

TITLE

**SYSTEM FOR ORGANIZING AND DISPLAYING
INFORMATION ON A DISPLAY DEVICE**

CROSS-REFERENCE TO RELATED APPLICATION

[0001] This application claims the benefit of U.S. Provisional Application No. 61/762,165, filed on February 7, 2013.

BACKGROUND OF THE INVENTION

Field of the Invention

[0002] The present application relates to a system for organizing and displaying information on a display device. More particularly, an aspect of the present invention relates to systems, methods, and computer program products for organizing and displaying information on a display, on which a shown object can be manipulated using a pointing device like a mouse, a touch-pad, or a physical pointer, such as a stylus or a user's finger, by gestures executed, and/or by voice commands given by the user.

DESCRIPTION OF RELATED ART

[0003] Presently, computer systems are being used to perform an ever-increasing variety of functions that were traditionally provided by tangible items. For example, computer systems are now being used

to convey information, such as in books, newspapers, and maps, which were traditionally provided in a paper format. Computer systems also are being used to enable users to enjoy multimedia, such as photographs, music, and videos. Home control features, such as heating control, and remote control of televisions, light switches, alarms, doorbells, and the like, are being performed with the assistance of computer systems. Computer systems also have revolutionized the personal communication and data communications industries.

[0004] Given the growing use of computer systems in providing information to users, it would be beneficial to have a sophisticated means of organizing and/or displaying such information that can cover a wide range of needs, and be platform independent, easy to use, and easy to expand.

SUMMARY OF THE INVENTION

[0005] The example embodiments herein provide systems, methods, and computer program products for organizing, displaying, and interacting with information on a display device. In accordance with one example aspect herein, a system includes a computer processor and a memory device. The memory device stores at least one piece of computer code executable by the computer processor as well as data used by the computer code. A display device is structured to display a graphical interface to a user based on the computer code executed by the computer processor. One or more input devices are structured to receive information from the user in most cases based on one or more images of the graphical interface displayed on the display device. The computer code includes a main display module for providing a main display area of the graphical interface, and for organizing digital objects in a plurality of layers. The layers include a base layer corresponding to a base-surface situated in a window shown on the main display or covering its entire extent, and a fixed layer corresponding to a fixed-surface in a window on the main display or its entire extent.

[0006] Further features and advantages, as well as the structure and operation, of various example embodiments of the present invention are described in detail below with reference to the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0007] The features and advantages of the example embodiments presented herein will become more apparent from the detailed description set forth below when taken in conjunction with the drawings.

[0008] FIG. 1 shows an example arrangement of various components of a system for organizing, storing, synchronizing and displaying information on a display device, according to an example embodiment herein.

[0009] FIG. 2 is a block diagram of a computer for use with various example embodiments herein.

[0010] FIG. 3 illustrates exemplary functional modules that may be included in a memory device and used for organizing, displaying and/or manipulating information on a display surface, according to various example embodiments herein.

[0011] FIG. 4 illustrates some digital objects that have been moved and arranged on the base-surface for parallel access.

[0012] FIG. 5 illustrates zooming of an object that has been enlarged (zoomed) relative to a previous size, which is shown in a calculator object 401 of FIG. 4, according to an example embodiment herein.

[0013] FIG. 6 illustrates an object that has been rotated on the base-surface, according to an example embodiment herein.

[0014] FIG. 7 illustrates docking of two objects, according to an example embodiment herein.

[0015] FIG. 8 illustrates docking of three objects, according to an example embodiment herein.

[0016] FIG. 9 illustrates naming of a digital object using a settings menu and a search input area, according to an example embodiment herein.

[0017] FIG. 10 illustrates searching using a search input area, according to an example embodiment herein.

[0018] FIG. 11 illustrates zooming using a zoom slider, according to an example embodiment herein.

[0019] FIG. 12 is a flowchart showing an example procedure for organizing, displaying and/or manipulating information on a display surface, according to various example embodiments herein.

[0020] FIG. 13 illustrates overlaying of multiple digital objects, according to an example embodiment herein.

DETAILED DESCRIPTION

[0021] The present invention relates to systems, methods, and computer program products for organizing and displaying information on a display device. The display surface may be a touch-sensitive display surface on which a displayed object can be manipulated using a pointing device like a mouse, a touch-pad, a physical pointer, such as a stylus or a user's finger, or by gestures executed or voice commands given by the user. Optionally, a display surface of the system may be a standard electronic display monitor, a wearable display device like glasses or lenses projecting an image onto the eye of the user, or an image projected onto any kind of surface on which a displayed object can be manipulated using an electronic pointing device, such as a mouse, a touch-pad, a stylus, a user's finger, gestures, voice commands, or the like. The electronic display monitor may be a computer screen, a television monitor, a tablet device, an interactive table or frame, a wearable display device, an image projection on any surface or the like.

[0022] The term "viewer" may be used herein to refer to a software portion of a system that enables the user to interact with one or more digital objects.

