United States Provisional Patent Application

for

DYNAMIC USER INTERFACE WITH ORBITING ELEMENTS AND RADIAL BRANCHES

Inventor:

Michael PATON

DYNAMIC USER INTERFACE WITH ORBITING ELEMENTS AND RADIAL BRANCHES

FIELD OF THE INVENTION

The present disclosure relates generally to the field of graphical user interfaces.

BACKGROUND

Displaying and interacting with conventional graphical user interfaces has become bland and unoriginal, with information being typically displayed in a static grid, and allowing limited maneuverability in either a vertical or horizontal plane. This type of user interface also tends to create information-overload, with too many icons on the screen, burdening the experience of users. This can also lead to difficulty in focusing on important information on the screen.

What is needed is an improved user interface which provides a more lively and interactive user experience.

SUMMARY

The present disclosure relates generally to the field of graphical user interfaces.

In one aspect, there is disclosed a dynamic user interface comprising a central point of reference or hub, around which a plurality of orbital interface elements orbit the hub in a clockwise or counterclockwise direction.

In an embodiment, the rotation of the orbital interface elements is dynamically animated, such that the user interfaces creates a sense of the orbital interface elements rotating around the hub in real time.

In an embodiment, the orbital interface elements are counterbalanced such that they are continually oriented in an upright position as they orbit the hub.

In an embodiment, a user can control the orbital interface elements with a user input, such as a user swiping a finger across a touchscreen in a given direction which causes the orbital interface elements to spin in the direction of the swipe. The direction of the swipe determines whether the orbital interface elements spin in a clockwise direction or a counterclockwise direction. The speed and/or force of the swipe across the touchscreen is sensed to determine the initial speed of the spin, and how long the orbital interface elements continue to spin before slowing to a stop. A damping control may be applied to bring the spinning orbital interface elements to a stop more quickly, or to allow them to spin longer before coming to a stop.

In an embodiment, the hub includes a scroll ring which is formed in an arc stretching around at least a segment of the hub. The scroll ring is adapted to receive a touchscreen user input which swipes a virtual knob around the scroll ring, in either a clockwise or a counterclockwise direction.

In another embodiment, the central point of reference or hub is positioned centrally with a plurality of spokes or paths which radiate outwardly from the hub. Each of the spokes or paths ends with a node which can receive an orbital interface element swiped towards it.

In another embodiment, the node may be adapted to perform a programmed function on any received orbital interface element, or to guide the user to another layer of the user interface.

In another embodiment, multiple orbital interface elements may be rotated and acted upon at the same time, with a common action or different action taken on each of the orbital interface elements, depending on the spoke on which they are positioned, and the programmed function of the outer node to which they are sent.

In another embodiment, the dynamic user interface has multiple concentric orbital paths around the hub, with each orbital path providing a path around which a plurality of orbital interface elements may spin around the hub.

In another aspect, the user interface comprises a central point of reference or hub with a plurality of branches extending outwardly in all directions.

In an embodiment, one or more of the branches may include secondary branches or tertiary branches along which branches a user may navigate to arrive at an intermediate node or end node.

In an embodiment, the one or more branches may branch out omnidirectionally, such that a user can navigate in any direction along the branches in a three-dimensional space. Navigation along these branches may be animated in the graphical user interface to give a sense of movement along the branches to focus on a particular portion of the network at any one time.

In another embodiment, “leaf” nodes can grow from the one or more branches, such that they terminate at a particular path. These leaf nodes may be temporary, and may disappear or fall off when they no longer serve a purpose in the network. Thus, the navigational paths can grow or contract over time, as required.

In another embodiment, roots can grow below a hub or a node, such that a user can navigate to a lower level. These roots can also grow and contract like branches, and effectively allow a user to navigate to any depth in any direction as may be required. Similar to branches, the roots may end in nodes that terminate a particular path along a root, and these roots may change dynamically over time.

Advantageously, these alternative graphical user interface designs create a more lively and dynamic user interface experience which focusses the user’s attention, and allows fluid navigation through multiple layers of user interfaces providing virtually endless navigation paths.

In this respect, before explaining at least one embodiment of the invention in detail, it is to be understood that the invention is not limited in its application to the details of construction and to embodiments set forth in the following description or the examples provided therein, or illustrated in the drawings. Therefore, it will be appreciated that a number of variants and modifications can be made without departing from the teachings of the disclosure as a whole. Therefore, the present system, method and apparatus is capable of other embodiments and of being practiced and carried out in various ways. Also, it is to be understood that the phraseology and terminology employed herein are for the purpose of description and should not be regarded as limiting.

BRIEF DESCRIPTION OF THE DRAWINGS

The present system and method will be better understood, and objects of the invention will become apparent, when consideration is given to the following detailed description thereof. Such description makes reference to the annexed drawings, wherein:

FIG. 1 shows a schematic diagram of a dynamic user interface in accordance with an illustrative embodiment.

FIGS. 2 – 4 show the dynamic user interface of FIG. 1 in which a plurality of orbital interface elements have progressive rotated clockwise (FIG. 1 > FIG. 2 > FIG. 3 > FIG. 4 > FIG. 1) or counterclockwise (FIG. 1 > FIG. 4 > FIG. 3 > FIG. 2 > FIG. 1).

FIGS. 5a – FIG. 8b illustrate another embodiment of the dynamic user interface. showing a clockwise rotation (FIG. 5a > FIG/ 6a > FIG. 7a > FIG. 8a > FIG. 5a) or a counterclockwise rotation (FIG. 5a > FIG. 8a > FIG. 7a > FIG. 6a > FIG. 5a).

FIG. 9 shows a schematic diagram of another embodiment of the dynamic user interface having multiple concentric orbital rings or paths.

FIGS. 1a – 13b show the dynamic user interface of FIG. 9 illustrating how orbital interface elements may be moved between a “higher” orbit and “lower” orbit relative to the hub.

FIGS. 14 – 18 show schematic diagrams of another embodiment of a dynamic user interface in which orbital interface elements may be rotated over a spoke or branch to travel to an outer spoke node, or towards the central hub.

FIGS. 19 – 22 show a dynamic user interface in accordance with another embodiment in which branches extend outwardly from a central node.

FIG. 23 shows a schematic block diagram of a generic computing device which may embody various embodiments of the dynamic user interface.

In the drawings, embodiments are illustrated by way of example. It is to be expressly understood that the description and drawings are only for the purpose of illustration and as an aid to understanding, and are not intended as describing the accurate performance and behavior of the embodiments and a definition of the limits of the invention.

DETAILED DESCRIPTION

As noted above, the present disclosure relates generally to the field of user interfaces.

As used herein, the term "user interface" means any device/method for rendering and/or presenting information to a user and/or requesting information from the user. A graphic user interface can include one or more elements such as, for example, a window, title bar, panel, sheet, tab, drawer, matrix, table, form, calendar, outline view, frame, dialog box, static text, text box, list, pick list, pop-up list, pull-down list, menu, tool bar, dock, check box, radio button, hyperlink, browser, video, audio, image, icon, button, control, dial, slider, scroll bar, cursor, status bar, stepper, and/or progress indicator, etc. An audio user interface can include a volume control, pitch control, speed control, voice selector, etc. A video or image interface can include similar controls to that of the aforementioned audio user interface, save, as pertains to the content type (such as filters that change the look/operation of an image/video (such as contrast, saturation, curves, etc.) and, when applicable: playback speed, closed caption, volume, resolution, screen size, and general operation controls (i.e. play, pause), etc. A holographic or volumetric user interface, or virtual reality interface, or augmented reality interface, or mixed reality user interface refers to any device or method used to render/display augmented or virtual reality elements (such as holograms through a device such as a Microsoft HoloLens), or through an entire-field-of-view-encapsulated headset, such as the Oculus Rift, (or a future product to accomplish the same or reasonably-deductible similar ability, such as smart contact lens(es) used for similar or differing function, or a brain chip that alters the information perceived by the brains' visual cortex, a wearable or implanted device or "smart tattoo"), in which any or all of which these interfaces display any form of the aforementioned user interface, as outlined above, or reasonably-deduced-similarities (these interfaces are advanced and many still in their infancy, and may or may not have haptic/sensational feedback, include 3d models, spatial mapping of the proximity around them (as in the HoloLens), change the view of the world around them, or themselves be a piece of software or game utilizing such an interface. As used herein, “interface elements” (as well “user interface” also) means any of the embodiments described (and, if applicable, the respective combinations thereof) of the aforementioned “user interface,” “graphic user interface,” “video or image interface,” “audio user interface,” “holographic user interface,” “volumetric user interface”, “mixed reality user interface,” or any reasonably-deduced-to-be-similar form(s) of interface(s).

It will be appreciated that for simplicity and clarity of illustration, where considered appropriate, reference numerals may be repeated among the figures to indicate corresponding or analogous elements or steps. In addition, numerous specific details are set forth in order to provide a thorough understanding of the embodiments described herein. However, it will be understood by those of ordinary skill in the art that the embodiments described herein may be practiced without these specific details. In other instances, well-known methods, procedures and components have not been described in detail so as not to obscure the embodiments described herein. Furthermore, this description is not to be considered as limiting the scope of the embodiments described herein in any way, but rather as merely describing the implementation of the various embodiments described herein.

As stated above, there are many problems which plague user experience while viewing/navigating a user interface. The present disclosure describes various embodiments of a dynamic user interface which address the aforementioned problems.

In one aspect, there is disclosed a dynamic user interface comprising a central point of reference or hub, around which a plurality of orbital interface elements orbit the hub in a clockwise or counterclockwise direction.

In an embodiment, the rotation of the orbital interface elements is dynamically animated, such that the user interfaces creates a sense of the orbital interface elements rotating around the hub in real time.

In an embodiment, the orbital interface elements are counterbalanced such that they are continually oriented in an upright position as they orbit the hub.

In an embodiment, a user can control the orbital interface elements with a user input, such as a user swiping a finger across a touchscreen in a given direction which causes the orbital interface elements to spin in the direction of the swipe. The direction of the swipe determines whether the orbital interface elements spin in a clockwise direction or a counterclockwise direction. The speed and/or force of the swipe across the touchscreen is sensed to determine the initial speed of the spin, and how long the orbital interface elements continue to spin before slowing to a stop. A damping control may be applied to bring the spinning orbital interface elements to a stop more quickly, or to allow them to spin longer before coming to a stop.

In an embodiment, the hub includes a scroll ring which is formed in an arc stretching around at least a segment of the hub. The scroll ring is adapted to receive a user input (e.g. touchscreen, touchpad, mouse, pointer, etc.) which swipes a virtual knob around the scroll ring, in either a clockwise or a counterclockwise direction.

In another embodiment, the central point of reference or hub is positioned centrally with a plurality of spokes or paths which radiate outwardly from the hub. Each of the spokes or paths ends with a node which can receive an orbital interface element swiped towards it. In an embodiment, the node may be adapted to perform a programmed function on any received orbital interface element, or to guide the user to another layer of the user interface.

