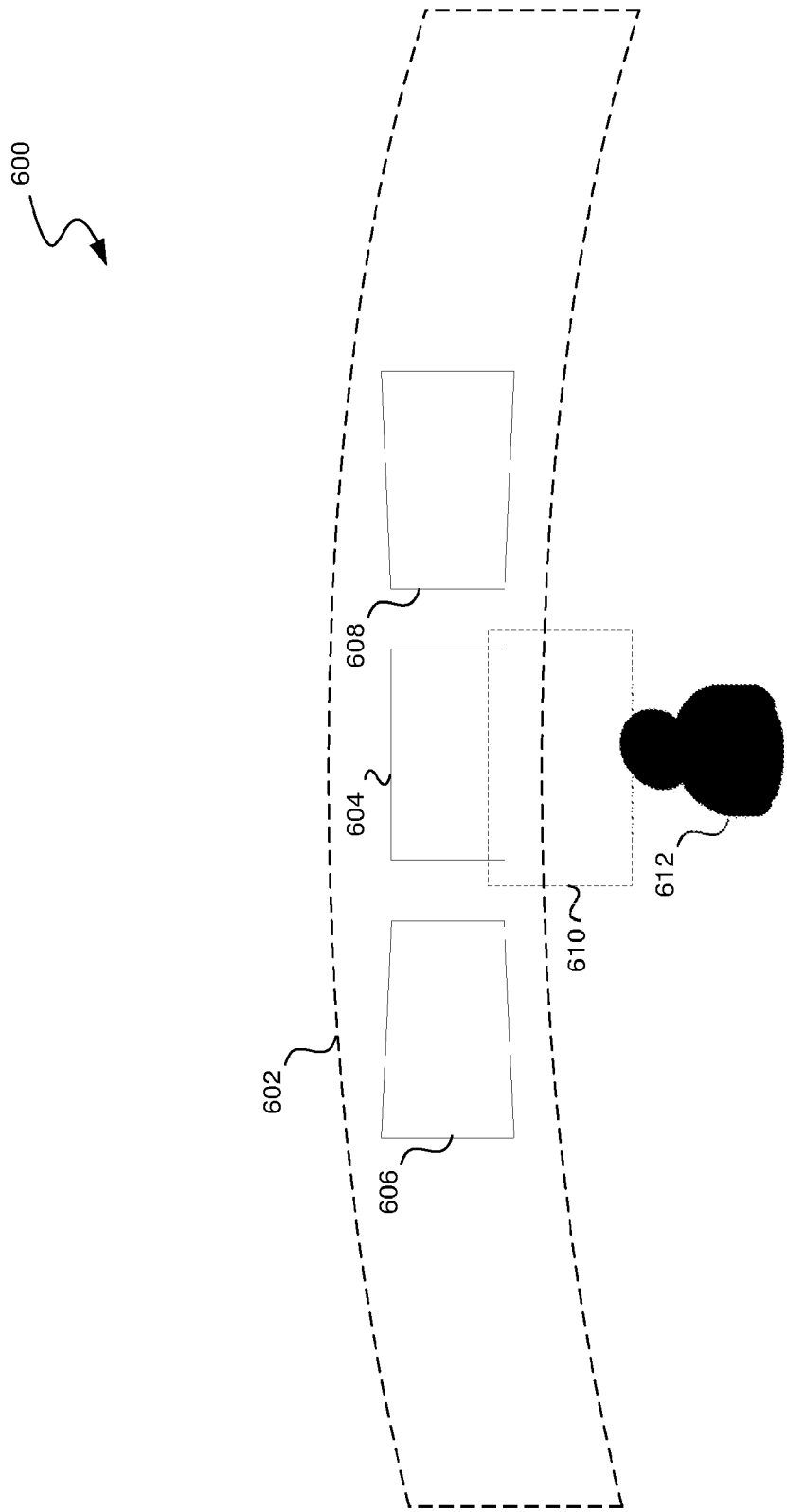


FIG. 6A

620

626

624

622

