(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
11 February 2021 (11.02.2021)

WIPO | PCT

(10) International Publication Number
WO 2021/026422 A1

(51) International Patent Classification:
G09G 5/00 (2006.01)

(21) International Application Number:
PCT/US2020/045344

(22) International Filing Date:
07 August 2020 (07.08.2020)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
62/884,020 07 August 2019 (07.08.2019) US
16/549,609 23 August 2019 (23.08.2019) US

(71) Applicant: **HUMAN MODE, LLC** [US/US]; 4005 Northwest Expy, Suite 610, Oklahoma City, OK 73116 (US).

(72) Inventors: **KERBER, William Xavier III**; 3101 Wheafall Ln., Edmond, OK 73034 (US). **SO, Yoon Joon**; 4608 NW Expressway, Apt D., Oklahoma City, OK 73132 (US). **CHOI, Jae Rim**; 8233 N. Rockwell Avenue, Oklahoma City, OK 73132 (US).

(74) Agent: **BROCKHAUS, Marc A.**; Dunlap Codding, P.C., P.O. Box 16370, Oklahoma City, OK 73113 (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, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, IT, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, 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.

(54) Title: AUTHORING SYSTEM FOR INTERACTIVE VIRTUAL REALITY ENVIRONMENTS

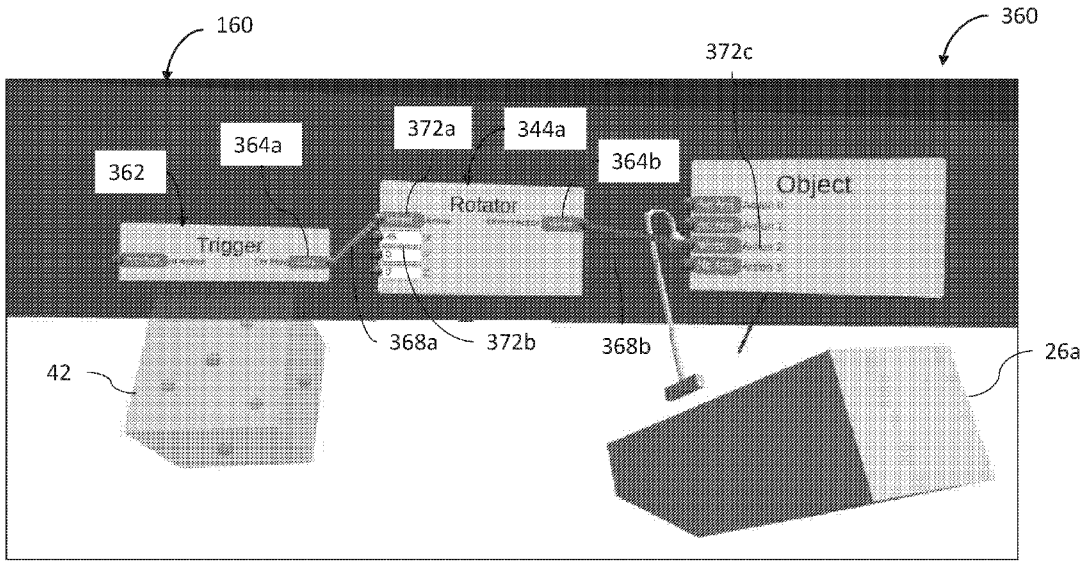


FIG. 12

(57) Abstract: A non-transitory computer readable medium storing computer executable code that when executed by a processor cause the processor to: project a mixed reality environment to a headset, the mixed reality environment having objects, each of the objects being created by object computer code having properties that define how the objects interact within the mixed reality environment; in response to a first input from at least one user interface device, display an action menu on the visual feedback screen having a list of pre-defined action modules, the action modules including a module of computer executable code having one or more functions, the one or more functions configured to affect one or more properties of a first object of the objects within the mixed reality environment; and update the object computer code of the first object to include a selected action module.

WO 2021/026422 A1

[Continued on next page]

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, 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, 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).

Published:

— *with international search report (Art. 21(3))*

AUTHORING SYSTEM FOR INTERACTIVE VIRTUAL REALITY ENVIRONMENTS

Incorporation By Reference

[001] The present patent application claims priority to U.S. Serial No. 16/549,609 filed on August 23, 2019; which claims priority to the provisional patent application identified by U.S. Serial No. 62/884,020, filed on August 7, 2019, the entire content of both patent applications being hereby incorporated herein by reference.

Background

[002] As virtual reality hardware advances, and becomes more capable, and the cost of virtual reality hardware decreases, there in an increased interest in Virtual Reality and its usefulness in various industries, however, many persons outside of the computer programming field do not have the required skills to program a virtual reality environment.

[003] Virtual Reality (VR) is a computer-generated simulation of a three-dimensional environment, including a physics engine, that can be interacted with in a seemingly real or physical way by a person using a Virtual Reality System. The Virtual Reality System includes a screen where multi-projected environments generate images, sounds, and other sensations that simulate a user's physical presence in the virtual environment and may include an interface device where a user can provide input to the virtual environment. Virtual reality systems enable a user to look around the artificial world, move around in it, and interact with virtual objects. This effect is commonly created by VR headsets consisting of a head-mounted display with a small screen in front of the eyes, but can also be created through specially designed rooms with multiple large screens. Virtual Reality typically incorporates auditory and video feedback, but may also include other environmental stimulation such as sensory and force feedback through haptic technology.

[004] Virtual reality environments, also referred to as virtual environments or VR applications, can include, for example, entertainment such as games and multimedia or training such as classroom education or virtual training. Programming virtual reality environments is a complicated and time intensive process further complicated by rendering a complex three-dimensional environment onto two-dimensional computer monitors. Additionally, advanced three-dimensional modeling, texturing, and lighting are skilled and time-consuming processes.

[005] Today, building virtual reality environments generally requires teams of individuals with a broad and advanced skillset including at least graphic art design, computer programming, understanding of three-dimensional math and physics, and audio engineering. Adding an individual with any one of these skills is costly due to demand and need. Moreover, due to the broad and advanced skillset needed to build virtual reality environments, the adoption of virtual reality is limited to those who have these skills or area able to build a team with these skills.

[006] What is needed is a system and method that provides users the ability to create and modify a virtual reality environment that is both simple and within the virtual reality environment being edited. It is to such a system to provide users the ability to create and modify a virtual reality environment that is both simple and within the virtual reality environment being edited that the present disclosure is directed.

Summary

[007] The problem of creating and modifying a virtual reality environment that is both simple and within the virtual reality environment being edited is solved with the systems described herein.

[008] In one embodiment, the present disclosure describes an electronic device comprising a processor, a headset, one or more user interface devices, and a non-transitory computer readable medium. The headset has a visual feedback screen. The non-transitory computer readable medium storing computer executable code that when executed by the processor cause the processor to project a mixed reality environment to the headset. The mixed reality environment has one or more objects. Each of the one or more objects is created by object computer code having properties that define how the one or more objects interact within the mixed reality environment. In response to a first input from at least one user interface device, an action menu is displayed on the visual feedback screen having a list of pre-defined action modules. The action modules include a module of computer executable code having one or more functions, the one or more functions configured to affect one or more properties of a first object of the one or more objects within the mixed reality environment. The object computer code of the first object is updated to include a selected action module.

Brief Description of Several Views of the Drawings

[009] The accompanying drawings, which are incorporated in and constitute a part of this specification, illustrate one or more implementations described herein and, together with the description, explain these implementations. The drawings are not intended to be drawn to

scale, and certain features and certain views of the figures may be shown exaggerated, to scale or in schematic in the interest of clarity and conciseness. Not every component may be labeled in every drawing. Like reference numerals in the figures may represent and refer to the same or similar element or function. In the drawings:

[010] FIG. 1 is an exemplary embodiment of a virtual reality world authoring system.

[011] FIG. 2 is a block diagram of an exemplary embodiment of a car object.

[012] FIG. 3 is a flow diagram of an exemplary embodiment of an object configuration process.

[013] FIG. 4 is a block diagram of an exemplary embodiment of a wheel object.

[014] FIG. 5 is a top plan view of a complex object formed of the car object of Figure 2 and the wheel objects of Figure 4.

[015] FIG. 6 is a block diagram of an exemplary embodiment of an object relationship model forming the complex object of Figure 5.

[016] FIG. 7 is an exemplary embodiment of a hierarchical object relationship model forming the complex object of Figure 5.

[017] FIG. 8 is a screen shot of an exemplary embodiment of a virtual environment showing an exemplary room settings menu.

[018] FIG. 9 is a screen shot of an exemplary embodiment of a virtual environment showing an exemplary tools selection menu.

[019] FIG. 10 is an exemplary embodiment of a network diagram for multi-user access to a virtual environment.

[020] FIG. 11 is a screen shot of an exemplary embodiment of a virtual environment showing an exemplary scriptables menu.

[021] FIG. 12 is a diagrammatic view of an exemplary embodiment of script builder.

[022] FIG. 13 is a diagrammatic view of an exemplary embodiment of a virtual environment.

[023] FIG. 14 is an exemplary embodiment of a computer system implementing the virtual reality world authoring system in accordance with the present disclosure.

Detailed Description

[024] Before explaining at least one embodiment of the disclosure in detail, it is to be understood that the disclosure is not limited in its application to the details of construction, experiments, exemplary data, and/or the arrangement of the components set forth in the following description or illustrated in the drawings unless otherwise noted.

[025] The disclosure is capable of other embodiments or of being practiced or carried out in various ways. For instance, although extent change of a residential building structure may be used as an example, the methods and systems may be used to assess other characteristics (by way of example and not limited to, changes in structure footprint or structure area) of other man-made objects, non-exclusive examples of which include other types of buildings such as industrial buildings, or commercial buildings and include infrastructure such as roads, bridges, utility lines, pipelines, utility towers. Also, it is to be understood that the phraseology and terminology employed herein is for purposes of description, and should not be regarded as limiting.

[026] As used in the description herein, the terms "comprises," "comprising," "includes," "including," "has," "having," or any other variations thereof, are intended to cover a non-exclusive inclusion. For example, unless otherwise noted, a process, method, article, or apparatus that comprises a list of elements is not necessarily limited to only those elements, but may also include other elements not expressly listed or inherent to such process, method, article, or apparatus.

[027] Further, unless expressly stated to the contrary, "or" refers to an inclusive and not to an exclusive "or". For example, a condition A or B is satisfied by one of the following: A is true (or present) and B is false (or not present), A is false (or not present) and B is true (or present), and both A and B are true (or present).

[028] In addition, use of the "a" or "an" are employed to describe elements and components of the embodiments herein. This is done merely for convenience and to give a general sense of the inventive concept. This description should be read to include one or more, and the singular also includes the plural unless it is obvious that it is meant otherwise. Further, use of the term "plurality" is meant to convey "more than one" unless expressly stated to the contrary.

[029] As used herein, qualifiers like "substantially," "about," "approximately," and combinations and variations thereof, are intended to include not only the exact amount or value that they qualify, but also some slight deviations therefrom, which may be due to computing tolerances, computing error, manufacturing tolerances, measurement error, wear and tear, stresses exerted on various parts, and combinations thereof, for example.

[030] As used herein, any reference to "one embodiment," "an embodiment," "some embodiments," "one example," "for example," or "an example" means that a particular

element, feature, structure or characteristic described in connection with the embodiment is included in at least one embodiment and may be used in conjunction with other embodiments. The appearance of the phrase “in some embodiments” or “one example” in various places in the specification is not necessarily all referring to the same embodiment, for example.

[031] The use of ordinal number terminology (i.e., "first", "second", "third", "fourth", etc.) is solely for the purpose of differentiating between two or more items and, unless explicitly stated otherwise, is not meant to imply any sequence or order or importance to one item over another or any order of addition.

[032] The use of the term "at least one" or “one or more” will be understood to include one as well as any quantity more than one. In addition, the use of the phrase "at least one of X, V, and Z" will be understood to include X alone, V alone, and Z alone, as well as any combination of X, V, and Z.

[033] Circuitry, as used herein, may be analog and/or digital components, or one or more suitably programmed processors (e.g., microprocessors) and associated hardware and software, or hardwired logic. Also, “components” may perform one or more functions. The term “component,” may include hardware, such as a processor (e.g., microprocessor), an application specific integrated circuit (ASIC), field programmable gate array (FPGA), a combination of hardware and software, and/or the like. The term “processor” as used herein means a single processor or multiple processors working independently or together to collectively perform a task.

[034] A programming language is a set of commands, instructions, and other syntax used to create software. Languages that programmers use to write code are called "high-level languages." High-level language code, which is designed to be read and understood by programmers, is compiled into a "low-level language," which is recognized directly by the computer hardware as computer readable instructions. This allows programmers to write source code in a more natural fashion, using logical words and symbols. Examples of high-level languages include C++, Java, Perl, and PHP. Low-level languages include assembly and machine languages.

[035] Software may include one or more computer readable instructions that when executed by one or more components cause the component to perform a specified function. It should be understood that the algorithms described herein may be stored on one or more non-transitory computer readable medium. Exemplary non-transitory computer readable mediums

may include random access memory, read only memory, flash memory, and/or the like. Such non-transitory computer readable mediums may be electrically based, magnetically based, optically based, and/or the like. Software modules are reusable portions of computer executable code having one or more specific functions.

[036] The Virtual Reality System includes a device where projected environments generate images, sounds, and other sensations that simulate a user's physical presence in the virtual environment and may include an interface device wherein a user can provide input to the virtual environment. The Virtual Reality System may further include a processing component, and any environmental stimulator configured to affect one of the senses of the user, such as, but not limited to, equipment configured to produce olfactory stimuli, orientation stimuli, or force stimuli. Virtual reality system may include, but is not limited to, the Oculus Quest, Oculus Go and Oculus Rift (Oculus VR, a division of Facebook Inc., Menlo Park, CA), the SAMSUNG HMD Odyssey+ (Samsung Group and Samsung Electronics, San Jose, California), or Google Cardboard VR platform (Google, LLC, Mountain View, CA). Augmented reality systems may include, but are not limited to, the Microsoft HoloLens (Microsoft Corporation, Redmond, WA) or the Vuzix Blade (Vuzix Corporation, West Henrietta, NY). The term virtual reality headset should be read to include the virtual reality system unless it is clear by the context that the virtual reality headset is intended to refer to the headset unit in isolation.

[037] The virtual environment may include an avatar, that is, a representation of the user within the virtual environment. The avatar is a virtual reality object and may be a photo-realistic representation of the user, may be a generic representation of the user, or may be a representation of a device controlled by the user. In some embodiments, the avatar may move in manner that mirrors or approximates the movements of the user. By mirroring or approximating the movements of the user, the avatar further enhances the virtual environment by creating a more immersive experience for the user by providing visual feedback for the user's real-world movements.

[038] The virtual environment may include a physics engine. A physics engine is a module of computer code that simulates physical and mass interactions for objects within the virtual environment. The physics engine may provide collision detection, that is, it provides a module of computer code that is used to determine whether a first object is intersecting a second object and raises events when the status of the intersection changes. For example, if the first object with a collision trigger scriptable action attached (described in more detail below)

intersects the second object, collision detection will raise an entered event and when each of the first and second objects no longer intersect, collision detection will raise an exited event. Each event has one or more properties that may include location of the collision as well as the properties of each of the objects in the collision, such as object velocity, object mass, and object material. Additionally, the physics engine provides dynamics simulations, that is, simulations of forces acting on one or more objects within the virtual environment, such as but not limited to the force of gravity or the simulation of the mechanics of materials.