In another embodiment, multiple orbital interface elements may be rotated and acted upon at the same time, with a common action or different action taken on each of the orbital interface elements, depending on the spoke on which they are positioned, and the programmed function of the outer node to which they are sent.

In another embodiment, the dynamic user interface has multiple concentric orbital paths around the hub, with each orbital path providing a path around which a plurality of orbital interface elements may spin around the hub.

In another aspect, the user interface comprises a central point of reference or hub with a plurality of branches extending outwardly in all directions.

In an embodiment, one or more of the branches may include secondary branches or tertiary branches along which branches a user may navigate to arrive at an intermediate node or end node.

In an embodiment, the one or more branches may branch out omnidirectionally, such that a user can navigate in any direction along the branches in a three-dimensional space. Navigation along these branches may be animated in the graphical user interface to give a sense of movement along the branches to focus on a particular portion of the network at any one time.

In another embodiment, “leaf” nodes can grow from the one or more branches, such that they terminate at a particular path. These leaf nodes may be temporary, and may disappear or fall off when they no longer serve a purpose in the network. Thus, the navigational paths can grow or contract over time, as required.

In another embodiment, roots can grow below a hub or a node, such that a user can navigate to a lower level. These roots can also grow and contract like branches, and effectively allow a user to navigate to any depth in any direction as may be required. Similar to branches, the roots may end in nodes that terminate a particular path along a root, and these roots may change dynamically over time.

Advantageously, by arranging interface elements around a visible/invisible central point of reference or hub, and/or along visible/invisible branch(es) (e.g. vines, tree-roots/branches, chain(s), spider web, neuron-synaptic connection(s), sunrays, galaxy/star-system(s), path(s)), an endless-interface becomes possible – and opens up numerous navigation and display techniques.

The shape of the orbit trajectory and distance between interface elements may be varied. For example, in one embodiment, the orbit may be a perfect circle with even spacing between the orbital interface elements. However, the orbit may take a different path, such as an elliptical path, or another type of curve such as a Bezier curve, for example. Furthermore, rather than a smooth curve, the orbital interface elements may also follow a path defined by a polygon (e.g. hexagon, octagon, etc.) or even an irregular path around a central hub. Thus, the path may be any shape, open or closed, which could be used as the trajectory followed by the orbital interface elements.

As additional examples, the paths may “yaw”, or a zigzag in a geometric back-and-forth motion such as a circle that has waves going inwards and outwards, as if being gravitationally pulled closer/farther away to the central point of reference (all of these either open or closed paths. This trajectory, be it a normal and/or custom shape and/or open/closed path can be either static or dynamic.

In an Augmented or Virtual Reality situation, the orbital interface elements may “follow” the user when the user’s eyes move in a different direction (e.g. through Microsoft Hololens built-in gaze-detection).

During user interaction with the interface through an input method (i.e. mouse, touch, gesture, sensor, device, thought, etc.), this "automatic and/or interaction-triggered orbiting" will be replaced with one or more parts of the following:

1.) Clockwise/counter-clockwise orbit and/or movement of parent container/interface elements (with or without a basis derived from the inertia of the input method), with parent container setting 1 variable for its angle and another for its angle’s inverse, and applying this inverse variable to child container(s)/interface elements (and/or setting their own variable(s) for their (X, Y) coordinates at any given time, (for usage with Embodiment II.) and/or additional Embodiment(s)).

2.) One or more interface elements hide and/or reveal themselves, effectively displaying new/old information (e.g. more/less orbits and/or interface-elements). In essence, this is a form of scrolling in both vertical and horizontal planes simultaneously. This could be like Embodiment II.) which reads the variable set from the parent container and upon requisite angle, changes the property of its (X, Y) coordinates to the middle behind **7** (either through a timed animation or instantaneously) while a new child-container appears beneath **7** at middle coordinates and makes its way to new (X, Y) coordinates, revealing itself. A scrollbar, ring or set of rings (e.g. semi-circle or partially-complete circle) can display the level of scroll, whilst still adhering to the feeling of an endless-interface. User interaction can trigger AJAX (or similar technology) to asynchronously load and display information (in the background, without interfering with the display and behavior of the existing interface/page).

3.) One or more interface elements hide and/or reveal themselves by reading the (X, Y) coordinates variable of the 2 nearest orbits and temporarily changing their (X, Y) coordinates property to permit unimpeded passage of new child container(s) from middle behind **7** to (X, Y) coordinates that place it in an “outer orbit” (a secondary encircling group of interface elements), preferably with a gravitational push [no direct contact].

4.) Disappearing of irrelevant interface elements (e.g. coalesce/fly-away animation) upon selection of an element which moves into the central point of reference in the user interface, and performing the relevant task (e.g. expanding to reveal information or refresh shown orbits, or retracting to hide/discard ones displayed).

5.) Separately, Embodiment V. utilizing the (X, Y) coordinates and angle to create a “branch” that leads in a direction, and changing (X, Y) coordinates of containers on that branch as the user moves along it by selecting it or swiping along the “landing-strip” of arrows which light up in every direction, (almost as if the interface is breathing).

6.) Step(s) 1, 2, 3 or 4 or 5 triggered by a specific rotation degree [of the parent container], such as 270 degrees, (X, Y) coordinates, or user interaction.

Alone (or in combination) branch(es) and/or orbiting interface elements and/or paths that create an endless and/or lively user interface (which evoke a sense of unlimited possibilities, akin to the universe). This is done so using branch(es) to navigate and/or orbits/rotation/multi-direction to scroll through interface elements, connecting the user to the user interface just as gravity holds molecules together, or, along a path(s) that represents the interwoven directions of interactivity.

The user interface achieves a fluid-nature regardless of the input method used (e.g. a computer mouse, touch-screen, gesture(s), augmented and/or virtual reality headset(s), peripheral(s), sensor(s), device(s), etc.). This makes it familiar to "pick up and use" by the end user [regardless of input method] due to virtually-identical interactions to accomplish the desired function(s). It can bridge the gap between leisure and productivity, and makes it possible to easily and intuitively control a user interface in ways not conventionally possible.

Rather than being static and stale, orbiting interface elements (and/or branch(es), and/or path(s)), with or without the endless interface, bring fluidity and dynamism to the user interaction. Rotational scroll highlights unlimited possibilities, while narrowing focus enough to increase productivity. And, [if present] pinning interface elements on branch(es) and/or manipulating orbit(s) and/or branch(es) to allow the user to customize their “workflow” (for a personalized user interface), to allow for navigation that’s customized to the user(s) style. This could be further enhanced with modifications to the distance/trajectory/size, shape/positioning/timing of branch(es) and/or orbiting and/or path interface elements.

The current user interface may also be used by people with disabilities (such as A.L.S., M.S., etc.) who can control the interface in a similar manner to how OptiKey can use eye-tracking to deduct keyboard-input (with subsequent text-to-speech conversation output). As an improvement over OptiKey’s eye-tracking user interface, instead of having to look across a slew of overwhelming information, a user with/without a disability can explore in all directions, by gazing their eyes along a desired path, or "landing strip" (e.g. illuminating interface elements in the direction of user interaction (i.e. to permit navigation into an off-screen area)). Instead of a distracting cursor that constantly moves based upon eye movement, an area on screen can act as a “lock” for the cursor, allowing a disabled/regular user to view the user interface in an all of its unimpeded beauty, yet briefly glance upon an interaction pane that can unlock/lock cursor movement. If present, one or more branches represent growth and/or connectivity to the world around us, and, separate or combined, orbits and/or branch structure can change to highlight favored information (most frequently accessed (through machine learning) and/or user-defined), be it apps, actions, people, chats, or other user interface information. In essence, the interface becomes a living embodiment of who we are, reflecting the actions that we take, and, in a way, merging man and machine: an experience intertwined with our interactions and an extension of who we are, rather than just a momentary operation of technology.

Lists and giant blocks of text can make it difficult to focus on, and creates an information overload, causing increased eye strain (due to moving eyes back and forth more excessively than necessary). In comparison to death, horizontal or vertical-only navigation makes the end of scrolling abrupt and unpleasant.

Utilizing orbiting and/or branching interface elements, the user can effectively stare at the same spot (with or without a trigger by user interaction(s)) while the information "comes to them," making it easier to delve deeper, or, allow it to fade into peripheral vision (as the next element(s) appear). Navigating an interface becomes less of a chore and more of a game that coincides with productivity, as information can ripple outwards beyond the bounds of the screen/input device.

Also, it can produce an interface that follows the location of the cursor (controlled by a mouse, gesture, eye-position, input device (e.g. Microsoft HoloLens), etc.), and, by tracking these (X, Y) coordinates, displays orbit(s) (and/or branch(es) and/or path(s)) around it (which follow as the cursor/input-method's location changes). This is akin to how humans expect to interact with the world (they move their body to see and interact with new things in reality (and this interface follows suit).

In an embodiment, the interface could be navigated with only gestures, allowing for the use of a sensor such as an Xbox Kinect, Leap Motion, or Microsoft Hololens to control it (with/without the use of eye-gaze tracking to allow for disabled/regular users to control the interface with and/or without any other method of input). In addition, an Apple TV remote, a WiiMote, an Xbox controller, a PlayStation controller, a mouse, a sensor, a smart piece of clothing (or wearable device), or even brain-activity (as determined by an applicable sensor) could theoretically (and likely will be utilized in the future as the technology becomes available) to control the interface in virtually any way imaginable (including basic/advanced thoughts). Utilizing projectors and/or holographic and/or volumetric displays to control a 3d and/or 2d representation of the orbiting/endless and/or branch and/or path interface could provide a modern means of interacting with the world.

Such an interface could be utilized in the medical field, education field, in the fields of military, public service, government and private/public organization(s). It could be used to display real-time, minimal and crucial information in emergency situations, allowing for quick access to relevant/curated information and/or action(s). Any combination of the claimed invention's methods of interacting with the user interface creates a field of possibilities that is more productive, convenient, fun and applicable to the modern world and the shifting state of technology than current industry approaches. Applicable uses lay in new computer interfaces, new phone interfaces, new augmented/virtual reality interfaces, new internet/browser interfaces, new app interfaces, new software interfaces, new TV interfaces, new holographic/projection interfaces, new neural-network interfaces, new wearable interfaces, etc. (anything that uses a user interface in some form). The claimed invention's interface could also be used as a learning tool for those with disabilities, or to give kids (and/or adults alike) new ways to discover polls/test or answer quizzes in a minimal and focused manner, or allow people to vote in elections or control their home's smart systems/appliances. People could discover new content or view profiles/public communities where people post content through this type of interface, where real time updates allow users to scroll through an unlimited amount of content without any noticeable load-time, due to the nature of the user interface only needing to load one item at a time (doing so before display) utilizing AJAX and/or other methods of real-time information display.