[0023] The term "data store" may be used herein to refer to a software portion of a system that stores data for digital objects. A data store may reside on a display device, on a storage device that is running within a user's premises, or on a storage device that is running remotely.

[0024] The term "display device" may be used herein to refer to hardware on which a copy of viewer software is running. A display device may or may not contain local data store software.

[0025] The term “storage device” may be used herein to refer to hardware on which a copy of data store software is running.

[0026] The term “viewer window” may be used herein to refer to a window provided by viewer software via a display device to enable the user to interact with one or more digital objects.

[0027] The term “code bundle” may be used herein to refer to executable source code and/or configuration data that is utilized to instantiate a digital object in a viewer.

[0028] The term “digital object” may be used herein to refer to an instance of a code bundle that represents a functional entity having a corresponding set of data stored in a portion of a data tree. Multiple digital objects may be controlled such that they are synchronized with one another.

[0029] The term “data tree” may be used herein to refer to a data structure and/or methods provided by a data store to store and deliver data for digital objects.

[0030] The term “digital object store” may be used herein to refer to a portion of a system that stores and delivers additional code bundles.

[0031] FIG. 1 shows an example arrangement of various components of a system 10 for organizing and displaying information on one or more display devices, in accordance with an example embodiment herein. The system 10 includes display devices 103, 104 on which viewer software 102, 105 is executed. The viewer software allows users to interact with one or more digital objects 100, which represent an instance of a code bundle, as described in further detail below. A code bundle represents a functional entity with its own set of data stored in a portion of a data tree. In particular, a code bundle includes executable source code and configuration data that, in some cases, is required to instantiate a digital object 100 in a viewer software 102, 105. Although not shown in FIG. 1 for purposes of convenience, the data trees are structures and methods that are stored in data stores 101 and that store and provide data for digital objects 100. The system 10 also includes storage devices 106, 108 on which various ones of the data stores 101 are replicated.

[0032] In order to handle multiple digital objects arranged on a single base layer of a device, a rendering process of the digital objects is optimized by using caching algorithms. Whenever a digital object is to be drawn, for example, because there is new data to show in connection with the object, the new content will be drawn as a bitmap in an invisible buffer, using a current zoom level. The bitmap will be cached (e.g., stored in memory) and used to draw the digital object when required, instead of rendering content over and over again when the object has to be redrawn. Additionally, if a digital object is not visible within the main drawing area or the object has been iconized before or during a zoom operation, then the object will not be rendered. Rather, a previously cached bitmap will be used to draw the object when required. Later, if the system is idle, then objects will be re-rendered if required to match the resolution or zoom level of the display surface.

[0033] In accordance with another example embodiment, data that is required in an object at a later time or a different place is made persistent by being temporarily stored in a large internal data tree. Each

object has a corresponding key that the object provides to the data tree in order to be granted access to store data therein and/or retrieve data therefrom.

[0034] Various portions of the data tree are available to a digital object. A first, private portion of the data tree is available only to a particular instance of an application (e.g., a digital object). The private portion of the data tree is used to store and access data that is not intended to be shared with other objects. A second, public portion of the data tree exists for each digital object, which the digital object can use in order to exchange data with other digital objects. For example, the public portion of the data tree can be used to exchange data between docked objects (explained further below). A third portion of the data tree is available separately for each digital object. The third portion stores information about the corresponding object's size, color, rotation, position, and other attributes. A fourth portion of the data tree, referred to as a local tree, makes data available to all digital objects that are running on a particular display surface. The fourth portion of the data tree is used to share data among all objects running in one location. A fifth portion of the data tree makes all attributes of the main display area (e.g., size, color, rotation, position, etc.) available to all local digital objects.

[0035] In one example embodiment, when an instance of a digital object is removed (e.g., un-instantiated) from the system, the private portion and the public portion of the data tree that were allocated for the object are removed from the system. In this way, valuable memory space is conserved by avoiding memory leaks that would otherwise consume memory space for objects that are no longer instantiated.

[0036] Because portions of the system supplying data trees are independent of a display surface and digital objects are allowed to easily access data trees from the display surface they are located on, the system allows for efficient synchronization of digital objects distributed across multiple display surfaces and/or locations. Data stores are the portions of the system that store and provide the data trees, and may reside on any device connected to any network that can be accessed from one or more display services. The system also allows for a portion of a data tree to be linked to a remote source in addition to keeping a copy of the data tree in a local cache, for example, for redundancy purposes.

[0037] FIG. 2 is a block diagram of a general and/or special purpose computer system 200 that may be employed in accordance with some of the example embodiments herein. The computer system 200 may be, for example, a user device, a user computer, a client computer and/or a server computer, among other things.

[0038] The computer system 200 may include, without limitation, a computer processor 201, a main memory 202, and an interconnect bus 203. The computer processor 201 may include without limitation a single microprocessor, or may include a plurality of microprocessors for configuring the computer system 200 as a multi-processor system. The main memory 202 stores, among other things, instructions and/or data for execution by the processor device 201. The main memory 202 may include banks of dynamic random access memory (DRAM), as well as cache memory.