[039] As used herein, the term Virtual Reality further includes supplemented reality such as augmented reality and mixed reality. Augmented reality, an interactive experience of a real-world environment where the objects that reside in the real-world are enhanced by computer-generated perceptual information, sometimes across multiple sensory modalities, including visual, auditory, haptic, somatosensory and olfactory modalities, and mixed reality, a merging of real and virtual worlds to produce new environments and visualizations where physical and digital objects co-exist and interact in real time, may use the same or similar equipment as the virtual reality system. Further, virtual environments may be interchangeable across each of the supplemented reality and virtual reality systems.

[040] A repository is a database or other file storage system stored either locally in the virtual environment generator component, on a computer in a local network, on a computer or server connected to the Internet, or otherwise available for the user 14 to access with permission from a repository owner. A repository owner is an entity that created or has full rights to the repository, including the rights to delete the repository, copy the repository, edit the repository, or share the repository with varying degrees of permissions to the user 14. Types of repositories include an online repository 304, a personal repository, and a core repository. The online repository 304 (described in detail below) is a repository that can store rooms, objects, and/or items available for reuse by any user provided access by the owner. The online repository 304 may or may not be connected to the internet, however, it is available via a network 420 (described in detail below). The online repository 304 may be located on a server either within the local network or on the internet. The personal repository is a repository of which the user 14 is the owner and is not accessible by other users unless the user 14 grants access permissions; further, objects, action scripts, scriptable actions, and other items may not be copied or edited by other users without permission from the owner, the user 14. The virtual reality world authoring system 10 also includes a core repository. The core

repository is a repository of rooms, objects, and/or items available to the user 14 that are provided by, owned by, and, by default, included with the virtual reality world authoring system 10. The user 14 has permissions to copy and then modify and use the objects within the core repository while the user creates virtual environments.

[041] Referring now to FIG. 1, shown therein and designated by reference numeral 10 is an exemplary embodiment of a virtual reality world authoring system having a user 14 with a VR headset 18 displaying a virtual environment 22 with objects 26. Also shown are the virtual environment generator component 28 storing one or more computer application having VR design features including an Object Builder 30, Environment Building Tools 34, a script builder 38, Trigger Objects 42, a physics engine 46, and a bidirectional state API 50.

[042] In one embodiment the user 14 is not a programmer and does not know a programming language. The user 14 uses the VR world authoring system to build the virtual environment 22 without writing computer code. The user 14 may build the virtual environment 22 alone or may work together with additional users 14. When working with others, user 14 may work on a local copy of the virtual environment 22 or may work in the virtual environment 22 shared with the other users 14.

[043] The VR headset 18 is a head-worn apparatus that covers the user's eyes for an immersive three-dimensional experience. The VR headset 18 may be self-contained, that is, the VR headset 18 may have a virtual environment generator component comprising a non-transitory memory storing computer code and a processor, that when executing the computer code, generates the virtual environment 22. In other embodiments, the VR headset 18 may be separate from the virtual environment generator component but instead be in communication with the virtual environment generator component through a communication component. The VR headset 18 may include other components such as a human interface device or environmental stimulators.

[044] The communication component is a component of the VR headset having circuitry or other hardware that implements a communication protocol that provides for communication between the VR headset 18 and other components. The communication protocol may include, but is not limited to, one or more of the following: wired communication, WiFi, Bluetooth, Wireless PAN, IR, or other radio-based or light-based communication conventions.

[045] In one environment, the user 14 uses the VR headset 18 to display the virtual environment 22 whereas in another environment the user 14 displays the virtual environment

22 on one or more other screens such that the displayed virtual environment 22 on the one or more screens immerses the user 14 in the virtual environment 22.

[046] The virtual environment 22 is established and run by the virtual environment generator component. The virtual environment 22 may be available to the user 14 only or it may be available to more than one user 14 at a time. In one embodiment, the virtual environment 22 is hosted on a server accessible by the user 14. The server may be connected to the headset over a network connection either locally (intranet) or externally (internet).

[047] The virtual environment 22 may be run by the user in at least a run mode or a design mode. In design mode, the user 14 experiences the virtual environment 22 in a more static simulation, that is, the user 14 can add or remove objects from the virtual environment 22 or may turn on or off physics and other object properties. In run mode, the user 14 experiences the virtual environment 22, with the simulation enabled, as though the user 14 is a part of the virtual environment 22. In order for the user 14 to enable run mode, the virtual environment 22 must first be created in design mode. When the simulation is enabled, the physics engine is fully enabled.

[048] In one embodiment, the virtual environment 22 comprises one or more simulation parameters and the plurality of virtual reality objects 26 (objects 26). An object is an empty container of items. Each item is a reference, comprising a pointer, to related files and content data associated with the object; an embedded file, that is, the object contains the file data for that specific item; or, another object. The object and each of the items contained in the object enjoin in a hierarchical relationship such that the object may be considered a parent to each of the items, which are in-turn, siblings to one another. In another embodiment, an item may also be a container for items, or subitems, such that the item is a parent to the subitems and is a child to the object. In one embodiment, the object 26 contains a metadata item, that is, a file that stores properties of the object 26 such as mass of the object 26, elasticity of object 26, and material of object 26. Each of the properties in the metadata item may be needed by the physics engine to properly simulate the interaction between the object 26 and any other object in the virtual environment.

[049] Referring now to Figure 2, shown is an exemplary embodiment of a car object 70 having a parent container 74 and a plurality of items including a 3d mesh 78, a texture 82, an audio 86, and an action script 90. As seen in Figure 2, each of the items attached to the parent container 74 of the car object 70 extend the car object 70. The plurality of items identified

above are for example purposes only and are not intended to be limiting. As defined above, an item is any file that may extend or provide additional information about the object 26. For example, items may also include document files, such as but not limited to doc, docx, ppt, pptx, pdf, and txt, hyperlinks, such as but not limited to URLs, video files, such as but not limited to mp4, avi, mpeg, or mov, or special purpose items.

[050] The 3D mesh 78 is a three-dimensional wireframe representation of the car object 70. It is a base model of an object before any texture or smoothing has been applied. Exemplary 3D mesh file types include, but are not limited to, stl, obj, fbx, collada, 3ds, vrml, and x3d.

[051] The texture 82 is a two-dimensional graphics file that contains the visual texture of the car object 70. When the car object 70 is rendered in three dimensions by the graphics processor, the texture 82 is applied to the 3D mesh 78. Exemplary 2D graphics file types include, but are not limited to, jpg, gif, png, bmp, and tiff.

[052] The audio 86 is an audio clip that is associated with the car object 70. The audio 86 may include any sounds that the car object 70 emits or may include any sound modifications to another sound file that may be played when another object interacts with the car object 70. Exemplary audio file types include, but are not limited to, midi, mp3, wav, and ogg.

[053] Special purpose items are items that exhibit complex functionality which would otherwise be difficult to obtain without programming and can be attached to an object similar to any other item. Special purpose items enable users 14 to build feature rich virtual environments without programming. In one embodiment, special purpose items may include, but are not limited to, items such as a video player item, an audio player item, a document viewer item, or a quiz item.

[054] The video player item provides the functionality to play video file items, such as those attached to objects, or video streams, such as those at the end-point of a hyperlink object. By providing this item, a user can stretch this item over a 3D mesh to form a virtual display such as, but not limited to, a virtual TV screen, a movie screen, or a computer monitor.

[055] Similarly, the audio player item provides the functionality to play audio file items, such as those attached to the object, or audio streams, such as those at the end-point of a hyper-link object. The user 14 may implement this item into simple features such as a radio within the car object 70.

[056] The document viewer item provides the code to allow the user 14 to implement a presentation by implementing the document view item on a mesh. Similarly, the quiz item may

be implemented by the user 14 by attaching the quiz item having multiple elements to a mesh where each element relates to an answer, output of the quiz item may be the index of the selected answer.

[057] An item is embedded in an object when the item's file data is saved together with the file data for the object. For example, the 3d mesh 78 is said to be embedded in the car object 70 when the 3d mesh 78 data is serialized and saved within the same file as the parent container 74 irrespective of the encoding used. Therefore, moving the file location of the parent container 74 also moves the data for the 3d mesh 78. Similarly, deleting the parent container 74 also deletes the embedded data including the data for the 3d mesh 78.

[058] An item is referenced by an object when a link or reference is stored within the parent container 74 of the car object 70. The link or reference may be, for example purposes only, a pointer to any of a location in memory, a file system, a web resource such as a hyperlink, or a database. When the item is referenced, the item's encoded data is not stored with the parent container 74 and is not serialized and packed with the parent container 74. Therefore, not only is the parent container 74 of a smaller size on the computer memory, but also, were the parent container 74 deleted, the pointer to the item will be deleted, however, the item data will not be deleted.

[059] In order to implement the object 26 within the virtual environment 22, the object 26 must first be configured. Referring now to Figure 3, shown is an exemplary embodiment of an object configuration process 100. Generally, the object configuration process comprises the steps of loading a 3d mesh from a selected repository (step 104), automatically creating the parent object (step 108), and adding one or more items to the parent container (step 112). In one embodiment, adding one or more items to the parent container (step 112) may further include using coordinate links to link objects to one another in order to form a complex object.

[060] Referring now to Figure 4, shown therein is an exemplary embodiment of a wheel object 120 having a wheel parent container 124, a 3d mesh 128, a texture 132, and an action script 136. The wheel parent container 124 contains references to or copies of the 3d mesh 128, the texture 132 and the action script 136. The 3d mesh 128 is a three-dimensional wireframe structure representing a wheel. The wireframe structure is comprised of points or vertices joined by lines or edges. Each of the vertices is connected to another vertex by at least one edge. The texture 132 is a texture file that, when processed with the 3d mesh 128, wraps the 3d mesh 128 in an image. For example, the texture 132 may wrap the 3d mesh 128 to

cause the 3d mesh 128 to appear like a wheel would in real-life. In other embodiments, the texture 132 is calculated based on other properties of the object, such as the object material and the properties of that material.

[061] Referring now to Figure 5, shown therein is an exemplary embodiment of a complex car object 150 having one car object 70 and four wheel objects 120a-d. The 3d mesh 78 of the car object 70 exists in three-dimensional space and has coordinates defining the boundary of the 3d mesh wireframe, the coordinates being a location in three-dimensions defined, for example, by cartesian coordinates in (X, Y, Z) form. The wheel objects 120a-d are attached to the car object 70 by coordinate links 154a-d such that the first coordinate link 154a links the coordinates of the wheel object 120a to the first coordinates of the car object 70, the second coordinate link 154b links the coordinates of the wheel object 120b to the second coordinates of the car object 70, the third coordinate link 154c links the coordinates of the wheel object 120c to the third coordinates of the car object 70, and the fourth coordinate link 154d links the coordinates of the wheel object 120d to the fourth coordinates of the car object 70. In one embodiment, the coordinate link includes a three-dimensional range, also called connection strength, which provides a range within which the objects are connected to one another.

[062] The coordinate links 154a-d maintain the relative distance between each of the wheel objects 120a-d and the car object 70 thereby attaching each of the wheel objects 120a-d to the car object 70. Each of the complex car object 150 items, car object 70 and wheel objects 120a-d, may operate independently of one another while maintaining a physical connection with the connection links 154a-d.

[063] As shown in an exemplary object relationship model 158 in Figure 6, the parent container 74 and the wheel containers 124a and 124b (as well as 124c and 124d, not shown) are joined in a lateral relationship through the connection links 154a-d, rather than a hierarchical relationship, in order to form the complex car object 150.

[064] As shown in another exemplary hierarchical object relationship model 158a, the complex car object 150 may be formed through a hierarchical, parent-child relationship between the car object 70 (the parent) and each of the wheel objects 120a-d (the children), as shown in Figure 7. Again, wheel object 120c and 120d are not shown for simplicity. The hierarchical object relationship model 158a shows the wheel containers 124a and 124b at the same hierarchical level as other items in the parent container 74 such as the 3d mesh 78, the

texture 82, the audio 86, and the action script 90 while the items within each of the wheel containers 124a, 124b are of a lower hierarchical level than the items within the parent container 74.

[065] In one embodiment, objects in a parent-child relationship forming a complex object do not maintain relative distance as determined by a connection strength between the objects, rather, the child objects are attached to the parent object such that any directional movement of the parent object likewise moves the child objects all the while maintaining the ability of each of the objects to interact independently of the complex object through the use of the action script item.

[066] The user 14 places the one or more objects within the virtual environment using an Environment Building System. As shown in Figure 8, while in a room 160, the Environment Building System provides the user 14 with a room settings menu 170 that enumerates controls including a room name field 174, a gravity control 178, a scale control 182, a private control 186, an editable control 190, an anonymously publish control 194, a run control 198, a reset control 202, a destroy control 206, a publish control 210, or a settings menu 214. Also shown in the virtual environment 22 are virtual hands 220a and 220b and a cursor 224.

[067] The cursor 224 is a virtual environment representation of a point within the virtual environment 22 that will be affected by input from the user 14 and is moveable by the user 14. For example, the cursor 224 may be a line extending from a first location within the virtual environment 22, that the user 14 would interpret as coinciding with the user's hand, to a second location within the virtual environment 22, that the user 14 would interpret as being the point within the virtual environment 22 selected by the user's hand. In another embodiment, the cursor 224 is a geometric figure located at the second location within the virtual environment 22 that the user 14 would interpret as being the point within the virtual environment 22 corresponding to the user's hand.

[068] The room 160 is a specific virtual environment 22 that the user 14 has made or downloaded. The room 160 provides the initial structure of the virtual environment 22 and can store environment settings that are applicable to any objects within the room 160. For example, the user 14 may use the gravity control 178 within the room settings menu 170 to adjust the gravity within the room. By using the gravity control 178, the user 14 adjusts how gravity behaves against all objects within this specific virtual environment 22. Additionally, the user 14 can use the scale control 182 to scale the room. In one embodiment, the scale control

182 will adjust the relative sizes of all objects within the room 160 such that, from the user's perspective, the user appears to grow or shrink. By selecting the private control 186, the user 14 can prevent the room 160 from being searchable within the repository. The user 14 can also adjust whether their room 160 is editable by other users by toggling the editable control 190. The user 14 also has the option to publish their room 160 anonymously to one or more repositories by using the anonymously publish control 194. By using the reset control 202, the user 14 can revert changes back to the most recent save. The room settings menu 170 also allows the user 14 to delete the room 160 and all instances of objects within the room 160 by selecting the destroy control 206, thereby reverting the room to a blank virtual environment similar to the environment upon initializing a new room. Upon selection, the publish control 210 allows the user 14 to publish the room 160 to one or more repository consistent with other settings within the room settings menu 170, such as anonymous publishing, whether the room is private, and whether the room is editable. Additionally, by publishing the room, all objects within the room are also published, thereby enabling a third-party user to download and open a fully functional room with all custom objects, mirroring the room as it was most recently published by the user 14. In another embodiment, upon selecting the publish control 210, the user 14 is presented with a repository menu thereby allowing the user to select the repository to which the room 160 is published. The room settings menu 170 also provides a settings menu 214 that enables the user 14 to select a different menu within the Environment Building System, such as but not limited to those menus described in more detail below.