To perform the desired function, user interaction comprises of one or more parts of one or more Embodiments (all of which utilize their requisite function(s) and/or variable(s)/pre-requisites). Best use case(s) of each Embodiment (and the individual/combination(s) are not indicative of the claimed invention's entirety. The claimed invention covers the scope of any amount of interface elements or parent/child container(s) being used to achieve the desired function, through the use of any combination of Embodiment(s) to achieve their function(s), or any reasonably-deductible-similarity). As such, while the best method of how the claimed invention works is explained [utilizing a Bezier curve with the GreenSock API], the claimed invention also covers the utilization of CSS key frames or JavaScript, as well as counterbalancing parent and child containers; it should be understood that this is not limited to one programming language or method, as there are multiple ways to accomplish the desired function, be it utilizing SVG's for animations, CSS, JavaScript, HTML5 canvas, Unreal Engine, Unity Engine, or the countless other method(s) that can be reasonably deduced to be usable to accomplish the desired function(s) (all of which fall within the scope of the claimed invention).

Various illustrative embodiments will now be described in detail with reference to the drawings.

Embodiment I

FIG. 1 shows a schematic diagram of a dynamic user interface in accordance with an illustrative embodiment. As shown in FIG. 1, a central point of reference or hub 7, which may be visible or invisible, is located centrally in each user interface 9, which may be displayed on a video display, for example, of a computing device embodying the user interface (see FIG. 23). A plurality of orbital interface elements 1, 2, 3, 4, 5, 6 are shown circling the central point of reference or hub 7 within an orbit 8.

As shown in FIGS. 2 – 4, the plurality of orbital interface elements 1, 2, 3, 4, 5, 6 can rotate clockwise (FIG. 1 > FIG. 2 > FIG. 3 > FIG. 4 > FIG. 1) or counterclockwise (FIG. 1 > FIG. 4 > FIG. 3 > FIG. 2 > FIG. 1) in their orbit 8 around the central point of reference or hub 7. The orbital interface elements may orbit around a central point of reference or hub 7 on their own, or as triggered by user interaction (e.g. a finger swipe across a touchscreen on a video display. The plurality of orbital interface elements 1, 2, 3, 4, 5, 6 may be acted upon individually, or all at once if they are all selected together. Such an interface streamlines the display of information, allowing the user to focus more intently on what's displayed (as research shows limitations in the number of things humans can focus on simultaneously).

In an embodiment, the central point of reference or hub 7 may be in the center of the interface, or shifted to other areas, such as along each item or node connected to a branch extending radially from the central point of reference or hub 7. This allows many levels of user interfaces to be interconnected, each of which may be accessed by navigating along the paths from one node to another.

In an embodiment, preceding any user interaction, the orbital interface elements can be set to automatically orbit a central point of reference on their own. For example, the orbital speed may be set to perform a full 360 degrees rotation every 24,000 milliseconds. However, it should be understood that the central point of reference can be static or dynamic, and the speed of rotation can be set to anything. Also, the angle at which the orbits counterbalance can fall anywhere within the realm of 360° (so while the best use-case involves a full 360° counterbalance in real-time, this could be substituted for any angle - triggered by user interaction - or not.

For Embodiment I.) During rotation of 9, variable(s) are set. One of these variable(s), logs the angle (degrees) of 9 at any given time. Simultaneously, a variable logs the inverse of this angle. Concurrently [if applicable] a variable is set for the unique (X, Y) coordinates of each child container(s)/interface element(s) of 9 (1, 2, 3, 4, 5, 6), with more/less orbits and/or variables depending upon the amount of components in 9). During rotation of 9: 1, 2, 3, 4, 5, 6 (all parts of 9, (*not limited or maximized to 6 parts*), transition along 8, each with their angle of rotation adhering to the inverse variable of the parent container (to counterbalance the angle of 9 in real time), and set their own unique variable of (X, Y) coordinates (if applicable). Rotation of 9 (and the contained orbits, represented by 1, 2, 3, 4, 5, 6) can occur clockwise and/or counter-clockwise. Before and/or after user-interaction with 9, FIG. 1, FIG. 2, FIG. 3, FIG. 4 cycle between each other on a loop, creating an “orbiting” appearance. Best use case of this has 9 make a full rotation (360°) every 24,000 milliseconds (with 1, 2, 3, 4, 5, 6 (any/all parts of 9) adhering to the parent container's inverse angle’s variable simultaneously). If 7 is present, it is a separate element from 9, and usually remains in front of 9 at all times (unless a child container/interface element goes in front of it during Embodiment IV.)). During user interaction with 9, the speed of rotation changes based upon the speed of the input method (such as swiping faster/slower on a touch screen), allowing users to control the pace of orbit in a clockwise/counter-clockwise direction as they see fit.

In this embodiment, the orbital interface elements are each defined by a unique variable (represented in the diagram by 1, 2, 3, 4, 5, 6 (each separate child container(s) with their own variables). Each child container is defined by a second unique variable ($\underline{\theta}$, $\underline{\delta}$, $\underline{\lambda}$, $\underline{\eta}$, $\underline{\mu}$, $\underline{\psi}$), if used in conjunction with Embodiment II.). In order to orbit and counterbalance, all child-container(s) are a part of a rotating parent-container 9. Upon rotation of the parent container (9), the requisite variable(s) are set ($\underline{\varepsilon}$, $\underline{-\varepsilon}$, and 1, 2, 3, 4, 5, 6 (+/- additional orbits, depending on amount of child-container(s) in 9, as well as $\underline{\theta}$, $\underline{\delta}$, $\underline{\lambda}$, $\underline{\eta}$, $\underline{\mu}$, $\underline{\psi}$ (if applicable), +/- variables depending on amount of child-container(s) in 9 and the requisite variable(s) needed for the desired function(s)). During rotation of 9, the variable $\underline{\varepsilon}$ logs the angle (degrees) of 9 at any given time, and the variable $\underline{-\varepsilon}$ simultaneously logs the inverse of $\underline{\varepsilon}$. During rotation of 9, all parts contained therein (i.e. 1, 2, 3, 4, 5, 6 +/- additional Embodiments) individually change their angle to match $\underline{-\varepsilon}$ in real time. For example, if 9 is rotated 270°, $\underline{\varepsilon} = 270°$ and $\underline{-\varepsilon} = -270°$ (and the angle of 1, 2, 3, 4, 5, 6 (+/- variables depending on amount of child-container(s) in 9) would each be set to -270° individually, as their angle is relative to 9 and not the interface as a whole (if 1, 2, 3, 4, 5, 6 (+/- child containers) were set to 0°, all of the containers within **9** would appear to be the same angle as **9**)).

When Embodiment I.) is used in conjunction with Embodiment II.), all parts of 9 concurrently set the unique (X, Y) coordinates of their respective position (represented by the variable(s) $\underline{\theta}$, $\underline{\delta}$, $\underline{\lambda}$, $\underline{\eta}$, $\underline{\mu}$, $\underline{\psi}$, with more/less variables depending on the amount of orbits contained in 9). The trajectory of 8 is optional, but works best when shaped as a perfect circle (same length/width). The distance between all parts of 9 is optional, but works best when spaced evenly between each other (and 7, if applicable). However, all distances/spacing of orbits within 9 are comprised of the claimed invention. The size and existence of 7 is optional, but best when present, and, larger than a particular element of 9 (or else, if the same/smaller, that the respective element utilize the aspect of Embodiment IV.) and/or scales the respective element larger/smaller if coalescing/retracting to/from 7). In Embodiment II.), 1, 2, 3, 4, 5, 6, 7, 8 all utilize Embodiment I.) (which counterbalances utilizing $\underline{-\varepsilon}$ applied to all child containers and/or interface elements. $\underline{\theta}$, $\underline{\delta}$, $\underline{\lambda}$, $\underline{\eta}$, $\underline{\mu}$, $\underline{\psi}$, $\underline{\zeta}$, $\underline{\varphi}$ have their variables change in order to converge in the center behind 7. Optionally, (but for the best use case), FIG. 5b and FIG. 8c represent the beginning or end of content, while FIG. 8b and FIG. 5c also represent the beginning/end of content. They could be flipped differently (such as 90 degrees to have a "yaw" movement).

One variable sets the (X, Y) coordinates individually for each child-container/interface element to consistently update with their current position. Another variable is set to match the angle (degrees) of the parent container at any given time, updating in real-time as any changes to the rotation occur. Inverse of this simultaneously is set as another variable (the opposite of the parent container's rotation angle at any given time, also updating as changes occur). This allows child containers/interface elements to counter-balance the rotation of the parent container in real time, ensuring they're always displayed in an upright position (while orbiting around a central point of reference through animation, or as a rotating-type scroll-interaction). This orbiting animation or scrolling-interaction rotates the parent container clockwise or counter-clockwise, setting the specified variables. Each child container/interface element individually changes its rotation property to counter-balance (reading from the variable that's inverse of the parent's rotation angle). To the user, this displays a uniform orbit around a central reference point. It also gives the appearance that each child container/object has a static individual rotation of 0 degrees (whilst orbiting). In reality, 0 degrees cannot simply be specified for child-containers, as it would pertain only in relation to the parent container (appearing upside down as it orbits). It therefore needs to utilize the variable which counter-balances it (e.g. if one variable reads "90" a corresponding variable should read "-90"). Additional interactions utilizing these variables make it possible to trigger additional actions, [in respect to specific rotation degrees or (X, Y) coordinates], such as an animation trigger to hide/reveal interface elements.

A parent container and the variables ensures that child containers/interface-elements contained therein are able to counterbalance and always display upright as they orbit around. A user can interact with the orbiting interface elements by spinning them around clockwise or counter-clockwise and selecting them. It works best when FIG. 1, FIG. 2, FIG. 3, FIG. 4 and FIG. 1 of Embodiment I.) loop between themselves (preceding and proceeding user interaction) at a rate of 360 degrees every 24,000 milliseconds (however this timeframe is optional, as any/all time-intervals can be used for the rotation). This allows a pleasant orbiting to occur, which brings life to the interface. To work best, it is preferable to utilize Embodiment II.) after all 4 figures of Embodiment I.) have concluded, allowing the user to begin "scrolling" through additional orbits, beginning when the parent container is rotated to an angle of 270 degrees. From here, users can keep scrolling clockwise and have a new orbit appear every 30 degrees (with the ability to reverse the process by scrolling counter-clockwise, doing the exact same thing that started at 270 degrees, except in reverse). These degrees (270 and 30, respectively) represent the best use cases, but are optional, as the claimed invention covers the scope of any angle(s) being used for this Embodiment(s) (or other Embodiment(s) individually, or together). It should also be understood that the claimed invention covers the interface elements entering after any period of time (so, if the user was to “play-around” by orbiting the same items currently displayed on the interface, after any interval of time, it could then begin the process of scrolling through additional interface elements (creating either a fluid or a delayed effect before/after scrolling through interface elements preceding/during/proceeding user interaction)).