[0039] The computer system 200 may further include mass storage device(s) 204, peripheral device(s) 205, input control device(s) 206, portable storage medium device(s) 207, graphics subsystem(s) 208, and/or one or more output display(s) 209. For explanatory purposes, all components in the computer system 200 are shown in FIG. 2 as being coupled via the bus 203. However, the computer system 200 is not so limited. Devices of the computer system 200 may be coupled via one or more data-transport devices known in the art. For example, the computer processor 201 and/or the main memory 202 may be coupled via a local microprocessor bus. The mass storage device(s) 204, the peripheral device(s) 205, the portable storage medium device(s) 207, and/or the graphics subsystem(s) 208 may be coupled via one or more input/output (I/O) buses. The mass storage device(s) 204 may be nonvolatile storage device(s) for storing data and/or instructions for use by the computer processor 201. The mass storage device(s) 204 may be implemented, for example, with one or more magnetic disk drive(s), solid state disk drive(s), and/or optical disk drive(s). In a software-related embodiment, at least one mass storage device 204 is configured for loading contents of the mass storage device 204 into the main memory 202.

[0040] Each portable storage medium device 207 operates in conjunction with a nonvolatile portable storage medium, such as, for example, a compact disc with a read-only memory (CD-ROM) or a non-volatile storage chip (Flash), to input and output data and code to and from the computer system 200. In some embodiments, the software for storing an internal identifier in metadata may be stored on a portable storage medium, and may be inputted into the computer system 200 via the portable storage medium device 207. The peripheral device(s) 205 may include any type of computer support device, such as, for example, an input/output (I/O) interface configured to add additional functionality to the computer system 200. For example, the peripheral device(s) 205 may include a network interface card for interfacing the computer system 200 with a network 210.

[0041] The input control device(s) 206 provide among other things, a portion of the user interface for a user of the computer system 200. The input control device(s) 206 may include a keypad, a cursor control device, a touch sensitive surface coupled with the output display(s) 209 or standalone, a camera, a microphone, infrared sensors, knobs, buttons, and the like. The keypad may be configured for inputting alphanumeric characters and/or other key information. The cursor control device may include, for example, a mouse, a trackball, a stylus, and/or cursor direction keys. In order to display textual and graphical information, the computer system 200 may utilize the graphics subsystem(s) 208 and the output display(s) 209. The output display(s) 209 may include a cathode ray tube (CRT) display, a liquid crystal display (LCD), a projector device, and the like. Each graphics subsystem 208 receives textual and graphical information, and processes the information for output to at least one of the output display(s) 209.

[0042] Each component of the computer system 200 may represent a broad category of a computer component of a general and/or special purpose computer. Components of the computer system 200 are not limited to the specific implementations provided here.

[0043] Portions of the example embodiments of the invention may be conveniently implemented by using a conventional general purpose computer, a specialized digital computer, and/or a microprocessor programmed according to the teachings of the present disclosure, as is apparent to those skilled in the computer art. Appropriate software coding may readily be prepared by skilled programmers based on the teachings of the present disclosure.

[0044] Some embodiments may also be implemented by the preparation of application-specific integrated circuits, field programmable gate arrays, or by interconnecting an appropriate network of conventional component circuits.

[0045] Some embodiments include a computer program product. The computer program product may be a storage medium or media having instructions stored thereon or therein, which can be used to control, or cause, a computer to perform any of the procedures of the example embodiments of the invention. The storage medium may include without limitation a floppy disk, a mini disk, an optical disc, a Blu-ray Disc™, a DVD, a CD-ROM, a micro drive, a magneto-optical disk, a ROM, a RAM, an EPROM, an EEPROM, a DRAM, a VRAM, a flash memory, a flash card, a magnetic card, an optical card, nanosystems, a molecular memory integrated circuit, a RAID, remote data storage/archive/warehousing, and/or any other type of device suitable for storing instructions and/or data.

[0046] Stored on any one of the computer-readable medium or media, some implementations include software for controlling both the hardware of the general and/or special computer or microprocessor, and for enabling the computer or microprocessor to interact with a human user or other mechanism utilizing the results of the example embodiments of the invention. Such software may include, without limitation, device drivers, operating systems, and user applications. Additionally, such computer readable media further includes software for performing example aspects of the invention, as described herein.

[0047] Included in the programming and/or software of the general and/or special purpose computer or microprocessor are software modules for implementing the procedures described herein.

[0048] Having described a general and/or special purpose computer that may be employed in accordance with some of the example embodiments herein, reference will now be made to FIG. 3, which illustrates example functional modules that may be included in a memory device 301, in accordance with example embodiments herein. In some example embodiments, the memory device 301 is included in the computer system 200, described above in connection with FIG. 2, further represents the main memory 202 in part or in whole, and is used for organizing, displaying, and/or manipulating information on a display device or surface. For example, although not shown in FIG. 3 for purposes of convenience, the memory device 301 is coupled to a computer processor (e.g., the computer processor 201) that, in turn, is coupled to one or more display surfaces (e.g., the output display(s) 209) and optionally to one or more capabilities to exchange data over a network (e.g., the network 210) or on dedicated connections. In one example embodiment, each display device 209 is structured to display a graphical interface (e.g.,

a GUI) to a user based on computer code (e.g., modules 302 through 310) executed by the computer processor 201. An input device (e.g., the input control device 206) is structured to receive information from the user based on one or more images of the GUI displayed on the display device 209.