[069] In one embodiment, the room 160 can be saved as a room save file containing all settings for the room, such as those settings adjusted within the room settings menu 170, as well as references to all of the objects within the room. For each object within the room, the room save file stores default settings including, but not limited to, a starting position for the object, a starting rotation for the object, a starting scale for the object, whether the object is affected by gravity, whether the object is grabbable, and the like. Object default settings such as the starting position, starting rotation, starting scale, and other similar properties may have values corresponding to three dimensions, such as X, Y, Z values for positioning or scale, and angular rotation values for the starting rotation for the object. The room 160 is saved with the value of the room name field 174, thereby enabling the user 14 to retrieve the room through an easily recognizable label, the room name.

[070] In one embodiment, the room settings menu 170 further includes a save control enabling the user 14 to save the room 160 in its current state to a save file having a save name. When the save control is selected, a save file location screen may or may not load, enabling the user 14 to save the room 160 in a different file location than the most recent save file. Alternatively, when the user 14 selects the save control to save the room 160, the room 160 is saved to the same save file as the most recent save file. The save file may be titled the same as the room name field 174 value, or the save file may further include a datetime stamp in the save name, thereby enabling the user 14 selecting a virtual environment to load to select a previous save file if so desired. In one embodiment, when the publish control 210 is selected, the room 160 is automatically saved or a save confirmation screen appears offering the user 14 at least the same options as the save control does.

[071] Referring now to Figure 9, shown therein is an exemplary embodiment of a tools menu 230 for the room 160 within the virtual environment 22. The tools menu 230 lists a suite of three-dimensional object manipulation tools including a move tool 234, a rotate tool 238, a scale tool 242, an info tool 246, a wire tool 250, a group tool 254, an ungroup tool 258, a duplicate tool 262, and an erase tool 266. The tools menu 230 also provides an undo tool 270 and the settings menu 214. Also shown within the room 160 are the virtual hands 220a and 220b as well as the cursor 224. While only the above tools are listed in the tools menu 230, the tools menu 230 is not an exhaustive list of the tools available to the user 14. Other tools may include those that would allow a user to copy dimensions from one object to another, copy attributes, such as texture or color, from one object to another, copy one or more object items from one object to another object, reset an object to its base settings, and/or other object manipulations. Further, the organization of the tools within the tool menu 230 is for convenience only, and the tools can be organized in other arrangements. In one embodiment, the user 14 can move one or more tools within the tool menu 230 to reorganize the tools in an order that the user 14 finds most convenient.

[072] The move tool 234 allows the user 14 to select an object within the room 160 and move that object to another location within the room 160. Similarly, the rotate tool 238 allows the user 14 to rotate a selected object in three-dimensional space, and the scale tool 242 allows the user to scale an object, that is increase or reduce its size, within the room 160. Other tools include the info tool 246 that will display the properties and metadata of a selected object, the duplicate tool 262 that will duplicate a selected object and its children, and the erase tool 266,

that will erase the selected object and its children items. Tools available for working with or creating complex objects further include the wire tool 250, which provides a method for the user to wire objects together using coordinate links 154, the group tool 254, which allows a user to group a selection of objects together forming a complex object comprising the selected objects, and an ungroup tool 258, which separates a complex object into its constituent objects. In one embodiment, the ungroup tool 258 can also separate an object and its items, that is removes the one or more items from the parent container, such that the items are independent of the object and are then considered objects themselves. The tools menu 230 also provides the undo tool 270 that allows the user 14 to undo, or reverse, the most recent change to one or more objects within the room 160. The tools menu 230 similarly includes the setting menu 214, which enables the user 14 to select a different menu within the Environment Building System, such as but not limited to the room settings menu 170 or a scriptables menu 340 (described in detail below).

[073] Referring now to Figure 10, showing an exemplary embodiment of a network diagram 300 having an online repository 304 and a virtual environment component 308 hosted on one or more servers 312 that enable a second user 316 and a third user 320 to connect to a virtual environment 22 developed and owned by user 14. The virtual environment component 308 comprises a hosted environment 324 and a user state synchronization component 328. The hosted environment 324 is the virtual environment owned by user 14 and may be uploaded and hosted online from within the virtual reality world authoring system 10. The user 14 may allow others, in this case the second user 316 and the third user 320, to join the hosted environment 324. The user 14 may also prohibit the second user 316 and the third user 320 from copying the hosted environment 324, the virtual environment 22, any of the one or more objects within the virtual environment, or any of the one or more items within the virtual environment into that user's personal repository or into any other environment that user has access to. The permissions of controlling the access to the hosted environment 324, the virtual environment 22 or the like can be implemented using standard computer techniques, such as access control lists, and the like.

[074] Any room created is said to be persistent, that is, the room, if saved, will store the state and location of all objects within the room at a first point in time, and when the user opens the room at a second point in time the state and location of all the objects within the room is the same as at the first instance in time. At a second instance in time after the first instance in time,

the user 14 can load the room 160 in design mode and edit the room 160. Additionally, the user 14 can copy the room 160 to create a second room. The objects and items in the second room will have the same state and location as all the objects and items in the room 160 at the point in time at which the second room was created.

[075] In one embodiment, there is an online marketplace wherein individuals can give or sell one or more action modules to a user 14 or submit actions to one or more repositories. A user 14, in need of a new action module or set of action modules can connect to one or more additional repositories wherein additional action modules or sets of action modules are available. By connecting to the repository, the user 14, will gain access to the action modules or sets of action modules available in the repository for use within the user's virtual environment 22.

[076] The user 14 or other users 14 can search within one or more repository for one or more rooms 160 that are searchable. In one embodiment, the search can be performed based off of the room name field 174 and/or the user's username or handle if the user 14 has not selected the anonymously publish control 194. In other embodiments the search can be made on environmental settings such as gravity level, theme, and/or any other searchable characteristic of the room 160.

[077] When the second user 316 or the third user 320 connect to the hosted environment 324, the connecting user 316 or 320 will automatically connect to the online repository 304 and download the room, objects, and items needed to virtualize the hosted environment 324 before joining the room 160. If the connecting user 316 and/or 320 have permissions to join the room 160, each likewise have at least read permissions for each of the objects and items within the room 160.

[078] Additionally, in order for the second user 316 or the third user 320 to connect to the hosted environment 324, each of the connecting users' coordinates within the virtual environment 22, each of the connecting users' necessary state parameters, and all objects and items are synchronized through the user state synchronization component 328 and sent to each of the one or more users within the hosted environment 324. Thus, when the user 14 moves within the hosted environment 324, both the second user 316 and the third user 320 observe the user 14 movements as though each of the users could see the others in person. The user state synchronization component 328 further provides any connected user the ability to see each other and move around the hosted environment 324, chat with one another

utilizing voice or audio communication components, handle and transfer objects from one user to another, and create notes that display at a specific three-dimensional coordinate point within the hosted environment 324.

[079] In one embodiment, the server 312 is a component of the user's virtual environment generator component. Here, the second user 316 and the third user 320 connect to the user's virtual reality environment component through a network. As discussed more below, the network can be the internet, a local area network, or any other data connection protocol that provides access to the user's virtual environment generating component by the second user 316 or the third user 320.

[080] The virtual reality world authoring system 10 provides the script builder 38 to users 14 enabling the user 14 to create interactive and complex objects within the virtual environment 22. The virtual reality world authoring system 10 does not require the user 14 to write programming code in order to create complex objects through the use of action scripts.

[081] Referring now to Figure 11, shown therein is the virtual environment 22 of the room 160 having an exemplary embodiment of the scriptables menu 340, the scriptables menu 340 comprising a plurality of scriptable functions 344a-n. The scriptable functions are computer functions having at least one input and at least one output. Scriptable functions may include functions used for manipulating three-dimensional objects dynamically, such as a mover scriptable that will, with an animated effect, move an object from one location to another; a rotator scriptable that describes the rotation of an object in three dimensions, an object position scriptable that immediately sets the three dimensional position of the object, and a disable scriptable that disables action scripts attached to the object. Scriptable functions may also include mathematical functions wherein the inputs may include values to be evaluated and the output is the result of the mathematical function. The mathematical functions may include add, multiply, logical AND, logical OR, logical XOR, and logical NOT. One or more mathematical functions may be connected to the input or the output of another scriptable in order to modify the input or output values respectively. Other scriptable functions may include functions related to time or the passage of time, such as delay or always on. The scriptable functions available can be expanded. For instance, if the user 14 routinely uses the NAND logical operation, the user 14 may form a new scriptable function by linking the logical not scriptable function output to the logical AND scriptable function input and saving the combined scriptable functions into a new, NAND scriptable function. The scriptables menu 340 further

has the settings menu 214, which, as described above, enables the user 14 to select a different menu within the Environment Building System, such as but not limited to the room settings menu 170 or the tools menu 230.

[082] Action scripts are modules of Turing-complete computer programmable code comprising one or more scriptable functions. By utilizing action scripts, the user 14 can build complex objects by wiring together action modules without understanding a computer programming language.

[083] Referring now to Figure 12, shown therein is an exemplary embodiment of a script builder diagram 360. Here, the trigger object 42 (discussed in more detail below) is made available within the room 160. Associated with the trigger object 42 is a trigger scriptable action 362. The trigger scriptable action output 364a is connected by scriptable connector 368a to the rotator scriptable action 344a. The scriptable connector 368a is a visual representation of the path from the trigger scriptable action output 364a to the rotator scriptable action active input 372a. By connecting a first output to a first input with the scriptable connector 368a, the user 14 is directing the script builder 38 to use the value of the first output as the input value to the first input. Also shown is a static rotator scriptable action X input 372b of 45 degrees, that is, 45 degrees is a pre-set input to the rotator scriptable action 344a that represents the number of degrees to rotate in the "x" direction within three-dimensional coordinate space. The rotator scriptable action 344a also accepts an input for the "y" direction and the "z" direction. The rotator scriptable action object output 364b is connected, by scriptable connector 368a, to a bathtub object input 372c of a bathtub object 26a. By connecting the rotator scriptable action object output 364b to the bathtub object input 372c of the bathtub object 26a, the user 14 is indicating that the rotator scriptable action 344a is to be applied to the bathtub object 26a, thereby causing the bathtub object 26a to rotate 45 degrees in the "x" direction when the trigger object 42 calls the rotator scriptable action 344a. By connecting the trigger scriptable action 362 to the rotator scriptable action active input 372a, the user 14 is able to configure when the rotator scriptable action 344a is active and when the rotator scriptable action 344a is inactive based on the output of the trigger scriptable action 362. The connected trigger scriptable action 362, rotator scriptable action 344a, and object property 376 create a first action script that is then stored as an item in the bathtub object 26a parent container. When the script builder diagram 360 is saved, the virtual reality

world authoring system 10 processes the script builder diagram 360 and compiles the diagram into a computer language.

[084] The same complex object behavior illustrated by the script builder diagram 360 can be created using a textual script builder. If the user 14 is so inclined, the user 14 can use the textual script builder to program the same bathtub object 26a as created with the script builder 38. The first action script can be written in a text editor and subsequently import the text file into the room 160 of the virtual environment 22. When the virtual environment 22 receives the text file, the virtual environment 22 transforms the first action script into a script builder diagram 360. An exemplary text file may be:

```
//TriggeredRotator.act – commented out filename
[fActivated] > Trigger > [fTriggerCheck]
[fTriggerCheck 45 $ $] > Rotate > [aRotateAction]
```

[085] In the above code, the "fActivated" is a function that receives a condition for the Trigger scriptable action and outputs the fActivated function result and calls the fTriggerCheck function. The fTriggerCheck function receives three parameters, 45, $ and $, where the $ represents a null, or nothing, value. The input values to the Rotate scriptable action are processed output to the aRotateAction as input to the action input of the bathtub object 26a.

[086] In some embodiments, the script builder diagram 360 may show scriptable actions that are not connected to any other scriptable actions. In that case, the scriptable action with no connections to its output will not have any effect on the object 26. For instance, in the script builder diagram 360, id the scriptable connector 368a were not there, the output of the trigger scriptable action 362 would have no effect on the bathtub object 26a, thereby causing the bathtub object 26a to continuously rotate if the rotator scriptable action active input 372a were set to true.

[087] Additionally, the user 14 may desire to create a complex action script with many separate scriptable actions involved. One solution is for the user 14 to select a second action script that is similar to the desired action script, download the second action script and make minor modifications to achieve the desired results. Moreover, when the user 14 searches for and finds an action script, the user 14 may drag the action script from the search and onto the object within the virtual environment 22 to which the user 14 desires to add the action. In

other embodiments, actions scripts may include complex actions such as collision avoidance and/or A* algorithm path following, among other complex three-dimensional actions.

[088] Trigger objects 42 are objects having a trigger scriptable action 362 that, when activated, causes one or more scriptable actions attached to one or more objects 26 to activate. The trigger scriptable action 362 may be activated when an object 26, including a user's avatar, interact with the trigger object 42 in a pre-specified manner, such as the user's avatar standing on the trigger object 42 or the object 26 landing on or passing through the trigger object 42. In one embodiment, the interaction may be determined by the physics engine such that if one or more objects 26 are stacked on the trigger object 42 and the one or more objects 26 have a total mass or weight above a trigger threshold, then the trigger scriptable action 362 will be activated. In this scenario, the sum of the objects' mass would be directed to the input of the trigger scriptable action 362. In other embodiment, any property of one or more objects 26 interacting with the trigger object 42 may be used as the input to the trigger scriptable action 362.

[089] Action scripts are created by the script builder 38 and used to add interactivity to the virtual environment 22. By creating action scripts that include trigger scriptable actions, the user 14 is able to quickly and easily create complex objects within the virtual environment without programming knowledge. While the breadth and variety of uses for triggers is bound only by the user's imagination, some other examples of using triggers to create interactive complex objects include adding a trigger to a button trigger object such that when pressed, an elevator door opens, adding a trigger to a gun trigger such that when the gun trigger is pulled, the gun fires, or adding a crossing trigger to a doorway such that when the user's avatar crosses the doorway threshold, a quiz object becomes viewable and requests the user 14 for an answer to the quiz presented by the quiz object. In another embodiment, the input to a trigger scriptable action 362 may be a time delay scriptable action such that when the time delay ends, an activate signal is sent from the output of the time delay scriptable action to the input of the trigger scriptable action.

[090] In one embodiment, collision detection is not enabled for objects 26 by default. Instead, to enable collision detection, a trigger object is added, scaled, and connected in a hierarchical, or parent-child, relationship to the object 26. By adding the trigger object 42 and surrounding the object 26, any triggering object that would interact with the object 26 must first pass through the trigger object 42 and thus activate the trigger scriptable action 362. When the

triggering object enters the trigger object 42, an entered() event for that trigger is raised and an entered() event message is created. Likewise, when the triggering object is no longer intersecting with the trigger object, the exited() event is fired. An event handler, and or message receiver, handles the entered() or exited() event or message and activates the trigger scriptable action.

[091] In one embodiment, the dynamics simulation of the physics engine limits the calculated laws of physics to collision detection, Newton's second law and gravity, and rigid body dynamics. If an object 26 within the virtual environment 22 has the physics property set to on and a valid mass property, the physics engine will subject the object 26 to the physics simulation. The rigid body dynamics simulation of the physics engine must account for the angular rotational behavior of objects 26. The combined effects of the calculated laws of physics provide a realistic simulation for most conditions of use. In another embodiment, additional physics simulations are utilized, such as, but not limited to, soft-body dynamics and/or electro-magnetic forces.