By utilizing this form of minimalism, whilst applying game mechanics to the user-interaction, navigation of the user interface becomes fun and engaging for the user. With the option to eliminate scroll bars and other conventional user interface controls, a seemingly-endless stream of information can be displayed (with technology such as AJAX).

If keeping scrollbars, this can be implemented into the interface using new methods (i.e. such as a ring, or a set of them, with a cutout to differentiate between beginning and end of information). Space can also be freed-up/utilized to display parcels of information and interface elements in new ways.

By way of example, and not by way of limitation, the following code provides an example of how the animation of the orbital interface elements may be achieved.

CSS3 Animation Keyframes:

Code to constantly counterbalance the object rotation, may be written as follows:

```
transform: rotate(45deg) translateX(150px) rotate(-45deg);
```

However, as constantly counterbalancing the orbital interface elements might have a negative effect on performance, the following code may be used:

jQuery:

This is the JavaScript code involved for an orbit:

```
 ( function ( $ ) {
        jQuery.fn.orbit = function(s, options){
            var settings = {
                orbits:     1     // Number of times to go around the orbit e.g. 0.5 = half an orbit
                ,period:    3000     // Number of milliseconds to complete one orbit.
                ,maxfps:     25     // Maximum number of frames per second. If # is
small gives a "flicker," and, being too large, uses lots of CPU power
                ,clockwise: true  // Direction of rotation.
            };
 $.extend(settings, options);  // Merge the supplied options with the default
.settings.
            return(this.each(function(){
                var p        = $(this);
```

```
/* First obtain the respective positions */
            var p_top    = p.css('top' ),
               p_left   = p.css('left'),
               s_top    = s.css('top' ),
               s_left   = s.css('left');

/* Then get the center positions of the objects */
            var p_x      = parseInt(p_top ) + p.height()/2,
               p_y      = parseInt(p_left) + p.width ()/2,
               s_x      = parseInt(s_top ) + s.height()/2,
               s_y      = parseInt(s_left) + s.width ()/2;
/* Find the adjacent and opposite sides of the right-angled triangle */
            var a        = s_x - p_x,
                 o        = s_y - p_y;
/* Calculate the hypotenuse (radius) and the angle(s) separating the objects */
            var r        = Math.sqrt(a*a + o*o);
            var theta    = Math.acos(a / r);
/* Calculate the number of iterations to call setTimeout(), the delay and the "delta" angle to add/subtract */
var niters  = Math.ceil(Math.min(4 * r, settings.period, 0.001 * settings.period * settings.maxfps));
            var delta    = 2*Math.PI / niters;
            var delay    = settings.period  / niters;
            if (! settings.clockwise) {delta = -delta;}
            niters      *= settings.orbits;
/* create the "timeout_loop function to do the work */
            var timeout_loop = function(s, r, theta, delta, iter, niters, delay, settings){
                  setTimeout(function(){
/* Calculate the new position for the orbiting element */
                  var w = theta + iter * delta;
                  var a = r * Math.cos(w);
```

```
                var o = r * Math.sin(w);
                var x = parseInt(s.css('left')) + (s.height()/2) - a;
                var y = parseInt(s.css('top' )) + (s.width ()/2) - o;
/* Set the CSS properties "top" and "left" to move the object to its new position */
                p.css({top:  (y - p.height()/2),
                    left: (x - p.width ()/2)});
/* Call the timeout loop function if we haven’t done all the iterations yet */
```

While the above code would allow for smooth movement of the orbital interface elements, it involves writing many low-level functions, like the mathematics of movement, time calculation, etc.

In contrast, using a web animations API such as one offered by GreenSock, it takes only one function to define the motion of an object, using the notion of Bezier curve. In this case, the programmer needs to identify an array of relative points as inputs to specify the path, the degree of curvedness, etc., and the API will takes care of creating the path:

GreenSock API:

```
TweenMax.to(logo, 1, {bezier:{type:'quadratic', values:[
       /*p1*/{x:0, y:0},{x:100, y:0},{x:100, y:100},
        /*p2*/{x:100, y:200},{x:0, y:200},
        /*p3*/{x:-100, y:200},{x:-100, y:100},
        /*p4*/{x:-100, y:0},{x:0, y:0}]}/*bezier end*/,
       ease:Linear.easeNone});
```

Thus, utilizing a web animations API such as the GreenSock API provides the functionality to handle user interactions, and optimize for smooth transition and performance usage.

Embodiment II

Now referring to FIGS. 5a – FIG. 8b, shown is another embodiment which utilizes the functions and/or variables of the earlier embodiment described with reference to FIGS. 1 – 4 in order to trigger the rotational/orbital scroll effect (as well as have containers or interface elements remain upright during orbit). The figures show a clockwise rotation (FIG. 5a > FIG/ 6a > FIG. 7a > FIG. 8a > FIG. 5a) or a counterclockwise rotation (FIG. 5a > FIG. 8a > FIG. 7a > FIG. 6a > FIG. 5a) of the orbital interface elements around the central point of reference or hub 7.

In this illustrative example, an orbital interface element can be moved into or out of the central point of reference or hub 7, such that a fixed number of orbital interface elements (in this example, six) can remain in orbit. For instance, if there are 8 orbital interface elements, 2 of the orbital interface elements may be hidden within or behind the central point of reference or hub 7. As such, it follows in chronological order of FIG. 5a > FIG. 6a > FIG. 7a > FIG. 8a upon clockwise rotation (or in reverse for counter-clockwise (FIG. 8a > FIG. 7a > FIG. 6a > FIG. 5a)), when not used with their "FIG. 5/6/7/8b" counterparts.

However, when these counterparts are used, the order goes as follows for clockwise rotational scroll: FIG. 5a & FIG. 5b > FIG. 6a & FIG. 6b > FIG. 7a & FIG. 7b > FIG. 8a & FIG. 8b. For counter-clockwise rotational scroll, the order would be as follows: FIG. 8a & FIG. 8b > FIG. 7a & FIG. 7b > 6a & FIG. 6b > FIG. 5a & FIG. 5b.

The reasoning for this order is as follows: FIG. 5a & FIG. 5b exist together, as it replaces there being nothing (or something else) in place of 7. It also uses the scroll indicator (16) and scroll ring (15) to give an indication of how far the user has/hasn't scrolled through the information contained in the user interface (represented by the orbiting spheres (1, 2, 3, 4, 5, 6, 10 and 11)), which shows that nothing has been scrolled yet (as 2 has not yet receded behind FIG. 5b and 16 is at the very top left of 15).

If a user selected 13, it would have no effect, as there is no way to scroll to the first item in a user interface if it is already displayed (just as you can't scroll any further up a webpage if the scrollbar is already at the top). Dragging 16 to the middle of 15 or selecting 12, however, would allow the user to scroll through additional interface element(s) (which in this case has occurred, and therefore the reasoning for an arrow showing how the movement of 2 will move to take place behind 7 and transition from FIG. 5a & FIG. 5b to FIG. 6a & FIG. 6b, where 10 has emerged from behind 7 (as shown by the arrow going upwards), as well as 16 has moved to the middle to represent that it has scrolled through 50% of the information in the user interface (since, in this example, there are only 8 interface elements, represented by orbits that are either shown, or hidden).

Then, when it transitions from FIG. 6a & FIG. 6b to FIG. 7a & FIG. 7b, it shows that 1 is about to recede behind 7 at the same angle that 2 did (which, in this case is 270^0). Since it has not yet receded, FIG. 7b coincides simultaneously with FIG. 7a, to show that it is still only halfway through all of the information available to the user in the user interface.

Following 270 degree parent container rotation from Embodiment I.) Embodiment II.) goes in order of FIG. 5, FIG. 6, FIG. 7, FIG. 8 to "rinse and repeat," hiding an old orbit and displaying a new one every 30 degrees of parent container rotation. This happens in a clockwise direction, but upon the beginning of a counter-clockwise direction of scroll, the order of Embodiment II.) reverses to hide newer objects and display the old hidden ones. The claimed invention does not limit the direction used to hide/reveal orbits, however.

As there are only 8 interface elements in this example, once the same process that was used to transition between FIG. 5a and FIG. 6a, FIG. 8a shows that 11 has appeared from behind 7, and, as such, FIG. 8b now shows 16 at the bottom left of 15, due to 11 being the last interface element to be scrolled through (akin to having scrolled to the bottom of a page in a web browser), thereby letting the user know that they are at the end of all the information in the user interface. They can scroll 16 counterclockwise or press 12 when in FIG. 8a or FIG. 6a to reverse the interface elements they have scrolled through (as, when interface elements such as 1 and 2 are scrolled through and disappear behind the middle, they are done so in a way similar to stacking a deck of cards, where you can then retrieve the interface elements in reverse order from how they were navigated through in the user interface). If many interface elements existed (such as a contact/chat list comprising of dozens or hundreds of people/chats), 14 could be selected in order to “shuffle” through the content displayed, where all the orbiting elements recede behind 7 and a random set of interface elements (in this case, orbits) appear from behind 7.

Upon user interaction, all of the orbital interface elements rotate around the central point of reference, for example following a Bezier curve. The placement of 9 represents the current angle of the container at any given time, which is utilized to determine when to trigger actions for interface element(s) (such as in rotational scroll, having one or more interface elements recede behind **7**.

Embodiment III

Now referring to FIG. 9, shown is a schematic diagram of another embodiment of the dynamic user interface having multiple concentric orbital rings or paths.

This embodiment uses the function(s)/variable(s) from Embodiment I to form a second (or additional) ring(s) of interface elements (orbits) around the outside of the first ring (using a Bezier curve with larger radius('s) than the inner ring (using the GreenSock API, or, counterbalancing the parent container from the child container(s) (if that method is employed), all whilst using the function(s)/variable(s) of 2. Embodiment II.) to scroll (altering the (X, Y) coordinates of one or more existing orbits/interface elements to permit transit of new orbits to their outside/respective ring without making contact with one other, just as 2 magnets can repel one another if they get too close)).

Embodiment IV

Now referring to FIGS. 10a – 13b, shown is the dynamic user interface of FIG. 9, illustrating how orbital interface elements may be moved between a "higher" and "lower" orbit ring relative to the hub. All orbits could come from the same origin (i.e. the same x,y coordinates) and/or go the same x,y origin (central reference point).

This embodiment utilizes the functions and variables from Embodiment I to determine the number of orbits/interface elements and follow a Bezier curve (or counterbalance, if necessary), and their respective coordinates in Embodiment(s) II.), III.), V.), and/or VI.) to coalesce orbit(s) into the central reference point by the shortest route possible (or repel off into the distance, dependent upon the type of user interaction or display of the user interface).

This allows the orbital interface elements to recede and reappear from behind the center object **7**, or off in the distance, which could be modified to have orbits get smaller/bigger as they arrive/leave visibility.

Embodiment V

FIGS. 14 – 18 show schematic diagrams of another embodiment of a dynamic user interface in which orbital interface elements may be rotated over a spoke or branch to travel to an outer spoke node, or towards the central hub.