[0049] As shown in FIG. 3, the modules stored within the memory device 301 include a main display module 302, a zoom module 303, a rotation module 304, a container module 305, a docking module 306, a management module 307, an authentication module 308, a digital object store module 309, and an overlay module 310. As will be described in further detail below, each of the modules 302 through 310 includes computer-executable code that imparts functionality to the computer system 200 when executed by the computer processor 201 as well as data related to that code. Additionally, the memory device 301 stores computer programs and data for applications that a user may interact with via display surface(s).

[0050] The main display module 302 provides a main display area of the graphical interface for the display surface. The main display area may cover all or a part of the display surface. The main display module 302 organizes digital objects on the main display area. Each digital object corresponds to a running interactive software application. The digital objects can represent at least one of: (1) a (collaborative) document (which may contain any one or a combination of text, one or more images, one or more videos, and/or one or more animations), (2) a media player (for playing music content, video content, and/or streaming media data), (3) a home appliance or device controller, (4) a game, (5) a navigation tool for revealing one or more particular portions of the main display area, (6) a social networking tool for providing access to a social network, (7) a reference tool, such as a dictionary, thesaurus and/or an encyclopedia, (8) a container for associating a plurality of digital objects into a group, for enabling the group to be displayed as an icon or in full size and function, and/or for connecting and synchronizing the group with a remote display device (9) a spreadsheet, (10) a calculator, (11) a web page being provided by an Internet website, (12) a photo album, (13) a camera, (14) virtual TV set, (15) a newspaper/newsfeed, (16) a book, (17) an e-mail client, (18) a slideshow display, (19) a door opener button, (20) a to-do list, (21) a drawing and sketching pad, (22) a text message sending element, (23) an element that shows detailed information about the software running, (24) a map, (25) a product catalog, (26) an element to search and browse images from the Internet, (27) a form for, for example, providing feedback about a subject, (28) a poll result from votes coming from an audience, (29) a simple visual programming element, (30) a recipe collection, (31) an address book, (32) a calendar, (33) a diary, (34) a time-table for, for example, for public transportation, (35) a phone directory, (36) a language translator, (37) a barcode/QR-code display, (38) an element to search and instantiate other digital objects, (39) an element to browse through and instantiate digital objects from a remote site, (40) an element enabling the user to control a general remote device, and (41) an element to receive and display data from a general remote device. A digital object appears in the main display area in an iconized state or a fully displayed state, and the main display module 302 enables the user to change the digital object to and from the iconized state and the fully displayed state. The main display module 302 enables the user to change a position of a digital object in the main display area (e.g., by

providing a dragging input with a mouse, stylus, finger, or the like) without changing a position of another digital object in the main display area. When the digital object appears in the fully displayed state in the main display area, the digital object may be locked to prevent it to be moved, rotated, renamed, colored, or zoomed.

[0051] The main display module 302 organizes the digital objects on the main display area in a plurality of layers, including a base layer and a fixed layer. Two or more digital objects may be synchronized with each other, such that manipulation of one affects the appearance and/or operation of the other(s). The base layer corresponds to a base-surface of the main display area. An appearance of one or more base-layer objects on the base-surface may be selectively altered by the user. The fixed layer corresponds to a fixed-surface within the main display area. In other words, the fixed layer can appear to be fixed and floating above the base layer within the main display area. The fixed layer allows the user to have certain objects arranged on it separately. An appearance of one or more fixed-layer objects on the fixed-surface is fixed or pinned when an appearance of a base-layer object is altered by the user. In other words, objects arranged on the fixed layer are not zoomed, rotated, or moved when the base layer changes zoom level, rotation, or position. In this way, the objects arranged on the fixed layer are available to the user independent of the base layer.

[0052] The user has the flexibility to select which objects are arranged on the fixed layer, if any at all. The main display module 302 enables the user to (1) selectively pin and unpin a digital object to the fixed-surface window as a fixed-layer object, (2) set a digital object as a base-layer object displayed on the basic-layer, (3) change a base-layer object to a fixed-layer object, (4) change a fixed-layer object to a base-layer object, and (5) create a group of base-layer objects, such that an appearance of the group of base-layer objects may be altered as a group.

[0053] In one example aspect, the main display module 302 also enables the user to (1) move, zoom, and rotate a base-layer object on the base-surface; (2) move, zoom, and rotate the base-surface within the main display area relative to a position of the fixed-surface window; and/or (3) selectively set a position of the fixed-surface window within the main display area.

[0054] Additionally, the user may set a color or define a high resolution image as a background for a digital object, rename it, control its transparency and, if set by the developer of the object, change the side ratio of the digital object.