[092] In some embodiments, the virtual reality world authoring system 10 further provides a bidirectional state API 50. The bidirectional state API 50 provides bidirectional state information from the real world into the virtual environment 22 and from the virtual environment 22 into the real world. When implementing the bidirectional state API 50, a real-world object may be modeled into a model object that acts as a proxy within the virtual environment 22. In one embodiment, a heart-rate monitor may be attached to the user 14. The heart-rate monitor may then be connected to the bidirectional state API and send heart-rate data from the heart-rate monitor to the virtual environment 22. A heart-rate model object within the virtual environment 22 may receive the heart-rate data as an input and connect the output from the heart-rate model object to the input of a display object thereby providing the user 14 with a display of the user's real-world heart-rate within the virtual environment 22. Likewise, information from within the virtual environment 22 can be transmitted outside the virtual environment 22 and into the real-world. For example, the user's avatar's arm movement data may be recorded and transmitted through the bidirectional state API 50 and into the real world. In the real-world, the arm movement data may be collected and sent to a robotic arm such that the movements of the user's avatar within the virtual environment are mirrored by the robotic arm within the real-world.

[093] One such need for the bidirectional state API 50 is in remote robotic manipulation within hazardous areas. A robot having one or more cameras could be deployed to a hazardous area that is inaccessible to or hazardous to a human person. The one or more cameras on the robot can collect imagery from the real-world and transmit the imagery to the virtual environment. The virtual environment 22 can then use the real-world imagery as texture files for objects generated to reflect the object in the real-world imagery. The user 14, having an avatar of the robot, could then interact with the objects, such interaction can then be mirrored in the real-world by collecting the robot avatar's movement data and transmitting that data back through the bidirectional state API 50 to the robot, thereby causing the robot to mirror, in the real-world, the movements of the robot avatar.

[094] The bidirectional state API 50 is an event processing system and it places events on a queue. The API events relate to messages that should be sent and received by the system. For example, the API event handler may implement the following pseudo-code wherein a received message calls a message_received function and a sent message calls a message_sent function:

[095]
```
switch(event.type)
{       case: "api_message_received:" api_message_recv(state);
        case: "api_message_send"  :   api_message_send(object);
}
```

[096] Exemplary pseudo-code of the above functions may be expressed as:

```
api_message_recv(state)
{       objectname = state.objectname
        object = getobjectwithname(objectname)
        object.property1 = state.property1
        object.property2 = state.property2
        object.property3 = state.property3
        …..
}

api_message_send(object)
{       state = new state
        state.objectname=object.name;
        state.property1 = object.property1
```

```
        state.property2 = object.property2
        state.property3 = object.property3
        ....
        send_state(state)
}
```

[097] The send state function sends state information when an api_message_send event occurs. It can do this through any one or more of a variety of protocols both connected and stateless by wrapping the message, serializing it, and sending through the network. The one or more protocols may include http, https, telnet, web sockets, signalr, sockiet.io, or tcp/ip sockets.

[098] Referring now to Figure 13, shown therein is an exemplary embodiment of the virtual reality world authoring system 10 wherein the user 14 is within the virtual environment 22, the virtual environment 22 being a basketball court having a complex basketball object 380, a basketball object 382, a basketball trigger object 384, a basketball goal object 388 and a complex basketball rim object 392.

[099] To create the complex basketball object 380, the user 14 creates the basketball object 382 having items including a basketball texture, a basketball bounce noise, a basketball metadata item storing basketball physical properties such as mass and elasticity, and a bounce action script. The user 14 also creates the basketball trigger object 384, the basketball trigger object 384 being a spheroid with a radius greater than that of the basketball object. The user 14 connects the basketball object 382 and the basketball trigger object 384 at their center points, thereby causing the basketball trigger object 384 to fully encompass the basketball object 382 within the virtual environment 22 and creating the complex basketball object 380. The bounce action script may take as input values the basketball physical properties including mass and elasticity, as well as a velocity value, to calculate the bounce action.

[100] In order for the complex basketball object 380 to bounce, there must be a force acting on the complex basketball object 380, therefore, collision detection is implemented along with a trigger scriptable action attached to the basketball trigger object 384. With collision detection, when an interfering object intersects with the basketball trigger object 384, a trigger event is raised. The trigger handler responds to the event and activates the trigger scriptable action by calling the input of the trigger scriptable action. When the user 14 selects the trigger scriptable action from the scriptable menu, the collision detection and event handling are

automatically implemented for the trigger scriptable action. The user 14 can then enter the script builder 38. Here, the user 14 can connect the output of the trigger scriptable action of the basketball trigger object 384 to the input on the bounce scriptable action of the basketball object 382 with a scriptable connector. Other inputs to the bounce scriptable action may include the velocity and/or mass of the interfering object, thereby preventing the bounce scriptable action from causing the complex basketball object 380 from bouncing when the complex basketball object 380 intersects a trigger object or another object that should not have an impact on the complex basketball object. The output of the bounce scriptable action can then be connected to the input action on the basketball object by a second scriptable connecter, the connections from the trigger scriptable action to the bounce scriptable action to the basketball object thereby causing the basketball to bounce when the trigger scriptable action is activated by collision detection.

[101] Similarly, the complex basketball rim object 392 is created by forming a hierarchical relationship between a basketball rim object and a basketball rim trigger object having a rim trigger scriptable action where the basketball rim trigger object is a circular plane circumscribed by the rim and the center-points of both objects are connected. Here collision detection is again needed to test whether one or more interfering objects intersects the basketball rim trigger object. The trigger scriptable action may have a condition that the intersecting object is the complex basketball object in order for the trigger scriptable action to activate the next scriptable action. The user 14 can then enter the script builder 38. Here, the user 14 can connect the output of the rim trigger scriptable action of basketball rim trigger object to the input on a goal scriptable action of a scoreboard object with a scriptable connector. Other inputs to the goal scriptable action are complex basketball object 380 properties such as which team last had control of the ball and a location on the court from which the complex basketball object 380 was released. The output from the goal scriptable action may include the number of points to increase the score and the team for which to increase the score. The output of the goal scriptable action can then be connected to the input action on the scoreboard object by a second scriptable connecter, the connections from the rim trigger scriptable action to the goal scriptable action to the scoreboard object thereby causing the scoreboard to change scores when the rim trigger scriptable action is activated by collision detection.

[102] Referring now to Figure 14, shown is an exemplary embodiment of the virtual reality world authoring system 10 carried out on one or more computer system 400. The computer system 400 may comprise one or more processing components 404, each processing component 404 having a computer processor 408, one or more non-transitory memory 412, and may include (or be communicatively coupled with) one or more communication component 416. The non-transitory memory 412 may store computer code that implements the virtual reality world authoring system 10. The computer system 400 may include a network 420 enabling bidirectional communication between the processing component 404, the one or more VR headsets 18, and/or a server system 424. The computer processor 408 or multiple computer processors 408 may or may not necessarily be located in a single physical location. In one embodiment, the processing component 404 is integrated with the VR headset 18.

[103] In one embodiment, the network 420 is the Internet, and the VR headsets 18 interfaces with the computer processor 408 via the communication component 416. It should be noted, however, that the network 420 may be almost any type of network and may be implemented as the World Wide Web (or Internet), a local area network (LAN), a wide area network (WAN), a metropolitan network, a wireless network, a cellular network, a Bluetooth network, a Global System for Mobile Communications (GSM) network, a code division multiple access (CDMA) network, a 3G network, a 4G network, a 5G network, a satellite network, a radio network, an optical network, a cable network, a public switched telephone network, an Ethernet network, combinations thereof, and/or the like. It is conceivable that in the near future, embodiments of the present disclosure may use more advanced networking topologies.

[104] In one embodiment, the processing component 404 having the computer processor 408 and the non-transitory memory 412 may be implemented with the server system 424 having multiple servers in a configuration suitable to provide a commercial computer-based business system such as a commercial web-site and/or data center.

[105] The following is a number list of non-limiting illustrative embodiments of the inventive concept disclosed herein:

[106] 1. An electronic device comprising:

a processor;

a headset having a visual feedback screen;

one or more user interface devices; and

a non-transitory computer readable medium storing computer executable code that when executed by the processor cause the processor to:

project a mixed reality environment to the headset, the mixed reality environment having one or more objects, each of the one or more objects being created by object computer code having properties that define how the one or more objects interact within the mixed reality environment;

in response to a first input from at least one user interface device, display an action menu on the visual feedback screen having a list of pre-defined action modules, the action modules including a module of computer executable code having one or more functions, the one or more functions configured to affect one or more properties of a first object of the one or more objects within the mixed reality environment;

update the object computer code of the first object to include a selected action module.

[107] 2. The electronic device of illustrative embodiment 1, wherein the first object is identified before the first input.

[108] 3. The electronic device of illustrative embodiment 1, wherein the first object is identified after the first input.

[109] 4. The electronic device of any one of illustrative embodiments 1-3, wherein the non-transitory computer readable medium storing computer executable code that when executed by the processor cause the processor to, in response to a triggering event, update the object computer code of the first object to include a selected action module.

[110] 5. The electronic device of illustrative embodiment 4, wherein the triggering event is raised when a user interacts with a trigger object thereby meeting a pre-determined trigger condition.

[111] 6. The electronic device of any one of illustrative embodiments 1-5, wherein the non-transitory computer readable medium storing computer executable code that when executed by the processor cause the processor to, in response to a second input from at least one user interface device selecting a specific trigger object from a trigger menu, display a trigger condition menu having a list of one or more trigger conditions for the trigger object; and in response to a selection of a first trigger condition for the trigger object, allows a user to place the trigger object into the mixed reality environment, the trigger condition, when met, causes the processor to update the object computer code of the first object to include a selected action module.

[112] 7. An electronic device comprising:

a processor;

a headset having a visual feedback screen;

one or more user interface devices; and

a non-transitory computer readable medium storing computer executable code that when executed by the processor cause the processor to:

project a mixed reality environment to the headset, the mixed reality environment having one or more objects, each of the one or more objects being created by object computer code having properties that define how the one or more objects interact within the mixed reality environment;

in response to a first input from at least one user interface device, selecting a specific trigger object from a trigger menu, display a trigger condition menu having a list of one or more trigger conditions for the trigger object; and in response to a selection of a first trigger condition for the trigger object, allows the user to place the trigger object into the mixed reality environment.

[113] 8. The electronic device of illustrative embodiment 7, wherein the trigger object is identified before the first input.

[114] 9. The electronic device of illustrative embodiment 7, wherein the trigger object is identified after the first input.

[115] 10. The electronic device of any one of illustrative embodiments 7-9, wherein the non-transitory computer readable medium storing computer executable code that when executed by the processor cause the processor to, in response to a triggering event, update the object computer code of the trigger object to include a selected action module.

[116] 11. The electronic device of illustrative embodiment 10, wherein the triggering event is raised when a user interacts with the trigger object thereby meeting a pre-determined trigger condition.

[117] 12. A non-transitory computer readable medium storing computer executable code that when executed by a processor cause the processor to:

project a mixed reality environment to a headset, the mixed reality environment having one or more objects, each of the one or more objects being created by object computer code having properties that define how the one or more objects interact within the mixed reality environment;

in response to a first input from at least one user interface device, display an action menu on a visual feedback screen having a list of pre-defined action modules, the action modules including a module of computer executable code having one or more functions, the one or more functions configured to affect one or more properties of a first object of the one or more objects within the mixed reality environment;

update the object computer code of the first object to include a selected action module.

[118] 13. The non-transitory computer readable medium of illustrative embodiment 12, wherein the first object is identified before the first input.

[119] 14. The non-transitory computer readable medium of illustrative embodiment 12, wherein the first object is identified after the first input.

[120] 15. The non-transitory computer readable medium of any one of illustrative embodiments 12-14, wherein the non-transitory computer readable medium storing computer executable code that when executed by the processor cause the processor to, in response to a triggering event, update the object computer code of the first object to include a selected action module.

[121] 16. The non-transitory computer readable medium of illustrative embodiment 15, wherein the triggering event is raised when the user interacts with a trigger object thereby meeting a pre-determined trigger condition.

[122] 17. The non-transitory computer readable medium of any one of illustrative embodiments 12-16, wherein the non-transitory computer readable medium storing computer executable code that when executed by the processor cause the processor to, in response to a second input from at least one user interface device selecting a specific trigger object from a trigger menu, display a trigger condition menu having a list of one or more trigger conditions for the trigger object; and in response to a selection of a first trigger condition for the trigger object, allows the user to place the trigger object into the mixed reality environment, the trigger condition, when met, causes the processor to update the object computer code of the first object to include a selected action module.

[123] 18. An electronic device comprising:

a processor;

a headset having a visual feedback screen;

one or more user interface devices; and

a non-transitory computer readable medium storing computer executable code that when executed by the processor cause the processor to:

project a virtual reality environment to the headset, the virtual reality environment having one or more object, each of the one or more object being created by a container having properties that define how the one or more object interact within the virtual reality environment, the one or more object having a three dimensional mesh with coordinates defining a boundary of the one or more object;

selecting a first object of the one or more object;

in response to a first input from at least one user interface device, display an action menu within the virtual reality environment on the visual feedback screen, the action menu having a list of pre-defined action scripts, the action scripts including a module of computer executable code having one or more function, the one or more function configured to affect one or more property of the first object of the one or more object within the virtual reality environment; and

update the container of the first object to include a selected action script.

[124] 19. The electronic device of illustrative embodiment 18, wherein the first object is identified before the first input.

[125] 20. The electronic device of illustrative embodiment 18, wherein the first object is identified after the first input.

[126] 21. The electronic device of any one of illustrative embodiments 18-20, wherein the non-transitory computer readable medium storing computer executable code that when executed by the processor cause the processor to, in response to a triggering event, update the container of the first object to include a selected action script.

[127] 22. The electronic device of illustrative embodiment 21, wherein the triggering event is raised when a user interacts with a trigger object thereby meeting a pre-determined trigger condition.

[128] 23. The electronic device of any one of illustrative embodiments 18-22, wherein the non-transitory computer readable medium storing computer executable code that when executed by the processor cause the processor to, in response to a second input from at least one user interface device selecting a specific trigger object from a trigger menu, display a trigger condition menu having a list of one or more trigger conditions for the trigger object; and in response to a selection of a first trigger condition for the trigger object, allows a user to

place the trigger object into the virtual reality environment, the trigger condition, when met, causes the processor to update the container of the first object to include a selected action script.

[129] 24. An electronic device comprising:

a processor;

a headset having a visual feedback screen;

one or more user interface devices; and

a non-transitory computer readable medium storing computer executable code that when executed by the processor cause the processor to:

project a virtual reality environment to the headset, the virtual reality environment having one or more object, each of the one or more object being created by a container having properties that define how the one or more object interact within the virtual reality environment; and

in response to an input from at least one user interface device, connect a trigger object selected from a trigger menu to a particular object of the one or more object thereby forming a complex object, the complex object including one or more action script, display, in the virtual reality environment on the visual feedback screen, a scriptable menu having one or more trigger scriptable action for the trigger object; and in response to a selection of a trigger scriptable action for the trigger object, allow a user to place the trigger object into the virtual reality environment and to associate the trigger scriptable action to the one or more action script of the complex object.

[130] 25. The electronic device of illustrative embodiment 24, wherein the trigger object is identified before the input.

[131] 26. The electronic device of illustrative embodiment 24, wherein the trigger object is identified after the input.

[132] 27. The electronic device of any one of illustrative embodiments 24-26, wherein the non-transitory computer readable medium storing computer executable code that when executed by the processor cause the processor to, in response to a trigger event, update the complex object to activate the action script.

[133] 28. The electronic device of illustrative embodiment 27, wherein the trigger event is raised when the user interacts with the trigger object thereby meeting a pre-determined trigger condition.