As exemplified by 38, the size of items connected to the branch is not limited to a single size, but rather could consist of interface elements/child containers of varying shapes and sizes. Each successive [connecting] branch is a child container of the previous, while all child container(s) of the entire branch as a whole. This allows for manipulation of a single branch individually, or multiple branches simultaneously, respecting the positioning of their adhering child and/or parent branches. Successive branches can be defined by the subsequent one(s) after a connected interface element, and/or when the branch changes direction. Distance between interface elements is optional, but best when spaced uniformly along branch(es). Upon selection of 15, 16, 17, 18, 19, 20, a fixed and/or variable (X, Y) scroll dependent upon interface design (whether it takes into account the amount of “out of bounds” elements or not) and whether it’s a hover in the direction of scroll, or a click of the button (could scroll out of bounds until user ceases hover in the desired direction).

As shown in FIGS. 19 – 22, Embodiment V.) may utilize one or more of Embodiments I.), II.), III.), IV.) to create an "endless interface" that utilizes the angle and (X, Y) coordinates to create a "branch" for each interface element or orbit - leading outside of the boundaries of the screen (yet navigable upon user interaction, such as mouse-over, click/tap, or swipe (in the respective direction along a pulsating landing-strip of arrows)).

Embodiment V.) thus permits navigation in any direction in which branches exist, and the orbital interface elements may be oriented to re remain upright at all times.

Optionally, Embodiment V. could employ one or more features from Embodiment(s) II.), III.) and/or IV.). For example, orbiting interface elements could be "anchored" to a galaxy's spiral-arms (or a tree's branches/roots), and moved around accordingly. Coordinates or angles of rotation that dictate changes in the interface or are triggered by user interaction could be altered (such as instead of it being at 270 degrees that an orbit hides, it could be at 300, or after a specific amount of rotations + a specific angle). Hiding/revealing interface elements could be done so in a spiral or other formation, of one or more item(s) at a time followed by the next and the next until all have disappeared behind the middle (in reference to Embodiment IV.). Going in one after another and getting smaller, it'd create a "closing aperture" type effect. Elements could get bigger or smaller as they reveal/hide in any of the Embodiments. Orbits could act like a shooting star upon user interaction, shooting up to expand into something more. An orbit could also go in front of **7** and expand into the entire middle to show more information. Parent container size/shape could change, as well as different shaped interface elements (such as rounded rectangles or hexagons) could replace some or all of the shapes. Interface elements could be larger/smaller, farther apart, or orbit a trajectory of a different shape (e.g. oval instead of perfect sphere, or matching up with real astronomical data). Interface elements could be non-uniform (e.g. spread out in a shape like a spiral, yet still orbiting with counter-balance).

<u>EmbodimentI VI</u>

Referring back to FIGS. 14 – 22, Embodiment VI.) may comprise and endless orbit interface utilizing one or more of Embodiments I.), II.), III.), IV.) and V.)

<u>Embodiment VII</u>

Embodiment VII.) may comprise an endless branch interface utilizing one or more of Embodiments I.), II.), III.), IV.) and V.).

<u>Embodiment VIII</u>

Embodiment VIII.) may comprise an endless interface utilizing V.) and VI.) with or without additional Embodiments.

<u>Illustrative Use Cases</u>

Using an input method such as a phone, tablet, sensor or device (and/or the user’s physical body), the user would scroll/swipe/gesture/verbally-command the direction they wish to navigate (a verbal example being “Scroll Ahead”). Since the parent container rotates, the user can swipe in the direction they wish the orbits to go (clockwise/counter-clockwise). Through a mouse, the user could use the scroll wheel, motion tracking of a controller, the analogue sticks, D-pad, and/or touch-pad. Using gestures, the user could make a tilting motion to one side or the other, causing the parent container to rotate as desired (thereby rotating the orbits and "scrolling"). Making a closed fist or holding a still palm over an object for a couple seconds could select said object (as well as through touch/click through a touch-screen/pad or computer). Rotating the parent container clockwise is akin to "scrolling down," and rotating the parent container counter-clockwise is akin to "scrolling up." Potentially, users could store "favorites" in an outer orbits of sorts that acts like a bulletin board.

Additionally: The endless interface and rotational/orbit scroll provide unique value on their own or used in combination with other elements. Many possibilities are inherent in singular use or combination of I.), II.), III.), IV.), V.), VI.), and VII.) utilizing one or more parts of one or more Embodiments. One such possibility involves a specific rotation degree or (X, Y) coordinates of parent and/or child container(s)/interface element(s) trigger something in the user interface, such as an animation, viewing more/less elements, or meeting pre-requisite for further user interaction. Users could interact with the interface through a browser, or to manage files or socialize through an application or website that utilizes orbits (and/or the branches) with or without the endless interface. Another method could be for the control of software programs such as a game engine or game itself, allowing the user to see a slew of options (such as raise/lower terrain, paint textures, control lighting, sounds, animations, place static meshes, etc.). Upon selection of an option, more/less interface elements could show in the orbiting (and/or branches) +/- the endless interface. The user could “pin” things to an inner or outer-orbit that moves at a slower/faster pace, or on specific branches to keep track of things categorically. It could be used to replace the taskbar and/or desktop of an online and/or offline operating system (or the entire OS interface). This outer-orbit trajectory (or that of any orbiting elements themselves and/or branches) could go in/outside the bounds of the screen itself (or, peripheral vision/direct field of view, in the case of an augmented and/or virtual reality headset(s)), unleashing the feeling of unlimited possibilities. This outside-the-bounds-orbiting and/or branches could have new things pop in from time to time, opening up the ability to rediscover favorite or forgotten apps, music, videos, websites, people and information of other sorts, as this could typically remain at the bottom of a list or on a rarely-trafficked screen, if navigated through conventional means (currently employed on computers, devices and websites). And any other, reasonably-deductible similarity to, or combination of, the claimed use cases.

Also, it can create: An interface navigated with only eye movement or gestures, allowing for the use of a sensor such as an Xbox Kinect, Leap Motion (or augmented/virtual reality headset) to control it. In addition, an Apple TV remote, a WiiMote, an Xbox controller, a PlayStation controller, a mouse, a sensor, a smart piece of clothing (or wearable device), or even a brain-activity reader could theoretically be used to control the interface in virtually any way imaginable (including basic/advanced thoughts). Utilizing projectors and/or holographic / volumetric displays to control a 3d and/or 2d representation of the orbiting, (and/or branches) +/- the endless interface, could provide a modern means of interacting with the world/interface.

Additionally: An interface utilized in the medical field permitting the display and/or control of doctor/patient information (statically or in real time) and communicate or issue commands to nurses/doctors/patients. It could be used in the field(s) of military and intelligence, to show active threats and/or display crucial information with/without tactical appraisal and to view and/or issue commands, or for control of a vehicle, ship, aircraft or spacecraft. It could be used to display information from a database, and in public service (i.e. police, firefighters, paramedics), and/or display ongoing, past or predicted crimes/incidents and to log and access information of the public and/or government officials (e.g. direct access to such contact(s) in issues of immediate precedence, such as backup from other police/military/public-service/government action, in the case of protests, fires, medical emergencies, etc.). It could be used to display minimal and crucial information in any normal/emergency situation, allowing for quick access to relevant/curated information and/or action(s). It could be used in the workplace (e.g. in the field of retail to control a cash register or view information pertaining to product(s)/service(s)/inventory/financial(s) data). It could also be used to control crucial infrastructure, such as that in the energy and communications sector.

Any singular use or combination of the [or a reasonably deductible similar] method(s) of interacting creates a field of possibilities that is more productive, convenient, satisfying and applicable to the modern world and the shifting state of technology than current approaches. Applicable uses lay in new computer interfaces, new phone interfaces, new augmented/virtual reality interfaces, new car interfaces, new internet/browser interfaces, new app interfaces, new television interfaces, new holographic/projection interfaces, new neural-network interfaces, new wearable interfaces, etc. The interface could also be used as a learning tool for those with disabilities, or to give kids and/or adults alike new ways to discover information. It could be used by the education system or workplace to view/answer quizzes/tests in a minimal and focused manner. People could discover new content or view profiles/public communities where people post content through this type of interface, where real time updates allow users to scroll through an unlimited amount of content without any noticeable load-time, due to the nature of the interface only needing to load one item at a time (doing so before display) with AJAX (or similar technology).

Now referring to FIG. 23 shown is a schematic block diagram of a generic computing device that may provide a suitable operating environment in one or more embodiments. A suitably configured computer device, and associated communications networks, devices, software and firmware may provide a platform for enabling one or more embodiments as described above. By way of example, FIG. 23 shows a generic computer device 2300 that may include a central processing unit ("CPU") 2302 connected to a storage unit 2304 and to a random access memory 2306. The CPU 2302 may process an operating system 2301, application program 2303, and data 2323. The operating system 2301, application program 2303, and data 2323 may be stored in storage unit 2304 and loaded into memory 2306, as may be required. Computer device 2300 may further include a graphics processing unit (GPU) 2322 which is operatively connected to CPU 2302 and to memory 2306 to offload intensive image processing calculations from CPU 2302 and run these calculations in parallel with CPU 2302. An operator 2310 may interact with the computer device 2300 using a video display 2308 connected by a video interface 2305, and various input/output devices such as a keyboard 2310, pointer 2312, and storage 2314 connected by an I/O interface 2309. In known manner, the pointer 2312 may be configured to control movement of a cursor or pointer icon in the video display 2308, and to operate various graphical user interface (GUI) controls appearing in the video display 2308. The computer device 2300 may form part of a network via a network interface 611, allowing the computer device 2300 to communicate with other suitably configured data processing systems or circuits. A non-transitory medium 2316 may be used to store executable code embodying one or more embodiments of the present method on the generic computing device 2300.

While illustrative embodiments have been described above by way of example, it will be appreciated that various changes and modifications may be made without departing from the scope of the invention, which is defined by the following claims.

CLAIMS:

1. A dynamic user interface, comprising:

a central point of reference or hub; and

a plurality of orbital interface elements adapted orbit the hub;

wherein the rotation of the orbital interface elements is dynamically animated, such that the orbital interface elements appear to rotate around the hub in real time.

2. The dynamic user interface of claim 1, wherein the orbital interface elements are counterbalanced such that they are continually oriented in an upright position as they orbit the hub.

3. The dynamic user interface of claim 1, wherein the orbital interface elements are adapted to spin in a clockwise or counterclockwise direction in dependence a direction of a user input.

4. The dynamic user interface of claim 3, wherein the spinning speed of the orbital interface elements is controlled by the speed and force of a user input.

5. The dynamic user interface of claim 1, wherein the central point of reference or hub includes a scroll ring which is formed in an arc stretching around at least a segment of the hub.

6. The dynamic user interface of claim 1, further comprising multiple orbital paths around the central point of reference or hub, each orbital path providing a path around which a plurality of orbital interface elements spin around the central point of reference or hub.