[0055] Additionally, the base layer, including all objects it carries, may also be rotated and zoomed independently of an orientation of the display surface. The part of the base layer visible within the main display area can also be controlled by moving this layer relative to a rectangle of the main display area. It is also possible to set a color or define a high resolution image as a background for the base layer.

[0056] In another example aspect herein, the main display module 302 enables the user to create a first group of base-layer objects different from a second group of base-layer objects, such that an appearance of the first group of base-layer objects may be altered in unison without altering an appearance of the second group of base-layer objects.

[0057] In a further example embodiment, the computer system 200 further includes a communication interface for connecting the computer processor 201 to at least one of (1) an apparatus connected to the computer system 200 via a dedicated communication line, to allow the computer system 200 to receive information from and send information to the apparatus; (2) a local area network (e.g., the network 210), to allow the computer system 200 to receive information from and send information to one or more other systems connected to the local area network; and/or (3) an Internet service provider, to allow the computer system 200 to receive information from and send information to an Internet address. The communication interface is configured to perform at least one of wireless communication and/or wired communication.

[0058] According to another example aspect, the input device 206 includes at least one of (1) a touch-sensitive sensor arrangement structured to receive pressure produced by the user on the display device, (2) a sound and voice receiver structured to receive sounds and/or commands produced by the user, (3) an electronic pointing device structured to be manipulated by the user to provide commands based on a location of a cursor on the display device, and (4) one or more cameras to recognize gestures, mimics, and moves a user may produce in front of it/them as well as other optical information like infrared signals and/or brightness information.

[0059] In a further aspect, the main memory 202 stores code bundles, each code bundle including executable code and configuration data for instantiating a corresponding digital object on the display device. Code bundles are small directory structures containing several code units, media resources like videos, images, sounds, and the like, and structured information about the code bundle. The directory structure of a code bundle may be provided as a compressed file and may optionally be encrypted to bind it to a particular display device or a particular user.

[0060] In some example embodiments, the display device can be at least one of (1) a touch-sensitive display surface that receives user input by sensing physical contact, and/or (2) an electronic display monitor that receives user input via at least one of a pointing device, gestures or mimics, and/or a voice command or any other signal delivered to one of the input control device(s) 206.

[0061] In a further example embodiment, a digital object may be moved, rotated, or zoomed by the user via at least one of (1) a swipe motion executed on the touch-sensitive display surface, (2) a tap on the touch-sensitive display surface, (3) a drag motion of the pointing device, (4) a click of the pointing device, (5) a spoken command, and/or one or more gestures executed by the user. When the digital object is moved, rotated, or zoomed, a visible appearance of the digital object transitions smoothly from an initial appearance to a final appearance.

[0062] In one example embodiment, the zoom module 303 enables the user to alter a size of at least one of: (1) a base-layer object, (2) a group of base-layer objects, (3) all base-layer objects, and/or (4) a viewable area of the base-surface window, displayed on the display device. For example, FIG. 5 shows a calculator object 501 that has been enlarged (zoomed) relative to a previous size, which is shown in the calculator object 401 of FIG. 4. By enabling the size of various objects to be altered, the zoom module

303 enables large quantities of objects to be arranged and organized within a limited amount of space available on a physical display surface, while still enabling the user to navigate from one object to another smoothly and with minimal effort.

[0063] The zoom module 303 is configured to provide various ways to control the zoom level as well as the visible area of the base level. In one example embodiment, the zoom module 303 is configured to provide at least one of (1) one-command zooming, such that a single command issued by the user and received by the input device causes zooming to a predetermined size and a predetermined position; (2) a sliding zoom scale, which enables the user to change the size smoothly and continuously in accordance with a slide position of the sliding zoom scale, the slide position being smoothly and continuously changeable by the user via the input device; and (3) a zoom-level changer, which enables the user to smoothly and incrementally change the zoom level .

[0064] One example procedure for zooming in and out of the base layer is by executing a double-click (e.g., on a mouse) or a double-tap using a finger or a stylus at any position on the surface, and remaining in a clicked position or a tapped position for at least a predetermined amount of time after the double-click or double-tap. This will cause a zoom slider to appear at this particular position (e.g., 1103 in FIG. 11) with the control button right at click location. In this case the center of the zoom will be the point at which the user started the interaction

[0065] In another example embodiment, the zoom module 303 enables the user to cause the sliding zoom scale to appear and disappear from the main display area by at least one of (1) a tap input, (2) a swipe input, (3) a click input, and/or (4) a voice-command input. When the sliding zoom scale appears in the main display area, the slide position of the sliding zoom scale is at a position corresponding to a current size of the base-surface window. The zoom module 303 enables the user to cause the sliding zoom scale to appear and disappear from the main display area by a tap input or a click input. A position of the tap input or the click input can be used to designate a position at which the sliding zoom scale appears in the main display area. Optionally, the zoom module 303 is configured to display the sliding zoom scale at a predetermined position in the main display area. For example, if the object selected is not yet centered in the main display area and not zoomed to fit the area in an optimal way, the zoom module 303 can shift and zoom the base layer to center and zoom the object in the main display area to fit in an optimal way. If the base layer has been shifted and zoomed to this state before, another double-click or double-tap causes the position and zoom level to revert back to their state before the first double click. In another example aspect, zooming in on an object also causes the entire base layer to automatically rotate in a way that makes the object appear in an upright position. Another double-click or double-tap causes the rotation to revert back to its state before the first double-click and back to the original orientation of the base layer at the second click. The zoom module 303 enables the user to change the zoom level by at least one of (1) a tap input, (2) a click input, (3) a gesture input, and (4) a voice-command input.