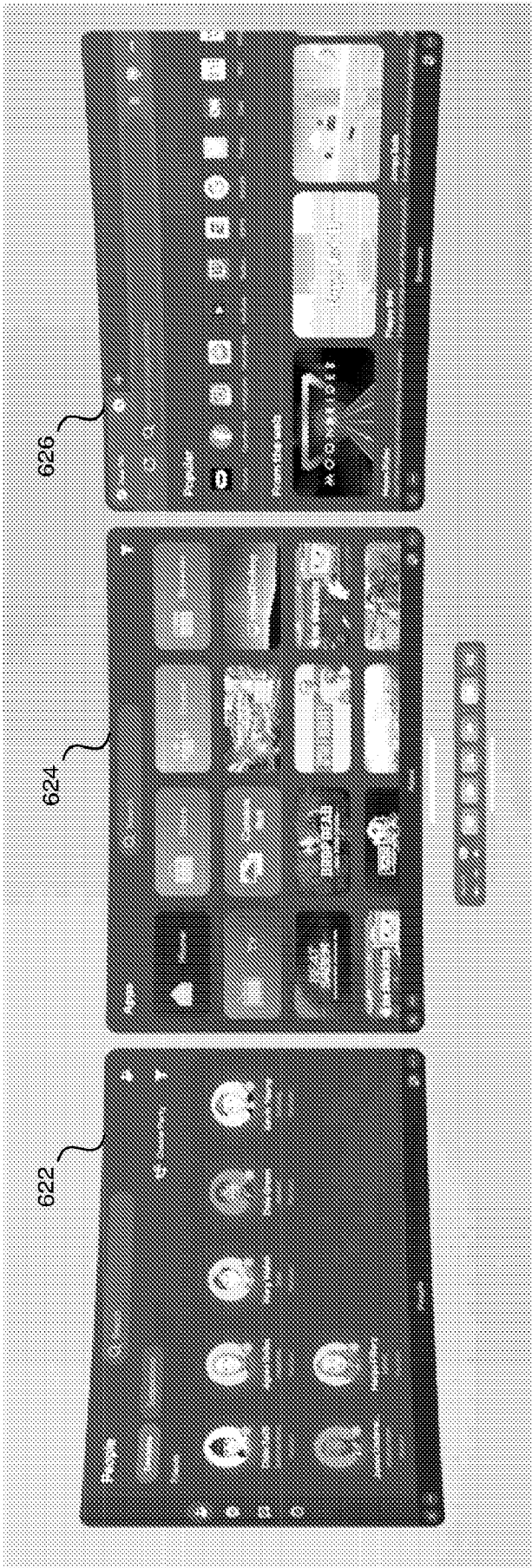


FIG. 6B

700

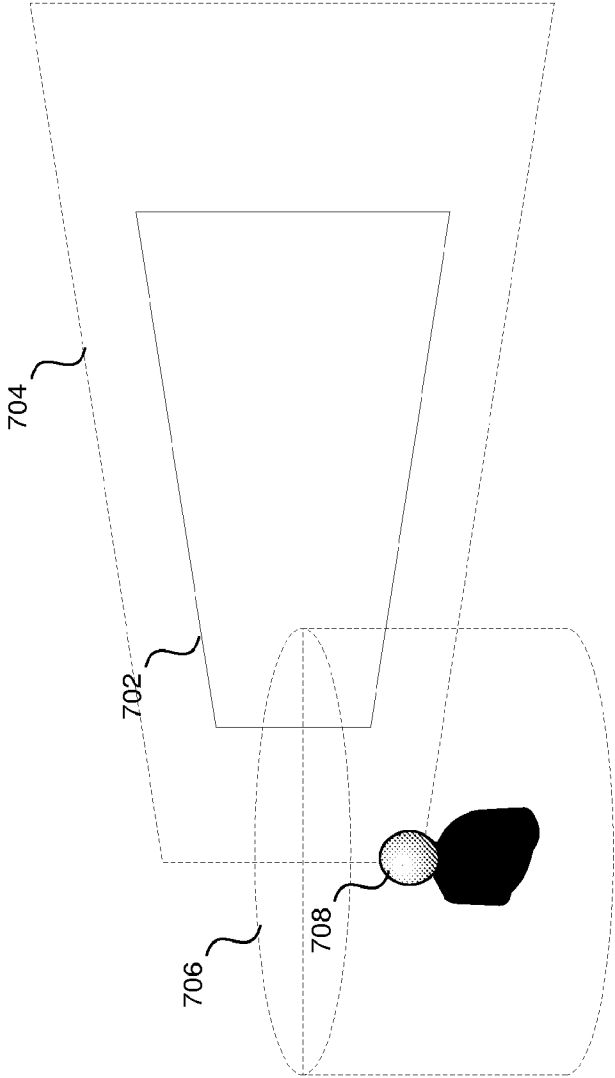