7. The dynamic user interface of claim 1, further comprising a plurality of spokes or paths which radiate outwardly from the central point of reference or hub.

8. The dynamic user interface of claim 6, wherein each of the spokes or paths ends with a node which can receive an orbital interface element swiped towards it.

9. The dynamic user interface of claim 7, wherein the node is adapted to perform a programmed function.

10. The dynamic user interface of claim 7, wherein the node is a pathway to another layer of the user interface.

11. The dynamic user interface of claim 1, further comprising omnidirectional branches and roots extending in any direction from the central point of reference or hub or any node of the interface.

ABSTRACT

There is disclosed a method of displaying and/or interacting with a user interface by orbiting and/or rotating and/or manipulating interface elements, creating an endless, lively interface is disclosed. By arranging interface elements around a visible/invisible central point of reference or hub, and/or along visible/invisible branches (e.g. vines, tree-roots/branches, chains, spider web, neuron-synaptic connections, sunrays, galaxy/star-systems, paths), a virtually endless user interface becomes possible, and opens up numerous navigation and display techniques.

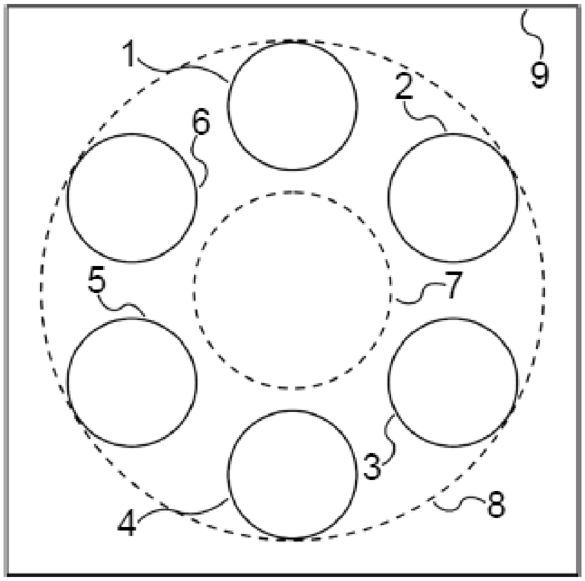


FIG. 1

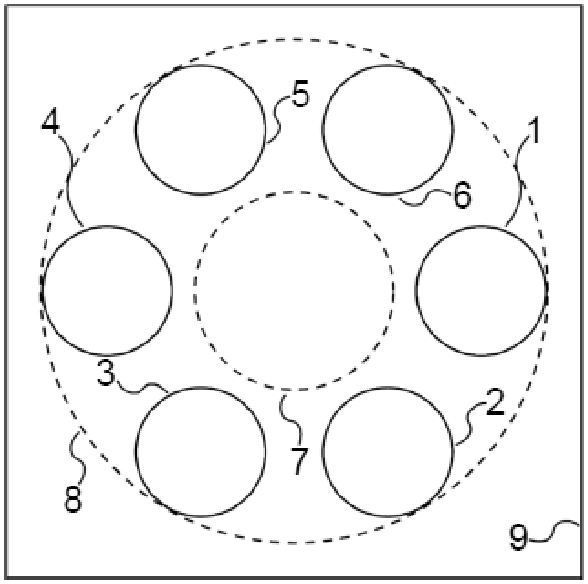


FIG. 2

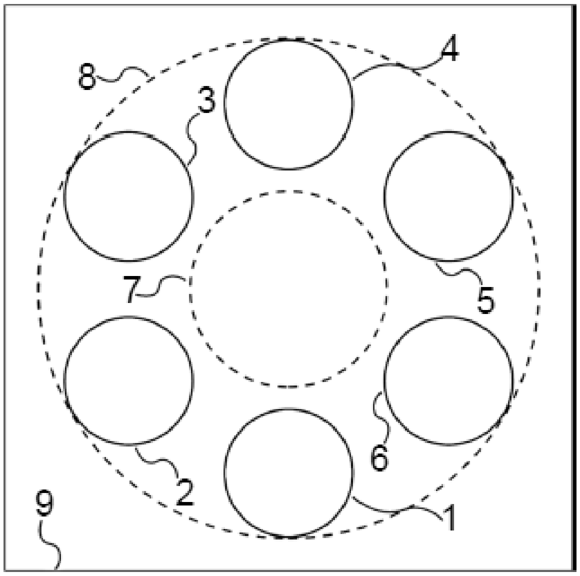


FIG. 3

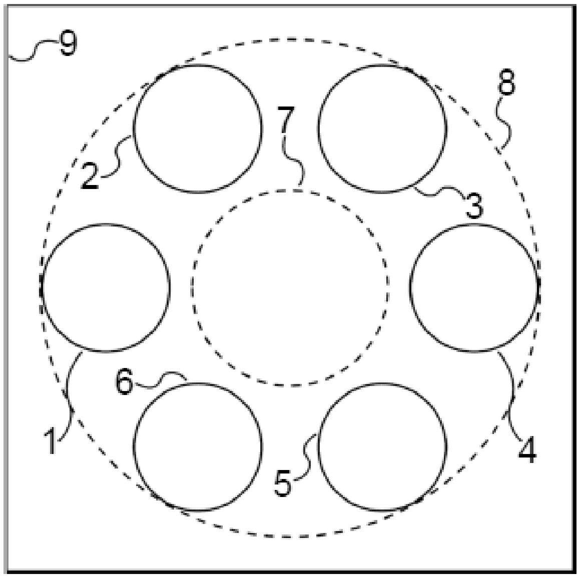


FIG. 4

Component II.)