[0066] As shown in FIG. 11 the zoom module 303 enables the user to tap or click an overview button 1101 at the bottom of the display area. This will cause the zoom module 303 to zoom, rotate and pan the base layer to show all digital objects placed on it at the same time at the largest zoom level.

[0067] Additionally if the user presses the overview button 1101 for more than a normal click/tap, the zoom module 303 to display a zoom slider 1102 at this particular point. In this case the center of the zoom will be the center of the visible part of the base surface.

[0068] The rotation module 304 enables the user to alter a rotational orientation of at least one of (1) a base-layer object, (2) a group of base-layer objects, (3) all base-layer objects, and (4) a viewable area of the base-surface, displayed on the display device. In one example embodiment, the rotation module 304 enables the user to change a rotational orientation of a base-layer object to an arbitrary angle without changing a rotational orientation of another base-layer object. For example, FIG. 6 shows a calculator object 601 that has been rotated within a base-surface window 602. The rotation module 304 also enables the user to change a rotational orientation of a group of base-layer objects to an arbitrary angle without changing a rotational orientation of the viewable area of the base-surface window.

[0069] In one example embodiment, the user is enabled to rotate and zoom a digital object in one coherent movement. This is achieved by combining zooming and rotating of digital objects as provided by the zoom module 303 and the rotation module 304 into one specific gesture that may be allocated for example to an area on a corner of every digital object. Clicking or touching this area with a pointing device like for example a stylus, a mouse, or a user's finger on a touch surface and dragging it from there, allows to zoom and rotate a digital object in one movement. The position opposite of the dragging area of the digital object being zoomed and rotated is used as an anchor point for this particular gesture. The gesture described here is different in purpose and function from the one used to resize (not zoom) the content area and the ratio of height and width of a digital object as offered for objects on screens of traditional systems. It also differs from a gesture often used in traditional systems that needs two points on screen to be selected in order to zoom and/or rotate an object. Reducing the number of touch points from two to one allows the same gesture to be used for zooming and rotation with a user's finger on a touch surface as well as with a pointing device like a stylus or a mouse.

[0070] In one example embodiment, the user is enabled to navigate (manually or automatically) through several configurations predetermined by the user, each configuration having a predetermined zoom level and a predetermined rotation for one or more digital objects. The user may manually navigate through the configurations by providing an input via a digital navigator object 402, remote control, a spoken command (e.g., "next," "previous," etc.), a gesture, and/or the like. Or the configurations may be automatically navigated through in response to an application running on a different device issuing one or more commands to the computer system 200.

[0071] One type of digital object is referred to as a container 1001. A container is an application that provides an area on which other objects can be arranged to enable them to be manipulated as a group. Containers are handled by the container module 305, which enables the user to associate a plurality of

digital objects into a group (see, for example, FIG. 7), such that a change in size, rotation, or position of the group causes a corresponding change in size, rotation, or position of each of the plurality of digital objects in the group. The user may designate a group of digital objects to be a hierarchy (discussed below), and give the container a name of his choice. When the user subsequently selects to utilize the hierarchy, a single selection of the user-given name causes all of the objects of the hierarchy to be launched for use.

[0072] A container 1001 may also be configured to represent another display device accessible to the user. This allows the simple exchange of digital objects between the local display device and the remote device by dragging object from the local surface to the container configured to represent the remote device and vice-versa.

[0073] In another example embodiment, an Internet browser object is provided that enables a user to access the Internet from within the computer system 200. In this way, one or more live views of a website can be used interactively, or as an integrated part of a presentation on stage, in one example.

[0074] The docking module 306 enables the user to dock a first digital object with a second digital object to form first and second docked digital objects that are operatively linked together. The first and second digital objects correspond to first and second interactive software applications, respectively. Data produced by the first interactive software program corresponding to the docked first digital object is used by the second interactive software applications corresponding to the second digital object to produce a combined output. For example, when a room temperature object is docked with a general unit conversion application, the user can select to have temperatures, which may be presented in degrees Fahrenheit as the default unit, automatically converted to a desired unit (e.g., Kelvin, Celsius, etc.).

[0075] According to one example aspect, illustrated in FIG. 7, a first digital object 701 is docked with a second digital object 702 based on a touching position 703 of the first digital object relative to the second digital object in the base-surface window (e.g., aligning the edges of the first and second digital objects). In an example embodiment, if the first and second digital objects are able to communicate, the user will be presented with visual feedback when the two digital objects are positioned near each other. For example, FIG. 7 shows a container with a room temperature object 701 docked with a unit converter object 702. FIG. 8 shows a container with a calculator object 801 docked with a unit converter object 802 and a room temperature object 803.