[134] 29. A non-transitory computer readable medium storing computer executable code that when executed by a processor cause the processor to:

project a virtual reality environment to a headset, the virtual reality environment having one or more object, each of the one or more object being created by a container having properties that define how the one or more object interact within the virtual reality environment, the one or more object having a three dimensional mesh with coordinates defining a boundary of the one or more object;

in response to a first input from at least one user interface device selecting a first object of the one or more object within the virtual reality environment, display an action menu on a visual feedback screen having a list of pre-defined action scripts, the action scripts including a module of computer executable code having one or more function, the one or more function configured to affect one or more property of a first object of the one or more object within the virtual reality environment; and

update the container of the first object to include a selected action script.

[135] 30. The non-transitory computer readable medium of illustrative embodiment 29, wherein the first object is identified before the first input.

[136] 31. The non-transitory computer readable medium of illustrative embodiment 29, wherein the first object is identified after the first input.

[137] 32. The non-transitory computer readable medium of any one of illustrative embodiments 29-31, wherein the non-transitory computer readable medium storing computer executable code that when executed by the processor cause the processor to, in response to a triggering event, update the container of the first object to include a selected action script.

[138] 33. The non-transitory computer readable medium of illustrative embodiment 32, wherein the triggering event is raised when a user interacts with a trigger object thereby meeting a pre-determined trigger condition.

[139] 34. The non-transitory computer readable medium of any one of illustrative embodiments 29-33, wherein the non-transitory computer readable medium storing computer executable code that when executed by the processor cause the processor to, in response to a second input from at least one user interface device selecting a specific trigger object from a trigger menu, display a trigger condition menu having a list of one or more trigger conditions for the trigger object; and in response to a selection of a first trigger condition for

the trigger object, allows a user to place the trigger object into the virtual reality environment, the trigger condition, when met, causes the processor to update the container of the first object to include a selected action script.

[140] From the above description and examples, it is clear that the inventive concepts disclosed and claimed herein are well adapted to attain the advantages mentioned herein. While exemplary embodiments of the inventive concepts have been described for purposes of this disclosure, it will be understood that numerous changes may be made which will readily suggest themselves to those skilled in the art and which are accomplished within the spirit of the inventive concepts disclosed and claimed herein. For exemplary purposes, examples of VR environment of a basketball court and examples of complex objects of a car are used, however, it is to be understood that the examples are for illustrative purposes only and are not to be construed as limiting the scope of the invention.

What is claimed is:

1. An electronic device comprising:

a processor;

a headset having a visual feedback screen;

one or more user interface devices; and

a non-transitory computer readable medium storing computer executable code that when executed by the processor cause the processor to:

project a mixed reality environment to the headset, the mixed reality environment having one or more objects, each of the one or more objects being created by object computer code having properties that define how the one or more objects interact within the mixed reality environment;

in response to a first input from at least one user interface device, display an action menu on the visual feedback screen having a list of pre-defined action modules, the action modules including a module of computer executable code having one or more functions, the one or more functions configured to affect one or more properties of a first object of the one or more objects within the mixed reality environment;

update the object computer code of the first object to include a selected action module.

2. The electronic device of claim 1, wherein the first object is identified before the first input.

3. The electronic device of claim 1, wherein the first object is identified after the first input.

4. The electronic device of any one of claims 1-3, wherein the non-transitory computer readable medium storing computer executable code that when executed by the processor cause the processor to, in response to a triggering event, update the object computer code of the first object to include a selected action module.

5. The electronic device of claim 4, wherein the triggering event is raised when a user interacts with a trigger object thereby meeting a pre-determined trigger condition.

6. The electronic device of any one of claims 1-5, wherein the non-transitory computer readable medium storing computer executable code that when executed by the processor cause the processor to, in response to a second input from at least one user interface device selecting a specific trigger object from a trigger menu, display a trigger condition menu having a list of one or more trigger conditions for the trigger object; and in response to a selection of a first trigger condition for the trigger object, allows a user to place the trigger object into the

mixed reality environment, the trigger condition, when met, causes the processor to update the object computer code of the first object to include a selected action module.

7. An electronic device comprising:

a processor;

a headset having a visual feedback screen;

one or more user interface devices; and

a non-transitory computer readable medium storing computer executable code that when executed by the processor cause the processor to:

project a mixed reality environment to the headset, the mixed reality environment having one or more objects, each of the one or more objects being created by object computer code having properties that define how the one or more objects interact within the mixed reality environment;

in response to a first input from at least one user interface device, selecting a specific trigger object from a trigger menu, display a trigger condition menu having a list of one or more trigger conditions for the trigger object; and in response to a selection of a first trigger condition for the trigger object, allows the user to place the trigger object into the mixed reality environment.

8. The electronic device of claim 7, wherein the trigger object is identified before the first input.

9. The electronic device of claim 7, wherein the trigger object is identified after the first input.

10. The electronic device of any one of claims 7-9, wherein the non-transitory computer readable medium storing computer executable code that when executed by the processor cause the processor to, in response to a triggering event, update the object computer code of the trigger object to include a selected action module.

11. The electronic device of claim 10, wherein the triggering event is raised when a user interacts with the trigger object thereby meeting a pre-determined trigger condition.

12. A non-transitory computer readable medium storing computer executable code that when executed by a processor cause the processor to:

project a mixed reality environment to a headset, the mixed reality environment having one or more objects, each of the one or more objects being created by object computer code having properties that define how the one or more objects interact within the mixed reality environment;

in response to a first input from at least one user interface device, display an action menu on a visual feedback screen having a list of pre-defined action modules, the action modules including a module of computer executable code having one or more functions, the one or more functions configured to affect one or more properties of a first object of the one or more objects within the mixed reality environment;

update the object computer code of the first object to include a selected action module.

13. The non-transitory computer readable medium of claim 12, wherein the first object is identified before the first input.

14. The non-transitory computer readable medium of claim 12, wherein the first object is identified after the first input.

15. The non-transitory computer readable medium of any one of claims 12-14, wherein the non-transitory computer readable medium storing computer executable code that when executed by the processor cause the processor to, in response to a triggering event, update the object computer code of the first object to include a selected action module.

16. The non-transitory computer readable medium of claim 15, wherein the triggering event is raised when the user interacts with a trigger object thereby meeting a pre-determined trigger condition.

17. The non-transitory computer readable medium of any one of claims 12-16, wherein the non-transitory computer readable medium storing computer executable code that when executed by the processor cause the processor to, in response to a second input from at least one user interface device selecting a specific trigger object from a trigger menu, display a trigger condition menu having a list of one or more trigger conditions for the trigger object; and in response to a selection of a first trigger condition for the trigger object, allows the user to place the trigger object into the mixed reality environment, the trigger condition, when met, causes the processor to update the object computer code of the first object to include a selected action module.

18. An electronic device comprising:

a processor;

a headset having a visual feedback screen;

one or more user interface devices; and

a non-transitory computer readable medium storing computer executable code that when executed by the processor cause the processor to:

project a virtual reality environment to the headset, the virtual reality environment having one or more object, each of the one or more object being created by a container having properties that define how the one or more object interact within the virtual reality environment, the one or more object having a three dimensional mesh with coordinates defining a boundary of the one or more object;

selecting a first object of the one or more object;

in response to a first input from at least one user interface device, display an action menu within the virtual reality environment on the visual feedback screen, the action menu having a list of pre-defined action scripts, the action scripts including a module of computer executable code having one or more function, the one or more function configured to affect one or more property of the first object of the one or more object within the virtual reality environment; and

update the container of the first object to include a selected action script.

19. The electronic device of claim 18, wherein the first object is identified before the first input.

20. The electronic device of claim 18, wherein the first object is identified after the first input.

21. The electronic device of any one of claims 18-20, wherein the non-transitory computer readable medium storing computer executable code that when executed by the processor cause the processor to, in response to a triggering event, update the container of the first object to include a selected action script.

22. The electronic device of claim 21, wherein the triggering event is raised when a user interacts with a trigger object thereby meeting a pre-determined trigger condition.

23. The electronic device of any one of claims 18-22, wherein the non-transitory computer readable medium storing computer executable code that when executed by the processor cause the processor to, in response to a second input from at least one user interface device selecting a specific trigger object from a trigger menu, display a trigger condition menu having a list of one or more trigger conditions for the trigger object; and in response to a selection of a first trigger condition for the trigger object, allows a user to place the trigger object into the virtual reality environment, the trigger condition, when met, causes the processor to update the container of the first object to include a selected action script.

24. An electronic device comprising:

a processor;

a headset having a visual feedback screen;

one or more user interface devices; and

a non-transitory computer readable medium storing computer executable code that when executed by the processor cause the processor to:

project a virtual reality environment to the headset, the virtual reality environment having one or more object, each of the one or more object being created by a container having properties that define how the one or more object interact within the virtual reality environment; and

in response to an input from at least one user interface device, connect a trigger object selected from a trigger menu to a particular object of the one or more object thereby forming a complex object, the complex object including one or more action script, display, in the virtual reality environment on the visual feedback screen, a scriptable menu having one or more trigger scriptable action for the trigger object; and in response to a selection of a trigger scriptable action for the trigger object, allow a user to place the trigger object into the virtual reality environment and to associate the trigger scriptable action to the one or more action script of the complex object.

25. The electronic device of claim 24, wherein the trigger object is identified before the input.

26. The electronic device of claim 24, wherein the trigger object is identified after the input.

27. The electronic device of any one of claims 24-26, wherein the non-transitory computer readable medium storing computer executable code that when executed by the processor cause the processor to, in response to a trigger event, update the complex object to activate the action script.

28. The electronic device of claim 27, wherein the trigger event is raised when the user interacts with the trigger object thereby meeting a pre-determined trigger condition.

29. A non-transitory computer readable medium storing computer executable code that when executed by a processor cause the processor to:

project a virtual reality environment to a headset, the virtual reality environment having one or more object, each of the one or more object being created by a container having properties that define how the one or more object interact within the virtual reality

environment, the one or more object having a three dimensional mesh with coordinates defining a boundary of the one or more object;

in response to a first input from at least one user interface device selecting a first object of the one or more object within the virtual reality environment, display an action menu on a visual feedback screen having a list of pre-defined action scripts, the action scripts including a module of computer executable code having one or more function, the one or more function configured to affect one or more property of a first object of the one or more object within the virtual reality environment; and

update the container of the first object to include a selected action script.

30. The non-transitory computer readable medium of claim 29, wherein the first object is identified before the first input.

31. The non-transitory computer readable medium of claim 29, wherein the first object is identified after the first input.

32. The non-transitory computer readable medium of any one of claims 29-31, wherein the non-transitory computer readable medium storing computer executable code that when executed by the processor cause the processor to, in response to a triggering event, update the container of the first object to include a selected action script.

33. The non-transitory computer readable medium of claim 32, wherein the triggering event is raised when a user interacts with a trigger object thereby meeting a pre-determined trigger condition.

34. The non-transitory computer readable medium of any one of claims 29-33, wherein the non-transitory computer readable medium storing computer executable code that when executed by the processor cause the processor to, in response to a second input from at least one user interface device selecting a specific trigger object from a trigger menu, display a trigger condition menu having a list of one or more trigger conditions for the trigger object; and in response to a selection of a first trigger condition for the trigger object, allows a user to place the trigger object into the virtual reality environment, the trigger condition, when met, causes the processor to update the container of the first object to include a selected action script.

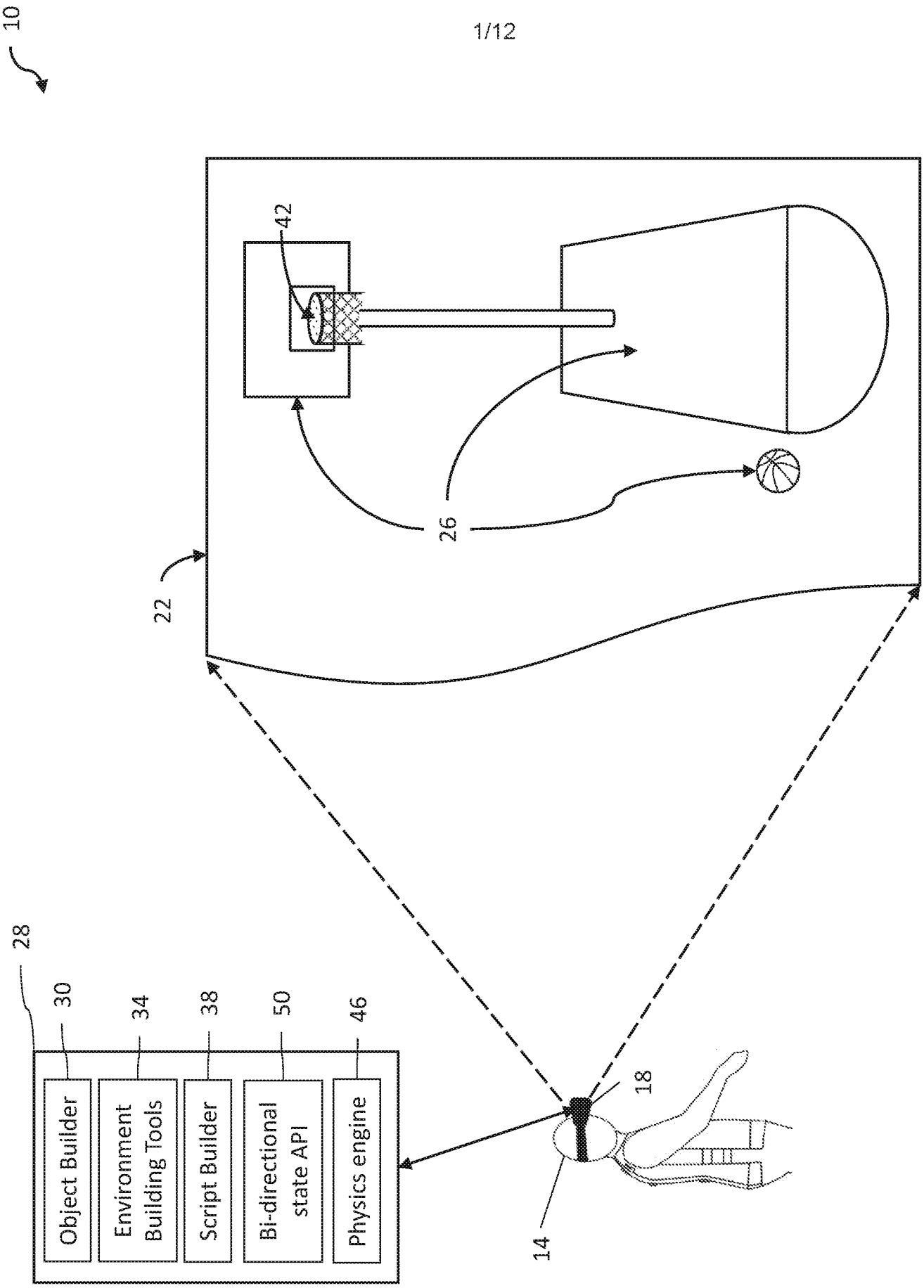

10
22
42
26
28
30
34
38
50
46
Object Builder
Environment Building Tools
Script Builder
Bi-directional state API
Physics engine
18
14