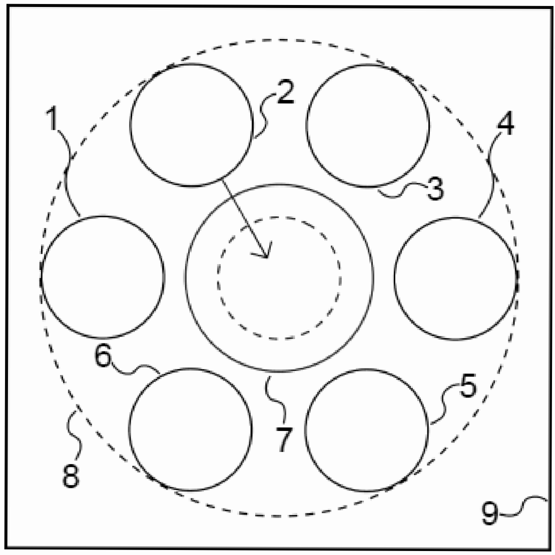


FIG. 5a

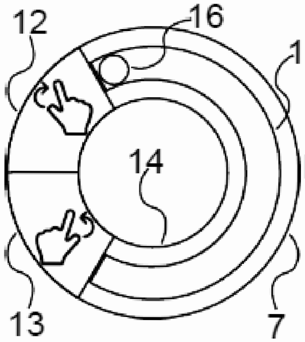


FIG. 5b

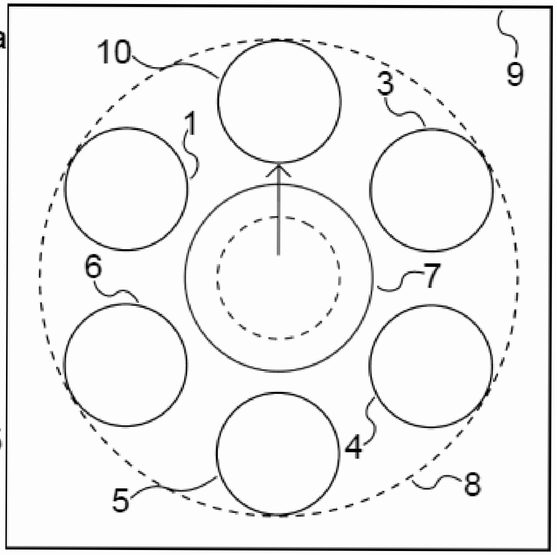


FIG. 6a

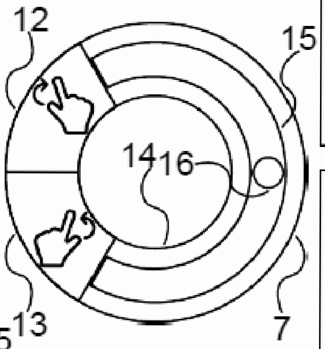


FIG. 6b, 7b

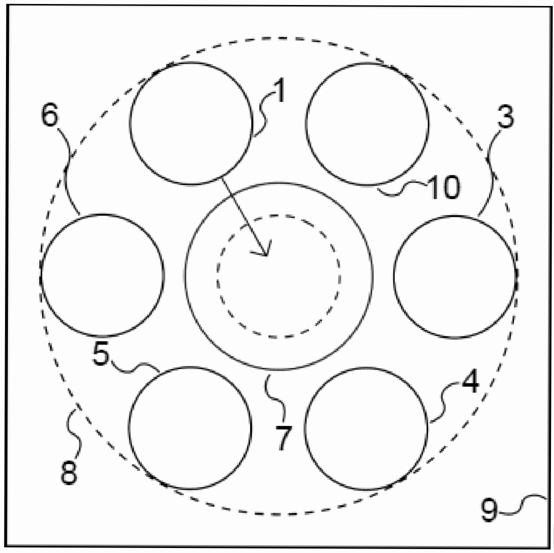


FIG. 7a

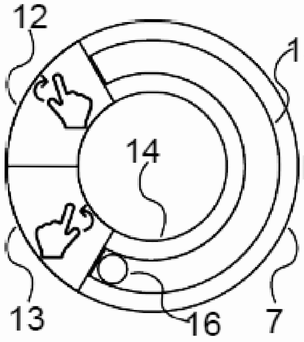


FIG. 8b

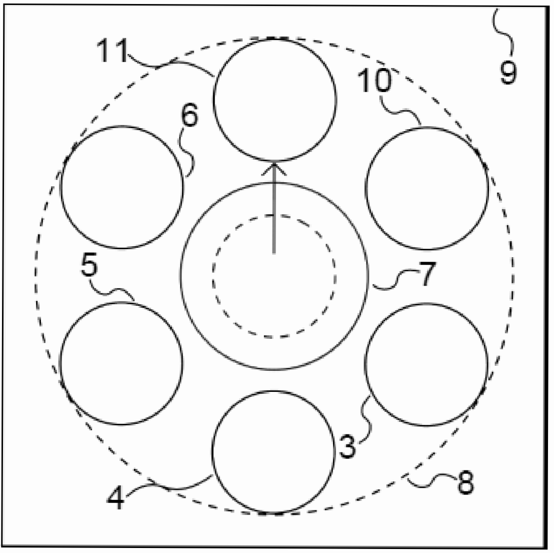


FIG. 8a

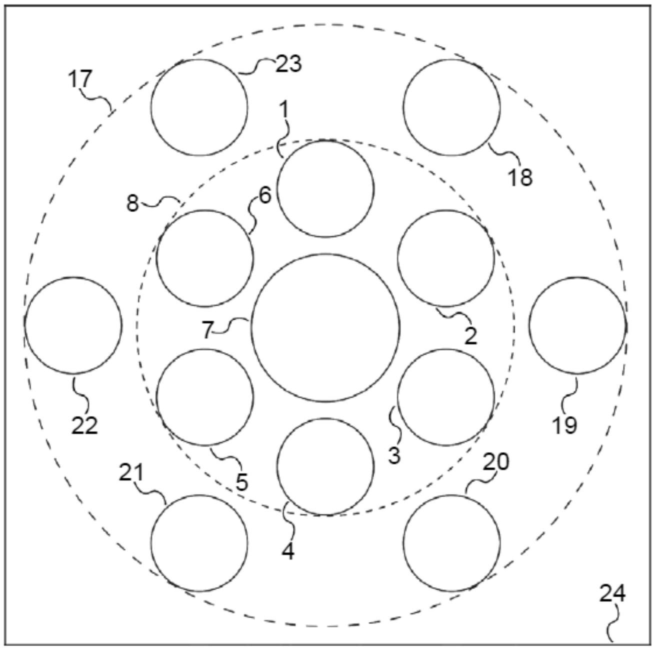


FIG. 9

FIG. 10a

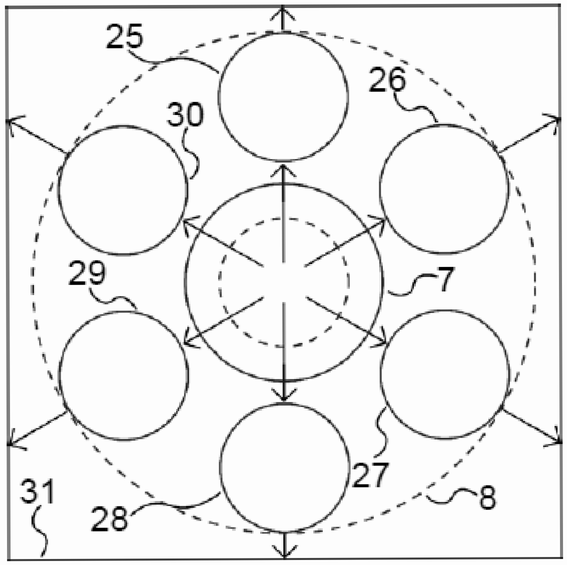


FIG. 10b, 13b

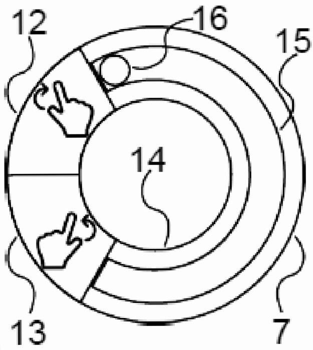


FIG. 11a

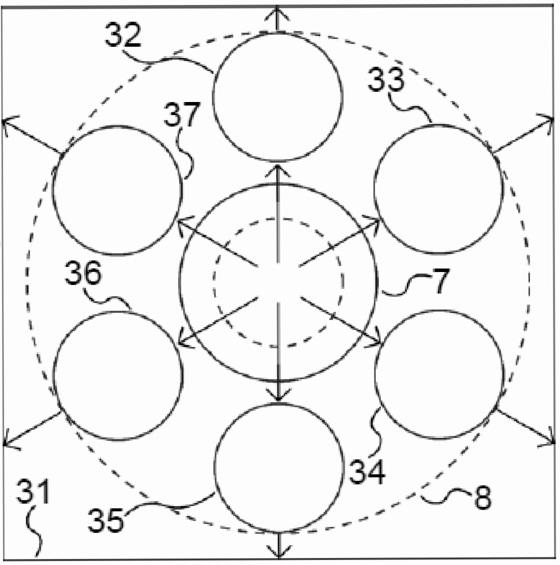


FIG. 12a

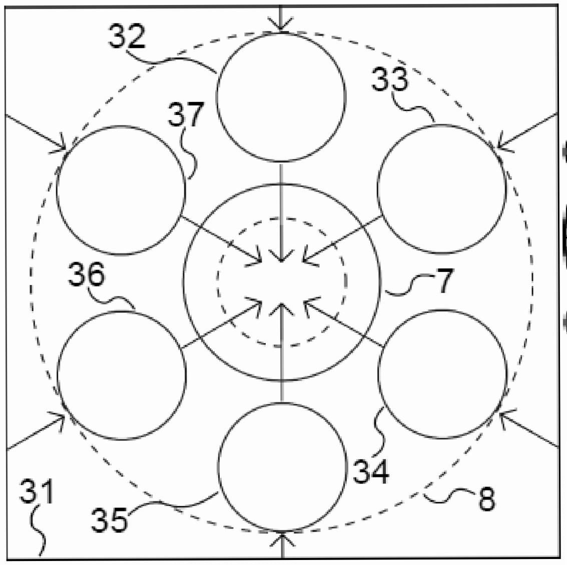


FIG. 11b, 12b

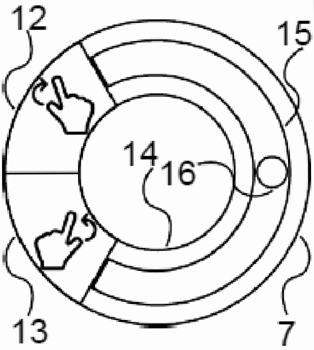
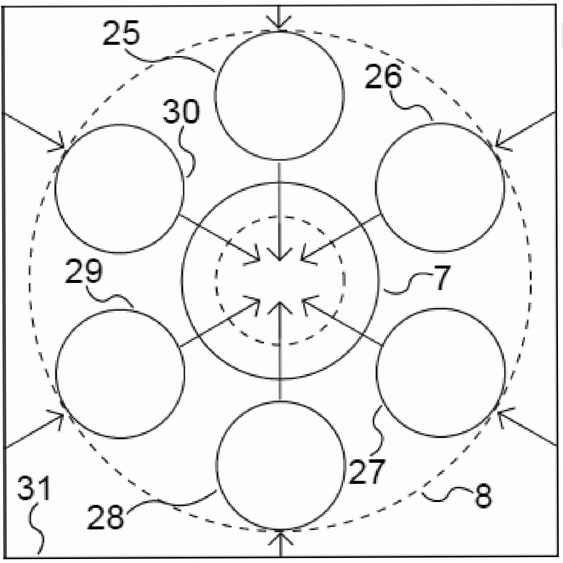