FIG. 7A

710

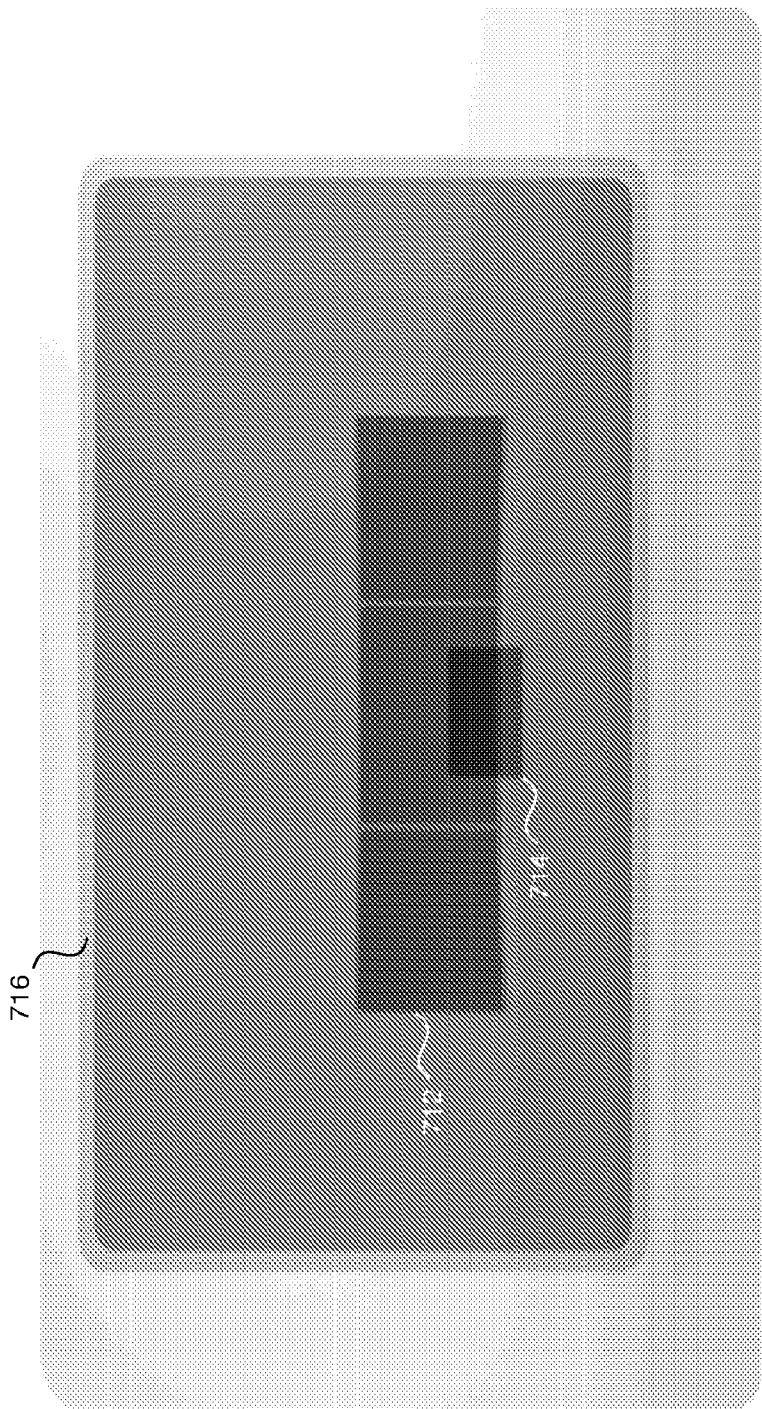


FIG. 7B