FIG. 1

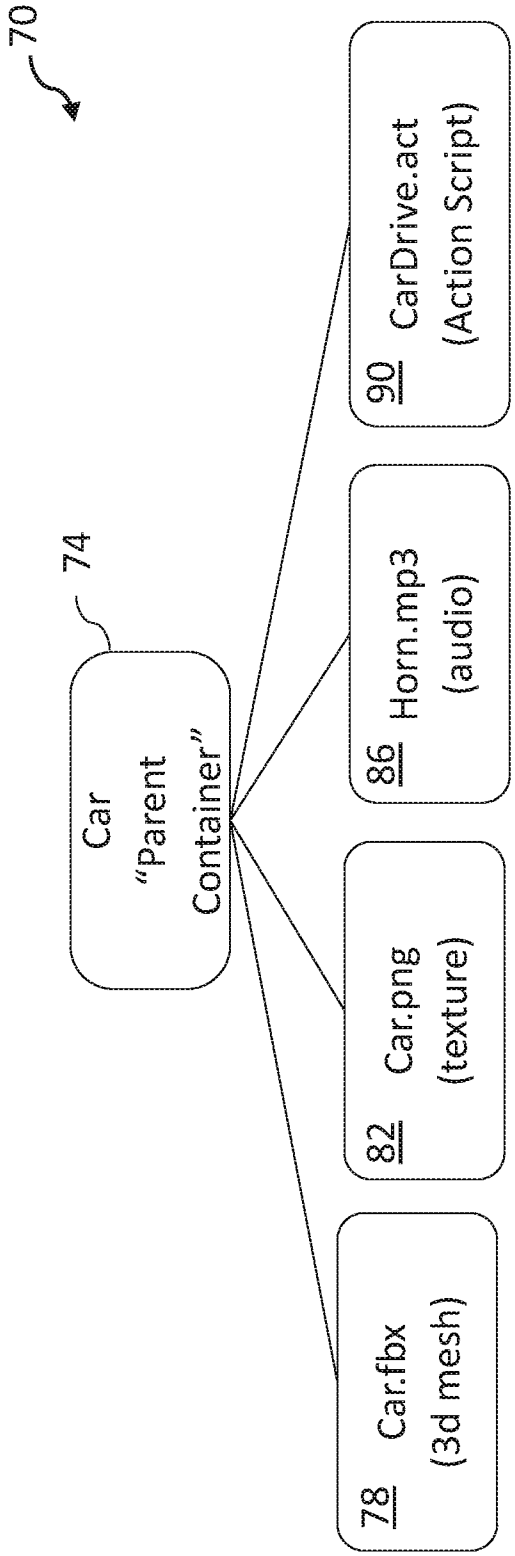

70
74
Car
"Parent
Container"
78 Car.fbx
(3d mesh)
82 Car.png
(texture)
86 Horn.mp3
(audio)
90 CarDrive.act
(Action Script)

FIG. 2

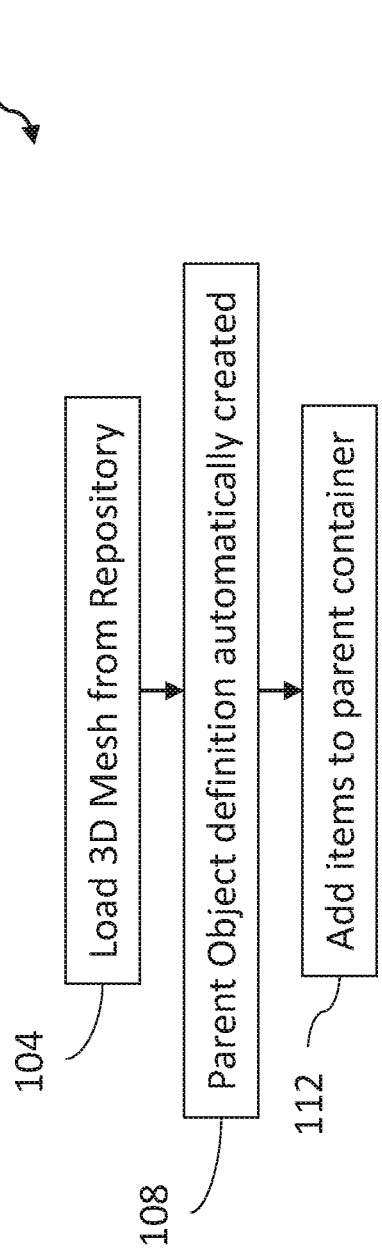

100
104
Load 3D Mesh from Repository
108
Parent Object definition automatically created
112
Add items to parent container

FIG. 3

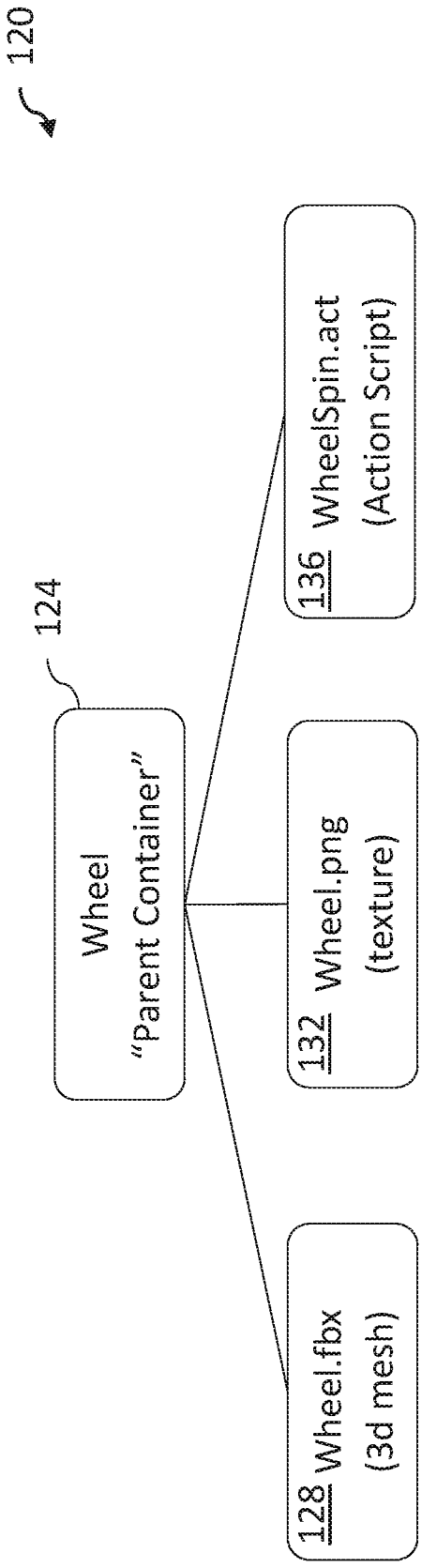

120
124
Wheel
"Parent Container"
128 Wheel.fbx
(3d mesh)
132 Wheel.png
(texture)
136 WheelSpin.act
(Action Script)

FIG. 4

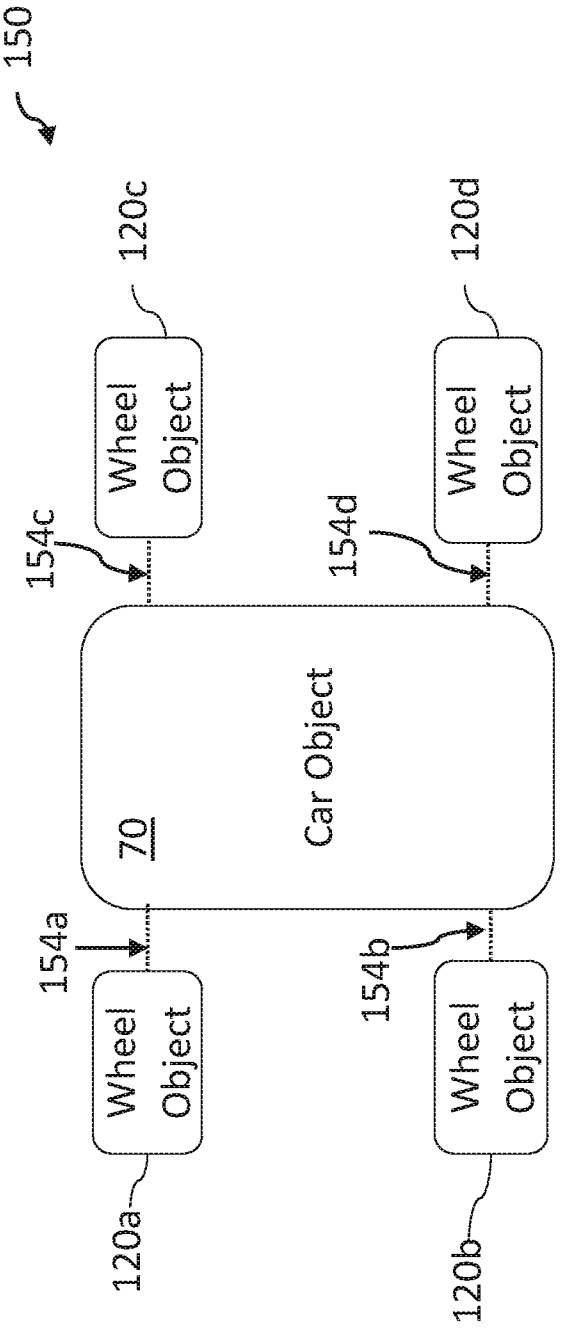

150
Car Object
70
154a
154b
154c
154d
Wheel Object
120a
120b
120c
120d

FIG. 5

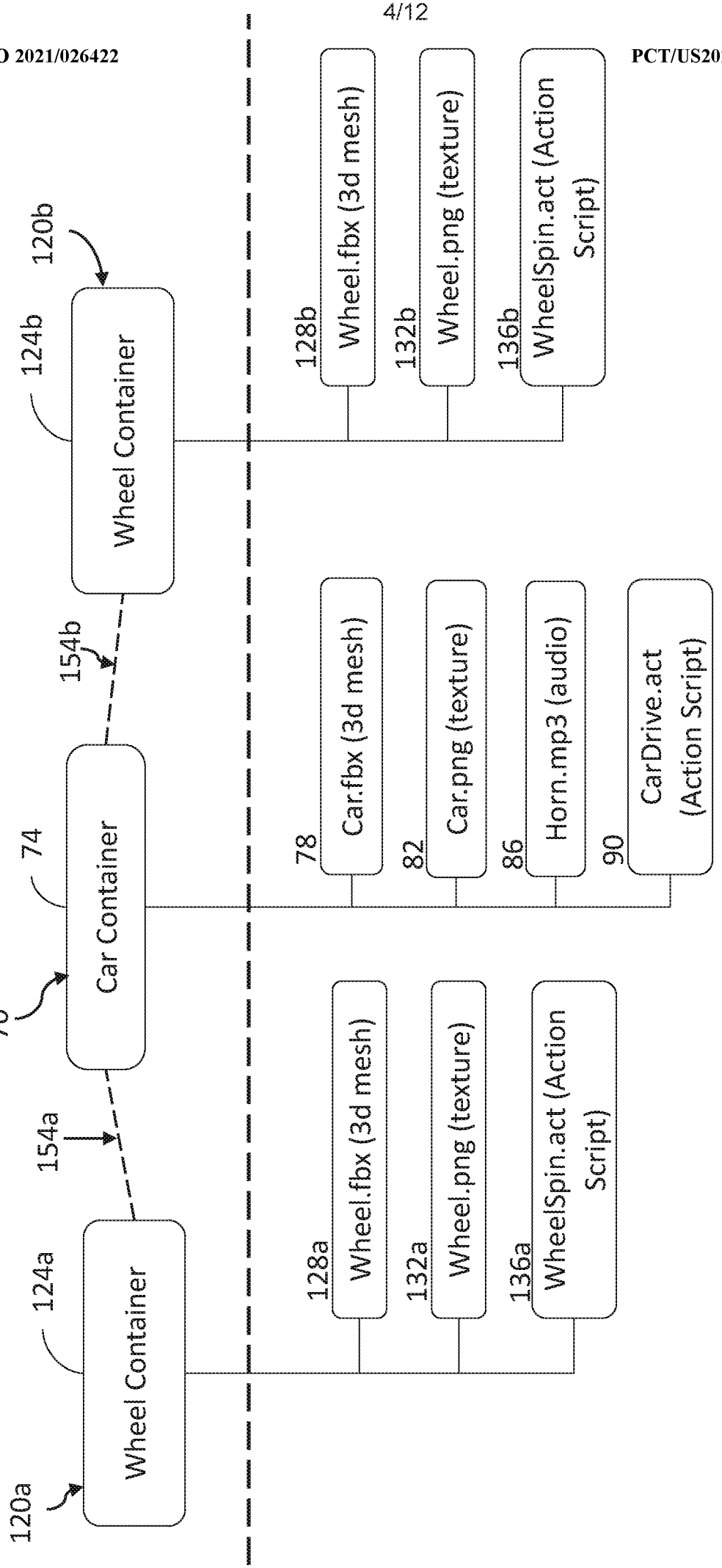

158
120a
Wheel Container
124a
128a
Wheel.fbx (3d mesh)
132a
Wheel.png (texture)
136a
WheelSpin.act (Action Script)
154a
70
Car Container
74
78
Car.fbx (3d mesh)
82
Car.png (texture)
86
Horn.mp3 (audio)
90
CarDrive.act (Action Script)
154b
120b
Wheel Container
124b
128b
Wheel.fbx (3d mesh)
132b
Wheel.png (texture)
136b
WheelSpin.act (Action Script)

FIG. 6

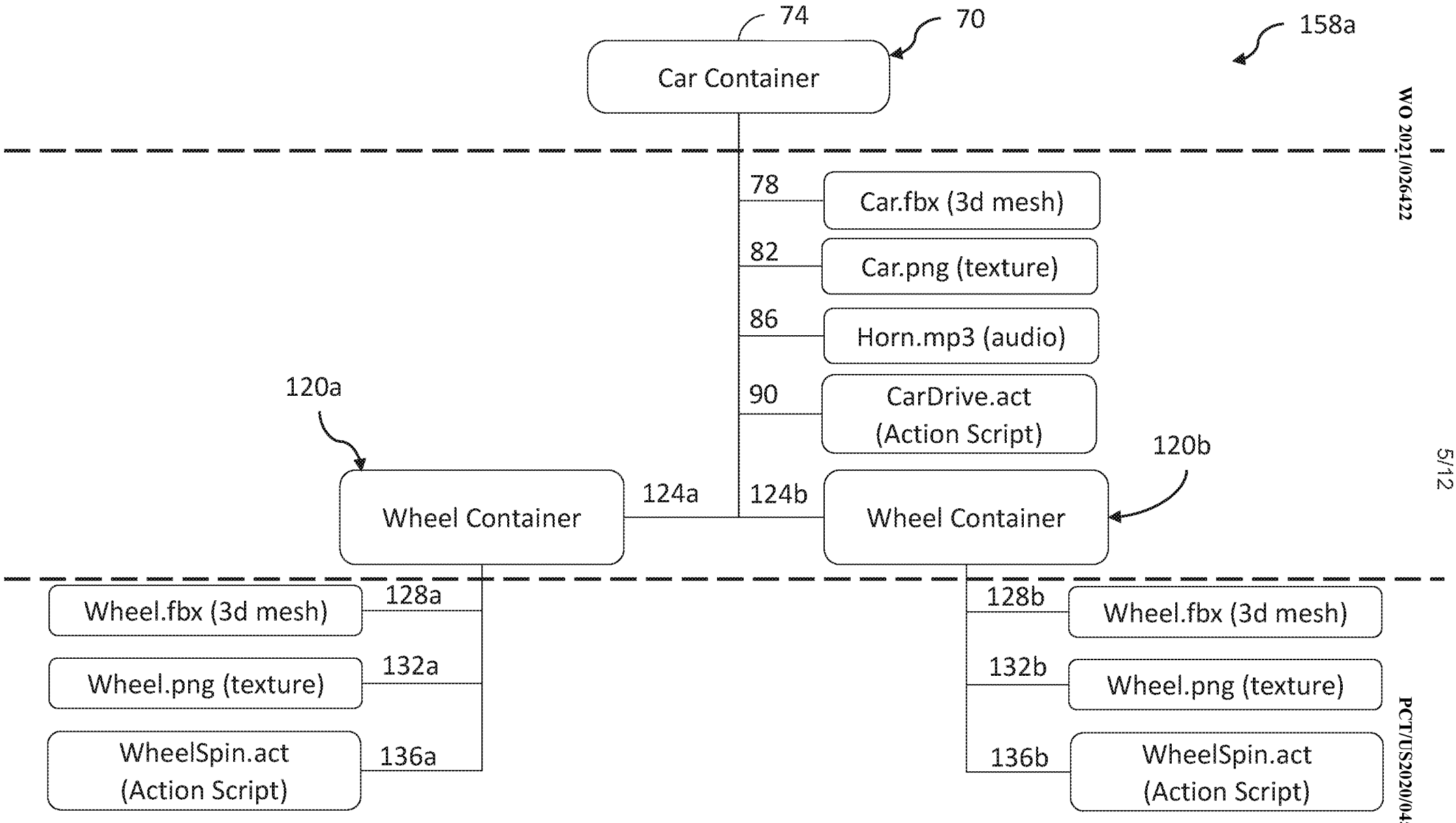

74
70
158a
Car Container
78
Car.fbx (3d mesh)
82
Car.png (texture)
86
Horn.mp3 (audio)
90
CarDrive.act
(Action Script)
120a
120b
Wheel Container
124a
124b
Wheel Container
128a
Wheel.fbx (3d mesh)
132a
Wheel.png (texture)
136a
WheelSpin.act
(Action Script)
128b
Wheel.fbx (3d mesh)
132b
Wheel.png (texture)
136b
WheelSpin.act
(Action Script)