[0076] In another example aspect, the first digital object is undocked from the second digital object based on a spaced-apart position of the first digital object relative to the second digital object in the base-surface window.

[0077] The management module 307 controls and manages the flow of data between the digital objects 100 in FIG. 1 and the local storage and caching facilities 101 provided for the viewer software 102 on local display devices 103 that do offer local storage. An optional local storage device 108 providing an additional data storage 109 may be used to synchronize data between the local display devices 103

within the user's premises. Without the local storage device 108, the viewer software 102 could as well use the remote storage device 106.

[0078] In another option, if the management module 307 of the viewer software 105 on a display device 104 is not able to provide local data storage to store digital object data persistently, it may use a local storage device 108 located on the user's premises or may store the data on a storage device 106 that is running remotely and accessed via the Internet. In both cases, the storage devices 106, 108 are considered to be part of the system 10 and are, based on their data stores 107, 109, providing the same set of services to the viewer software 102, 105 as the set of services that can be provided by the data stores 101.

[0079] The management module 307 allows the user to manage a repository of digital objects selectable for use, either on a local system or a data store accessible by the viewer.

[0080] In particular, the management module 307 includes a main digital object 403 (see FIG. 4) that manages access by the user to the repository of previously installed code bundles. The main digital object 403 providing this access may be hidden or revealed in the main display area in accordance with a command inputted by the user via one or more input buttons 404. The main digital object 403 is a user interface for an interactive software application that provides a menu of the digital objects in the repository selectable by the user to be a base-layer object. The main digital object 403 also provides a menu of hierarchies of digital objects in the repository selectable by the user. Digital objects belonging to a hierarchy share a common characteristic, such that a selection of a hierarchy from the menu results in a submenu of digital objects belonging to the selected hierarchy to be provided for selection by the user to be a base-layer object.

[0081] The main digital object also provides a search input area 405 for the user to input a search term to search for a digital object in the repository.

[0082] A digital object of the repository selected by the user via the main digital object 403 causes a copy of the digital object to be instantiated in an object window within the base-surface window. The main digital object 403 also enables multiple copies of a digital object of the repository selected by the user to appear in multiple object windows within the base-surface window.

[0083] The authentication module 308 controls and manages access to a user's digital objects and their data based on any kind of user authentication, such as (1) entering a username and a password via the viewer, (2) any biometric identification like, for example, a fingerprint reader, voice recognition, face recognition, and/or an iris scan, and/or (3) any kind of token like, for example, a secure card, an RFID token, a USB stick with a key file, and/or a secure exchange of such a user token over any kind of connection from a personal device to the display device 103 running the viewer software 102 of the viewer.

[0084] Once the user has been identified by the viewer software, the authentication module 308 will monitor any access of the management module 307 to digital objects and their data in a local data store

101 as well as access to remote data stores 109 and/or 107 and will refuse access to such objects and data if the user does not have the access rights necessary.

[0085] Having the authentication module 308 to control access to a user's digital objects and their data allows sharing of the same display device 103 among multiple users while still ensuring privacy and confidentiality of personal digital objects. This is particularly relevant for display devices in public places like restaurants, hotels, bus stops, etc., but can also be relevant in an office where users may share desks or in a household where it is not intended that all members of the family have access to the same digital objects and their data.

[0086] The digital object store module 309 provides access to a separate external server infrastructure called the digital object store in order to search for and add code bundles of new, previously not installed digital objects. The module 309 provides metadata about available code bundles like, for example, a description of the functionality, ratings and comments by other users, the author of the code, the code version, prerequisites to execute the code bundle, the price to buy it, etc. A user's access to this source of additional code bundles will usually take place in a digital object specifically designed for this purpose. In order to instantiate a new digital object that is not yet available as a local code bundle, the digital object store module 309 will first download the necessary code bundle, store it in an available data store and then instantiate the digital object requested by the user using the management module 307.

[0087] The digital object store module 309 is also responsible for regularly scanning local code bundles for outdated versions, and downloading and installing new versions of such bundles as well as re-instantiating existing digital objects based on the new code bundle supplied.

[0088] Another function of the digital object store module 309 is the handling of encrypting and decrypting code bundles. As code bundles used in the computer system 200 may be stored in readable form as source code, the digital object store module 309 will ensure integrity of the code and protect it from being copied without permission by keeping the code in encrypted form and decrypting it only to make it executable within the computer system's main memory 202.