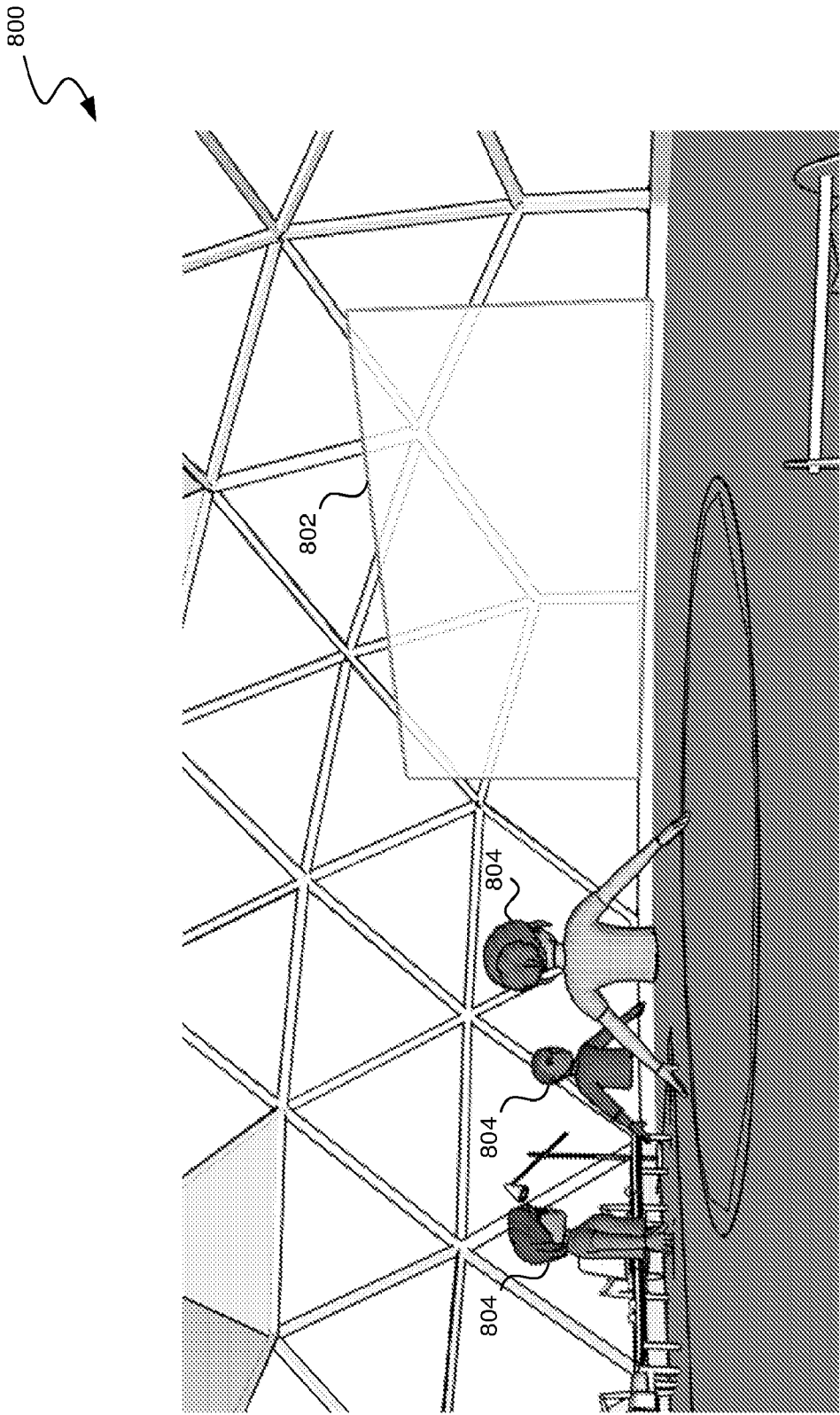
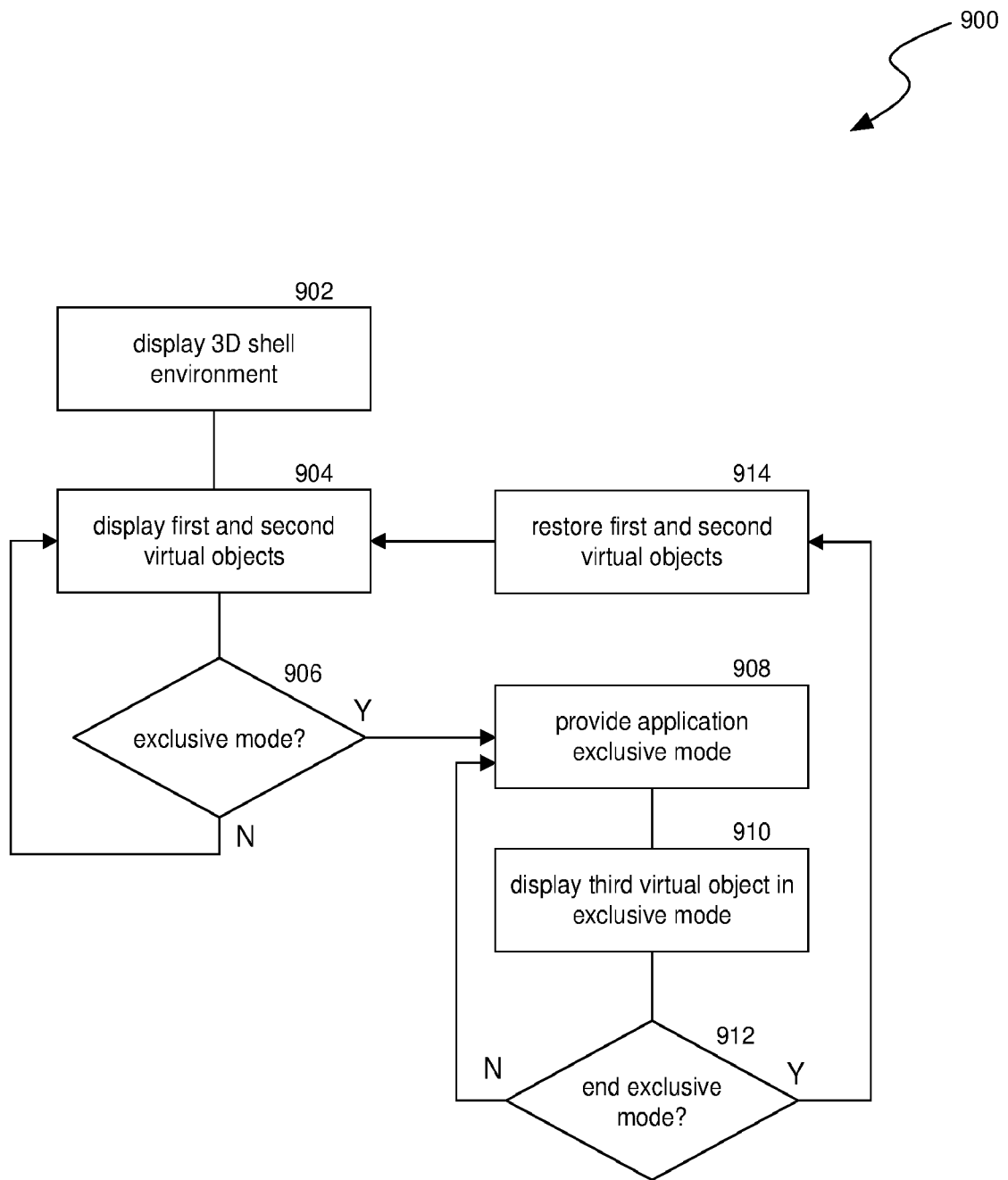