FIG. 7

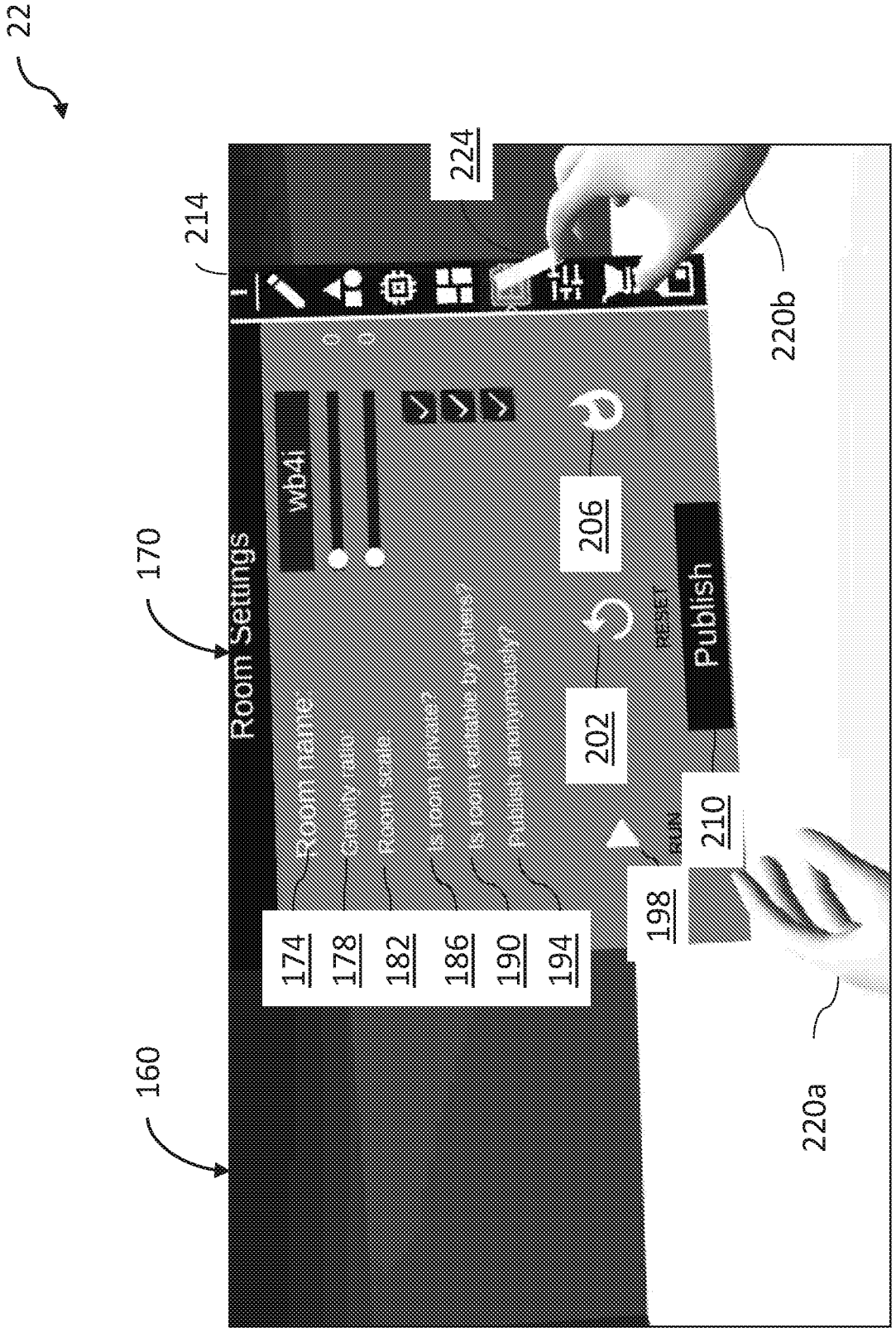

22
214
224
170
160
220b
220a
Room Settings
wb4j
Publish
RESET
174
178
182
186
190
194
198
202
206
210

FIG. 8

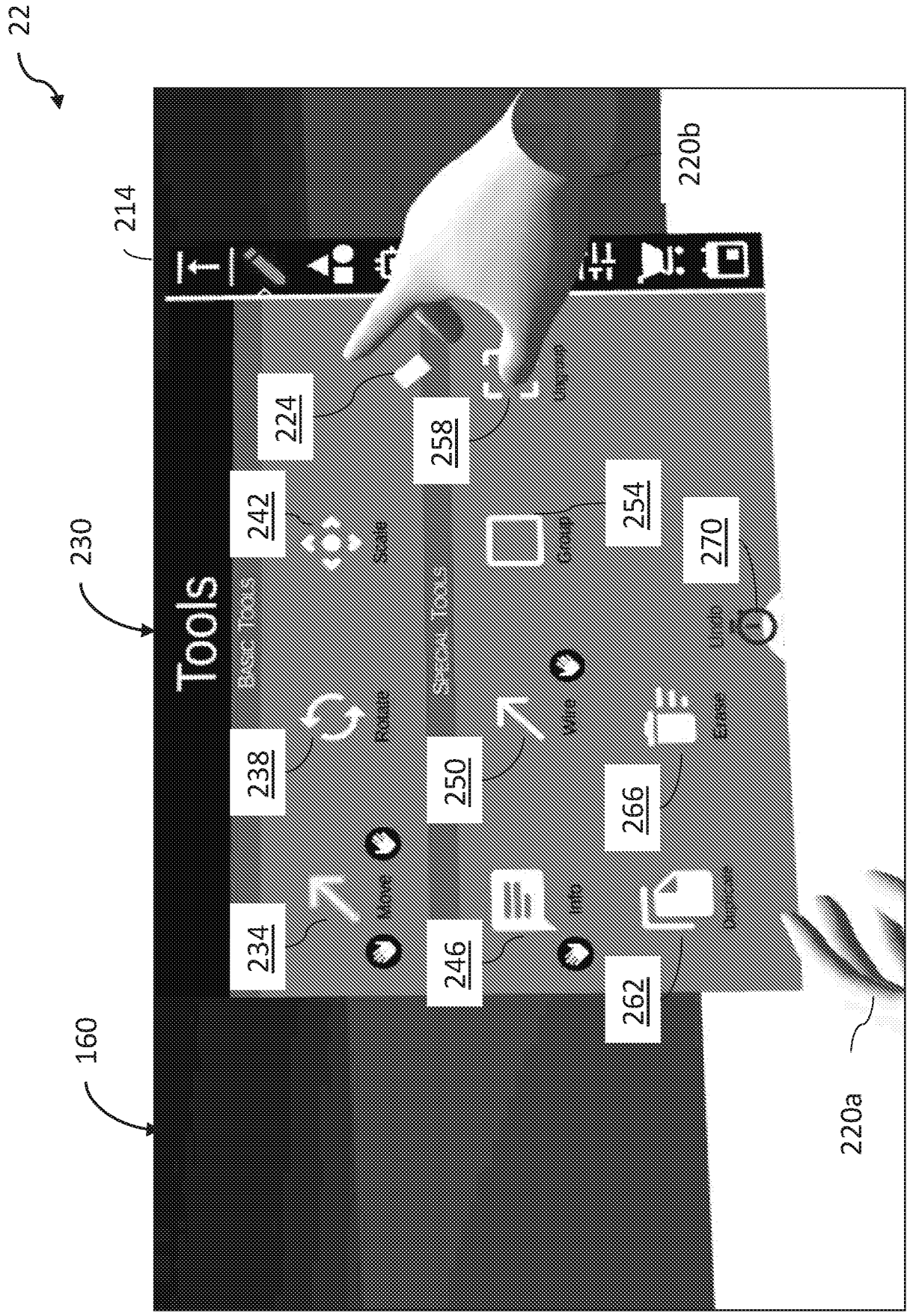

22
214
230
160
220b
220a
Tools
234
238
242
224
258
254
270
246
250
266
262
Move
Rotate
Scale
Create
Wire
Edit
Duplicate
Explode
Delete

FIG. 9

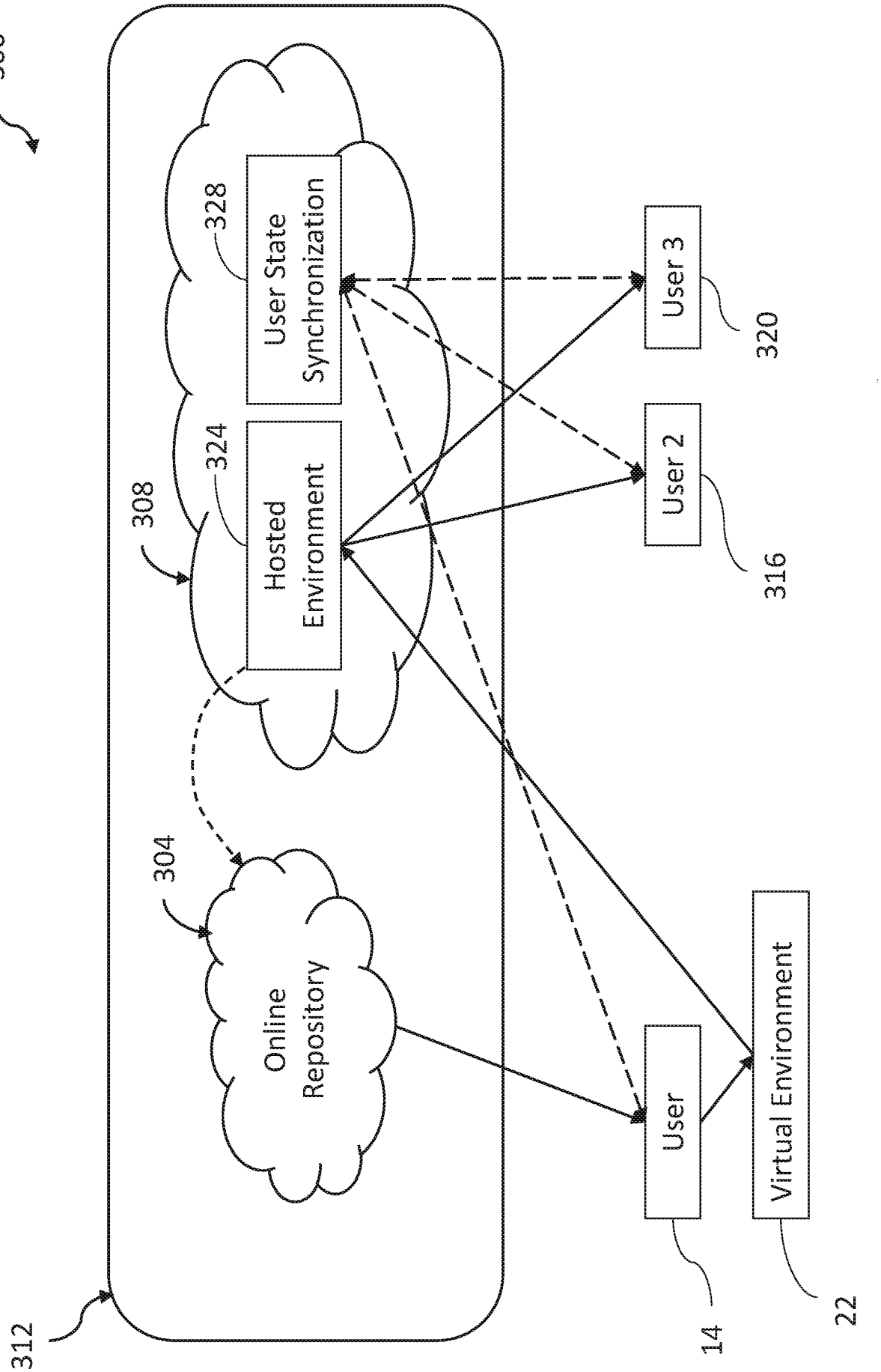

300
312
308
328
User State Synchronization
324
Hosted Environment
304
Online Repository
User 3
320
User 2
316
User
14
Virtual Environment
22