FIG. 13a



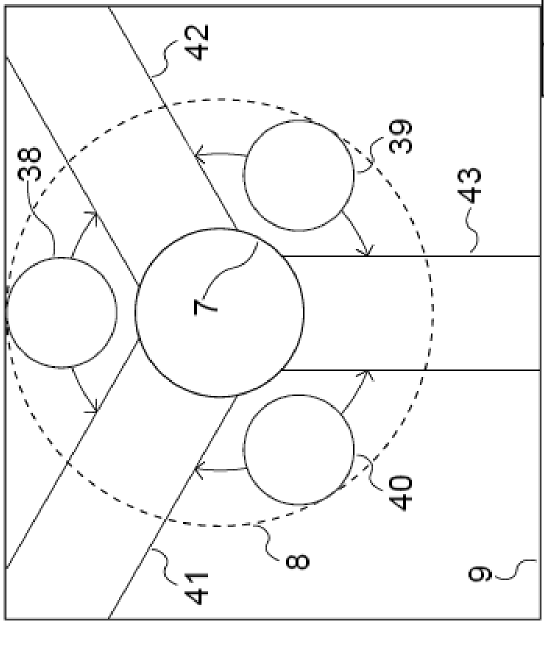


FIG. 14

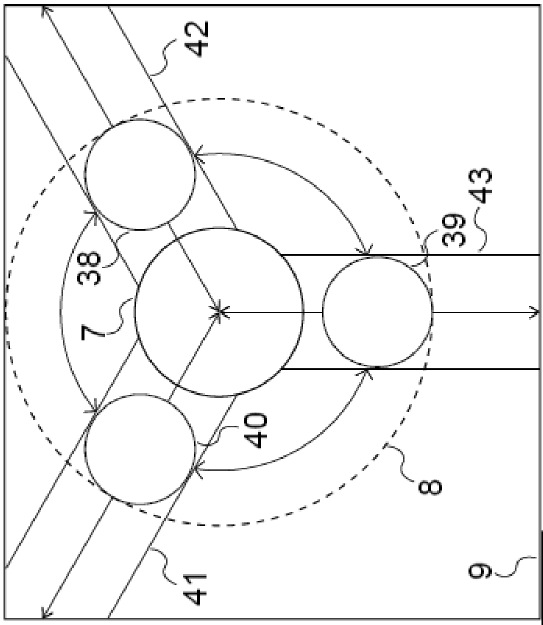


FIG. 15

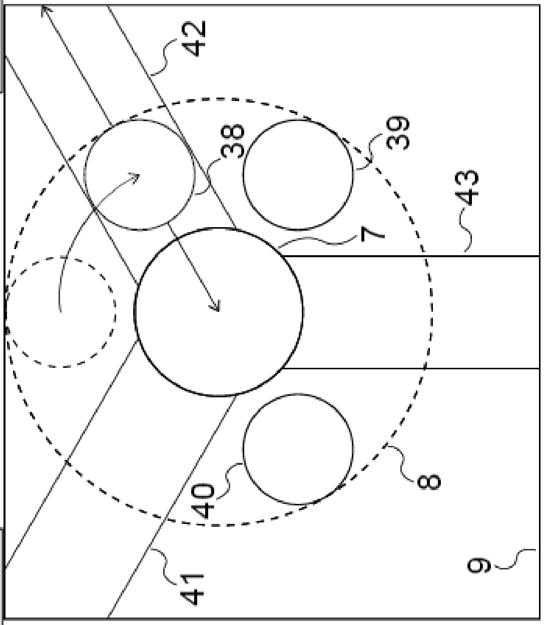


FIG. 16

FIG. 17

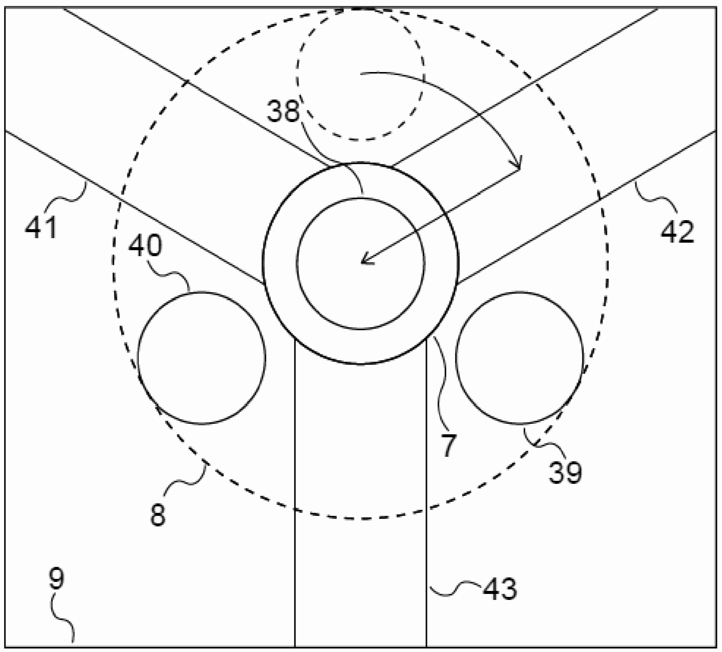


FIG. 18

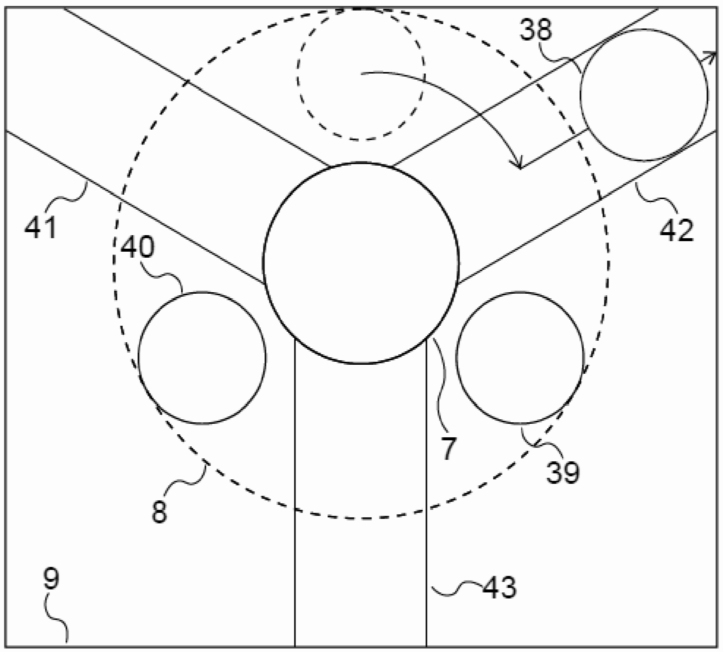


FIG. 19

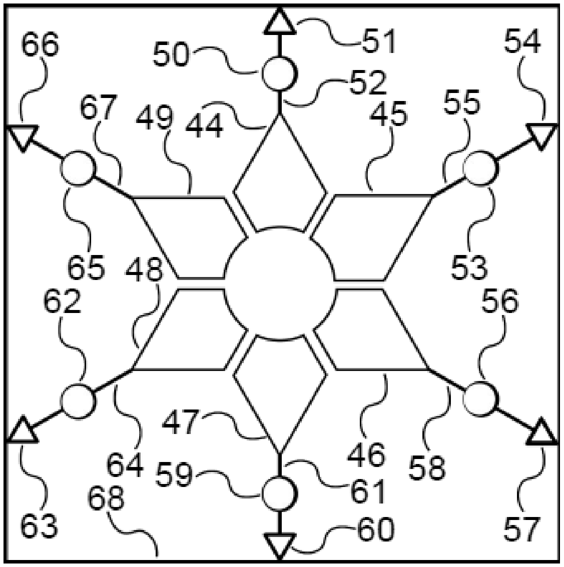


FIG. 20

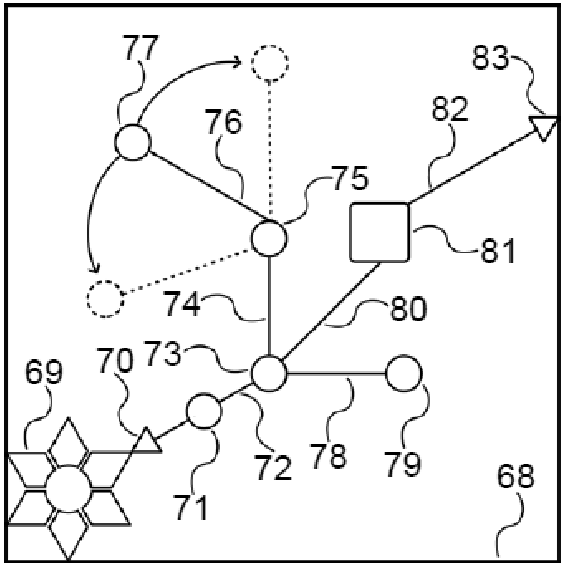


FIG. 21

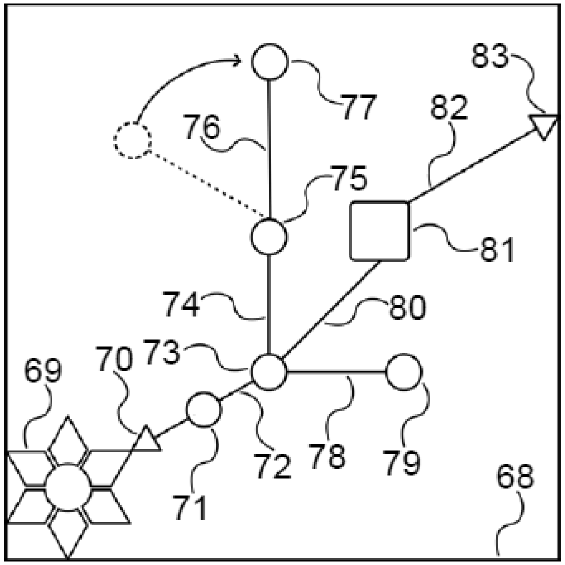
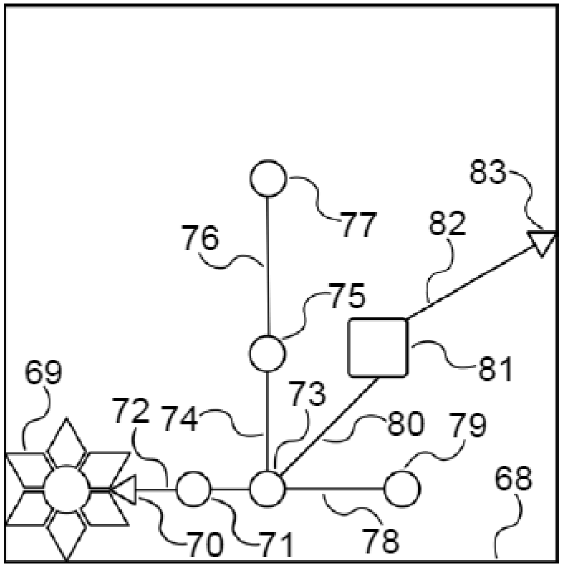


FIG. 22



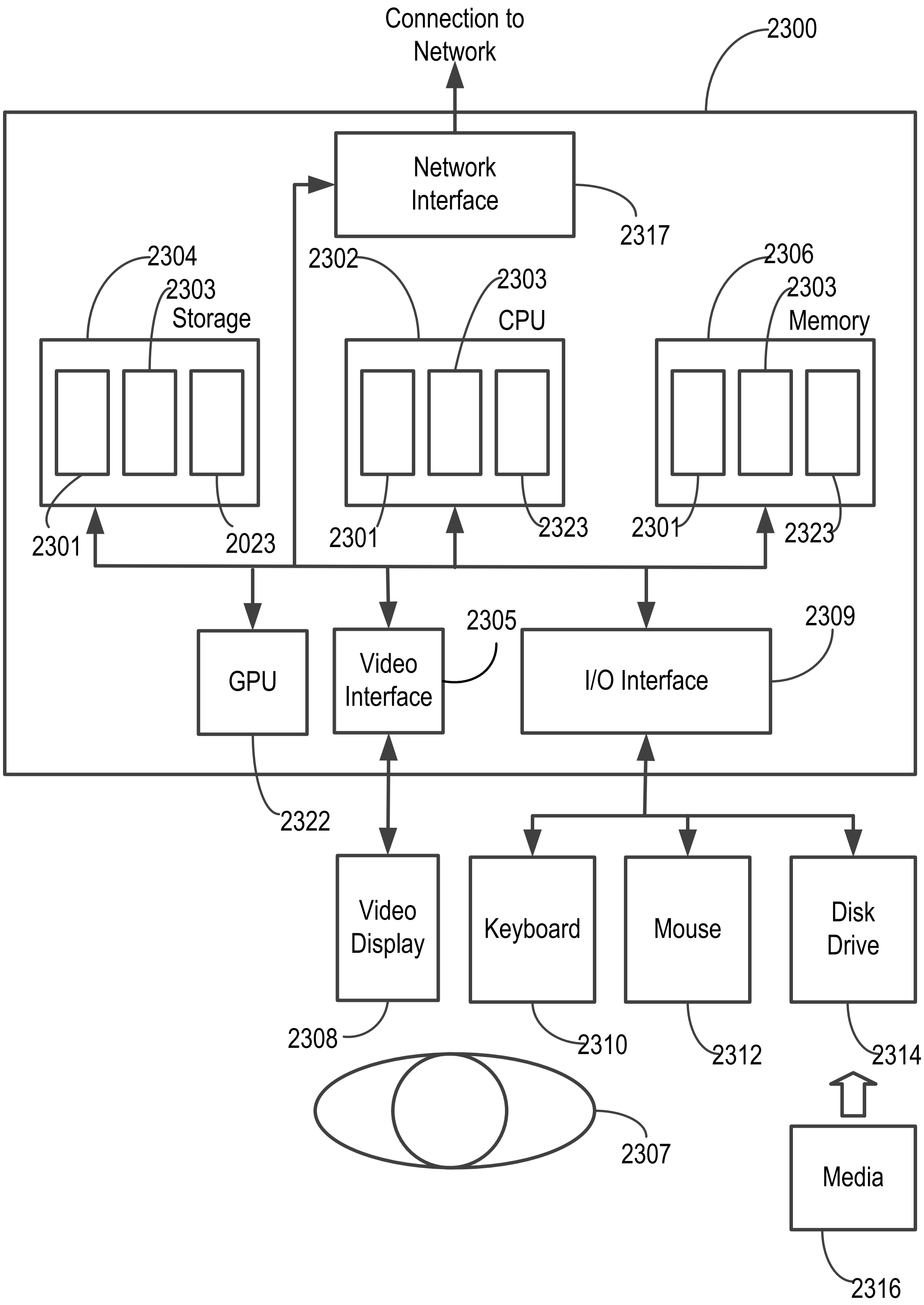

Connection to Network
2300
Network Interface
2317
2304
2303
Storage
2302
2303
CPU
2306
2303
Memory
2301
2023
2301
2323
2301
2323
GPU
Video Interface
2305
I/O Interface
2309
2322
Video Display
Keyboard
Mouse
Disk Drive
2308
2310
2312
2314
2307
Media
2316

FIG. 23