FIG. 8

**FIG. 9**

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2023/035591

A. CLASSIFICATION OF SUBJECT MATTER INV. G06F3/01 G06F9/451 G06F9/48 G06F9/54 G09G5/14 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 G09G 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		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A,O	Tyriel Wood - Vr Tech: "MULTITASKING IN VR w/ Oculus Core 2.0 Oculus Dash FIRST IMPRESSION", / 8 December 2017 (2017-12-08), XP093126606, Retrieved from the Internet: URL:https://www.youtube.com/watch?v=s8EwXI uCqEY [retrieved on 2024-02-01] * the whole video *	1-15
A	US 2011/144970 A1 (JIANG YINAN [CA] ET AL) 16 June 2011 (2011-06-16) paragraph [0045] the whole document	1-15
A,P	WO 2023/102139 A1 (APPLE INC [US]) 8 June 2023 (2023-06-08) the whole document	1-15
☐ Further documents are listed in the continuation of Box C. ☒ See patent family annex.		
* Special categories of cited documents : "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier application or patent but published on or after the international filing date "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) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed "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 "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 "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 "&" document member of the same patent family		
Date of the actual completion of the international search 1 February 2024		Date of mailing of the international search report 19/02/2024
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 del Rey, Marco

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/US2023/035591

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2011144970 A1	16-06-2011	CN 102782747 A	14-11-2012
		EP 2513895 A1	24-10-2012
		JP 5506120 B2	28-05-2014
		JP 2013513873 A	22-04-2013
		KR 20120099776 A	11-09-2012
		US 2011144970 A1	16-06-2011
		WO 2011072379 A1	23-06-2011

WO 2023102139 A1	08-06-2023	US 2023221830 A1	13-07-2023
		WO 2023102139 A1	08-06-2023