FIG. 10

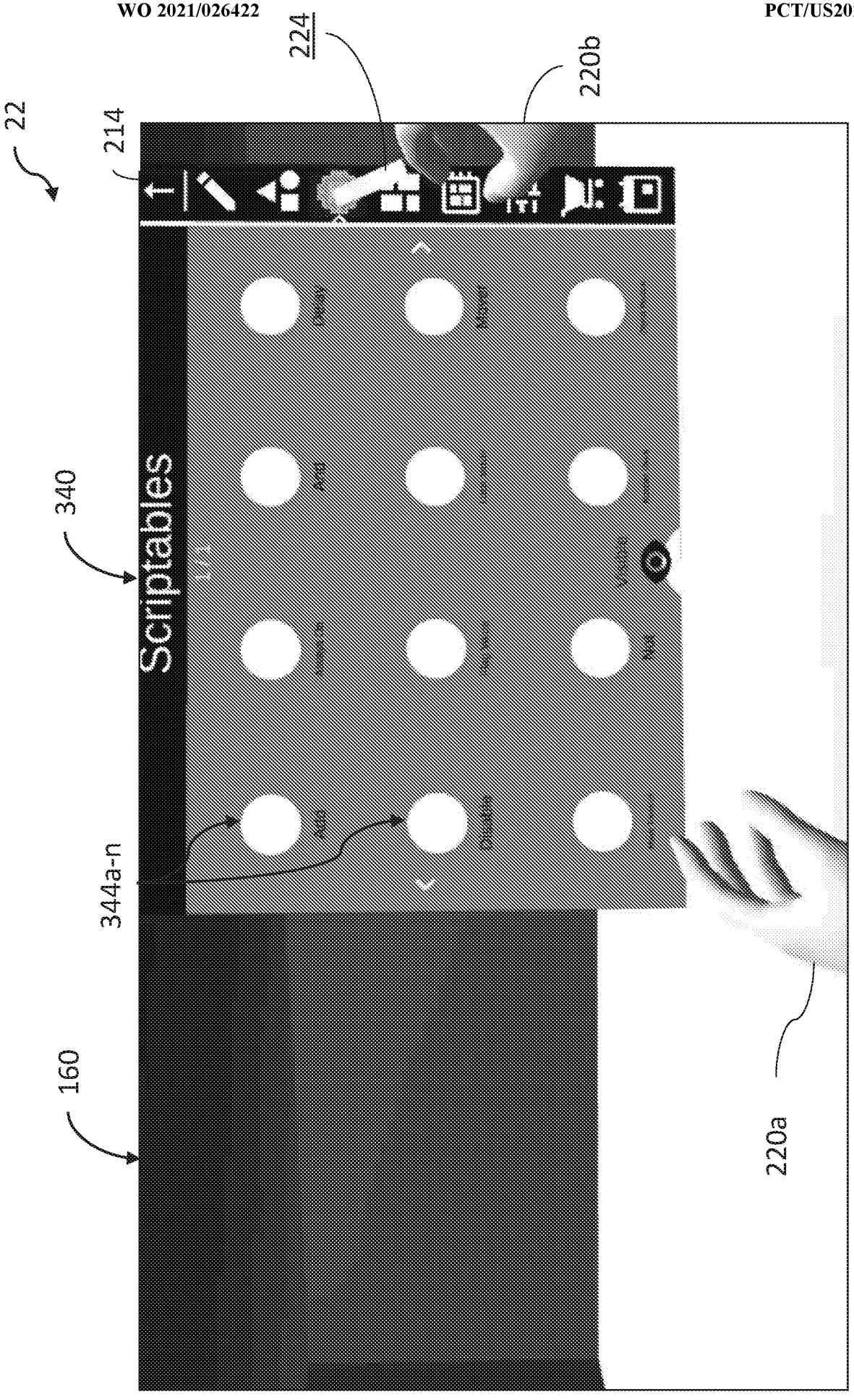


FIG. 11

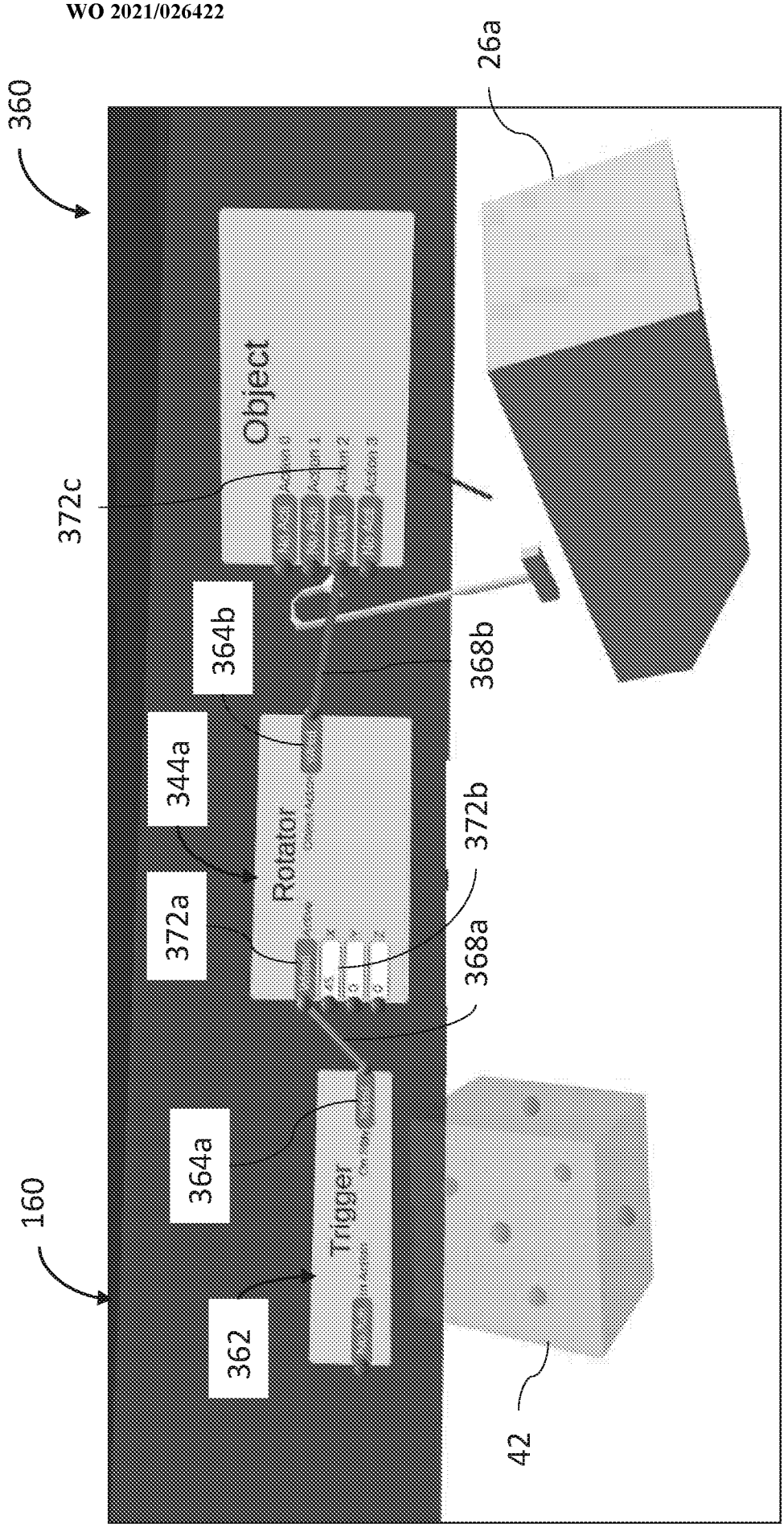

360
26a
372c
Object
364b
344a
Rotator
372b
372a
368a
368b
364a
Trigger
160
362
42

FIG. 12

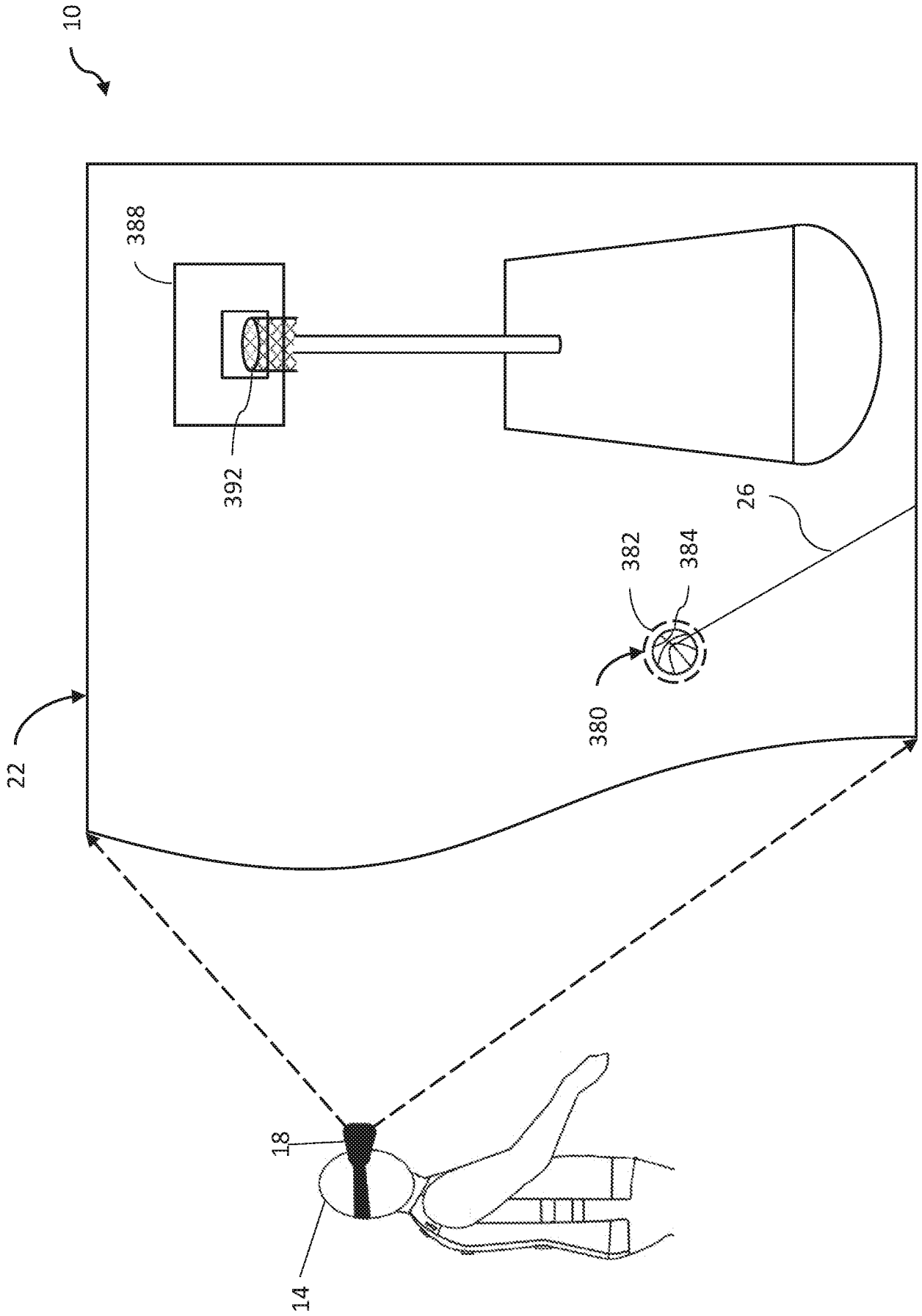


FIG. 13

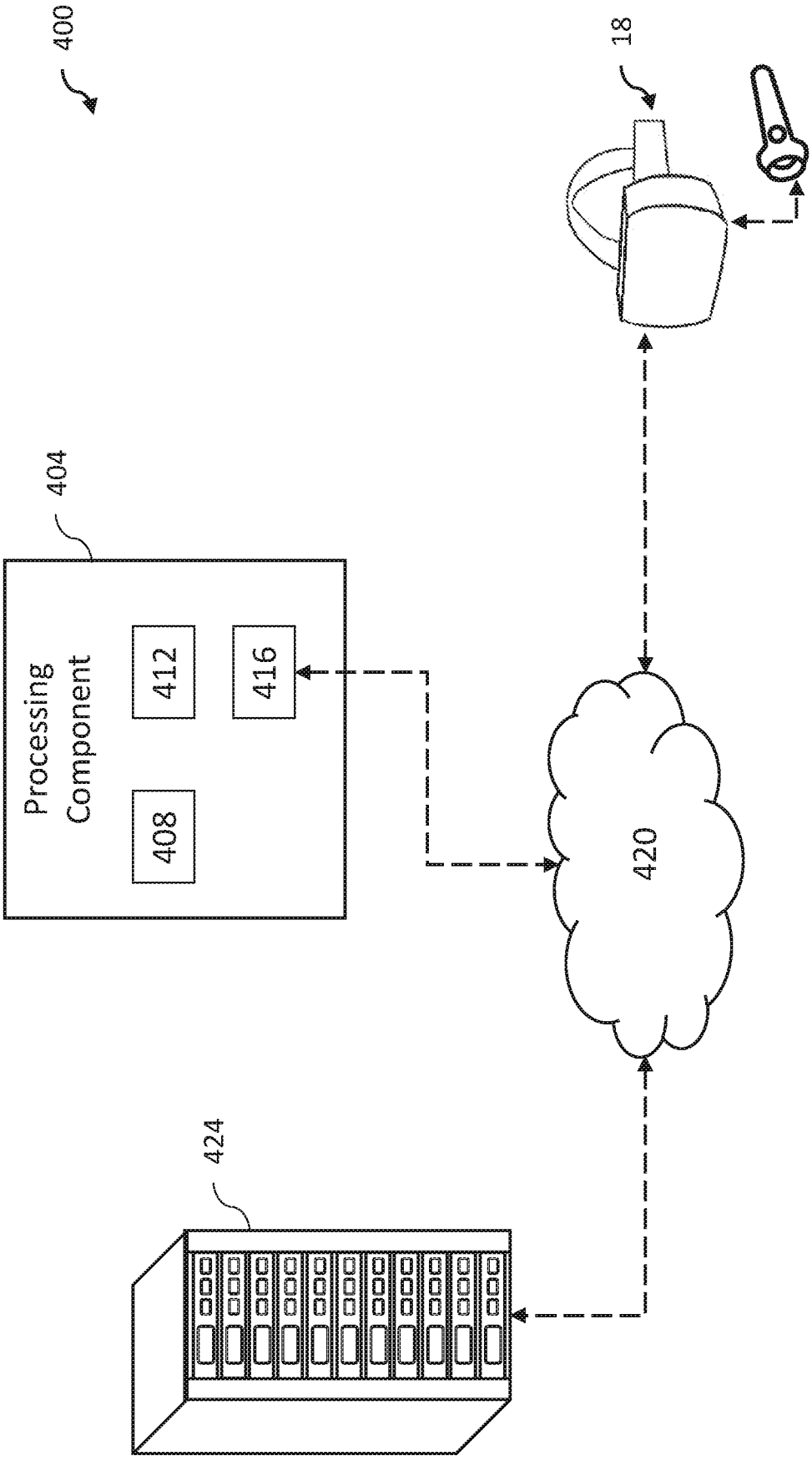


Fig. 14

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US 20/45344

Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. ☐ Claims Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:

3. ☒ Claims Nos.: 6, 17, 23, 34
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

1. ☐ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.

2. ☐ As all searchable claims could be searched without effort justifying additional fees, this Authority did not invite payment of additional fees.

3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:

4. ☐ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest

☐ The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.

☐ The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.

☐ No protest accompanied the payment of additional search fees.

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US 20/45344

A. CLASSIFICATION OF SUBJECT MATTER

IPC - G09G 5/00 (2020.01)

CPC - G02B 27/017, G02B 27/0172, G02B 2027/0132, G02B 2027/0187, G02B 27/0176, G09G 3/003, H04N 13/0497, H04N 13/0285, H04N 13/044, H04N 13/0059, G06T 19/006, G06T 19/00, G06F 3/011, G02B 27/017, G06F 3/012

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)
See Search History document

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
See Search History document

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
See Search History document

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2019/0004791 A1 (UMAJIN INC.), 03 January 2019 (03.01.2019), entire document, especially Abstract; para [0015], [0018], [0074], [0080], [0105], [0122], [0133], [0221], [0336], [0378]-[0379], [0401], [0434], [0445], [0539]	1-5, 7-16, 18-22, 24-33

☐ 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
"D" document cited by the applicant in the international application
"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	Date of mailing of the international search report
11 October 2020 (11.10.2020)	06 NOV 2020
Name and mailing address of the ISA/US Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-8300	Authorized officer Lee Young Telephone No. PCT Helpdesk: 571-272-4300

Form PCT/ISA/210 (second sheet) (July 2019)