[0089] The overlay module 310 enables a user to arrange a first digital object upon a second digital object (i.e., to overlay the first digital object upon the second digital object such that at least a portion of the first digital object overlaps with at least a portion of the second digital object) to enable both the first digital object and the second digital object to be utilized and/or manipulated as a group, for example, in a manner similar to the manner described above in connection with the container module 305. In one example, one or more objects can be made partially or fully transparent, and can be overlaid upon one or more additional objects. This arrangement enables the functionalities of multiple objects to be combined and used in new ways. For example, as shown in FIG. 13, a sketching object 1302 may be overlaid upon a map object 1301 or a photo object. The user may then use the sketching object 1302 to manually sketch or draw additional information, such as a route or a proposed change to the image, upon the map object 1301. In one example, by virtue of the positional relationship between the sketching object 1302

and the map object 1301, the combined output of the sketching object 1302 (e.g., the additional information that has been sketched or drawn) and the map object 1301 can convey more information than the sketching object 1302 or the map object 1301 might be able to convey by themselves.

[0090] In another example, as shown in FIG. 13, the overlay module 310 may enable the user to overlay upon a floor plan 1303 or a photograph one or more digital objects (e.g., a lamp control object 1304, a temperature control object 1305, and/or the like) that can be used to control one or more corresponding remote devices. For instance, the lamp object 1304 may be overlaid upon a portion of a floor plan that corresponds to a particular room in a dwelling, thereby enabling the user to interact with the lamp object 1304 to cause a lamp in that room to toggle on or off. In this way, based on the positions at which digital objects are overlaid relative to the floor plan or photo, those digital objects may be used to control lamps, loudspeakers, room temperature, or other corresponding remote devices in specific rooms.

[0091] In one example embodiment, the computer system 200 includes a plurality of display devices (e.g., the devices 209) or surfaces each structured to display a graphical interface to the user based on the computer code executed by the computer processor 201, wherein a digital object may be transferred from a first display device to a second display device. When the digital object is transferred from the first display device to the second display device, a visible appearance of the digital object moves smoothly from the first display device to the second display device. The digital object is transferred from the first display device to the second display device if the first display device electronically recognizes the second display device.

[0092] For example, a photo album object displaying a photo may be transferred from one display surface (e.g., a computer screen) to another display surface (e.g., a television monitor, a car display monitor, etc.). According to another example, a teacher may write a math problem on a teacher display surface (e.g., a large board at the front of a classroom) and transfer the math problem to multiple student display surfaces (e.g., tablets) so that the students can solve the math problem individually on their own display surface and return (e.g., electronically transfer) the solved problem back to the teacher.

[0093] In another example embodiment, the computer system 200 further includes a communication interface for wirelessly connecting the computer processor 201 to an external controller (not shown in FIG. 2), with the computer system 200 being portable by the user. When the computer system 200 enters a location recognized by the computer system 200 and controlled by the external controller, a signal received by the computer processor 201 from the external controller automatically initiates execution of a portion of the computer code to cause the display device to display one or more digital objects corresponding to the location.

[0094] In a further example embodiment, the system computer 200 further includes a sensor (not shown in FIG. 2) that senses a condition of an environment of the computer system 200, with the computer system 200 being portable by the user. When the sensor senses a predetermined condition of the environment of the computer system 200, the sensor provides a signal to the computer processor 201 to initiate execution of a portion of the computer code to cause the display device to display one or more

digital objects corresponding to the predetermined condition of the environment of the computer system 200. Example types of sensors include a camera, a microphone, a heartrate sensor, a radio frequency identifier (RFID) reader, a bar code scanner, a humidity sensor, etc. Example types of predetermined conditions that may be sensed by the sensor include a predetermined time of day or night, a location of the computer system 200, movement of a physical object being present or moving in view of a camera, the presence of a sound signal, an RFID signal, or a bar code signal, a humidity level, a person entering an area near a display device or giving a spoken order, a current temperature level exceeding a predetermined threshold, etc.

[0095] In another example embodiment, the computer system 200 further includes a communication interface and/or wireless sensor (not shown in FIG. 2), such as a proximity sensor known in the art, that detects the spatial proximity of one or more additional computer systems 200 running the viewer software 102. In one example, the software 102 is configured so as to enable the two or more spatially proximate computer systems 200 to exchange information as they are brought close to one another. For instance, the display devices 103 can interact in a way that creates one virtual base-layer that spans across the two or more computer systems 200 recognized to be in spatial proximity with one another. This configuration enables the user to move digital objects 100 across display devices 103 as if they were one large display device.

[0096] In accordance with one example embodiment, a digital object can be configured to follow a user. When a sensor detects that a user enters a particular place or area in a room it instantiates a digital object on the display surface that is near the user. For example, in a museum, when a sensor detects a person near a display surface a virtual tour guide can be displayed on the display surface and provide a narration (e.g., continuing where it left off when the user moved away from the previous display surface).

[0097] In another example aspect herein, a television object showing a news broadcast can follow a user throughout an apartment. The television object can be displayed wherever the user enters an area near a display screen and can cease to be displayed wherever the user leaves the display screen.

[0098] According to one example embodiment, a virtual keyboard object is provided on a display surface in physical environments where a physical keyboard is either impractical or does not exist (e.g., at a kitchen counter or in a car).

[0099] Reference will now be made to FIG. 12, which shows a flowchart illustrating an example procedure 1200 for organizing, displaying, and interacting with information on a display device.

[00100] At block 1201, a graphical interface is displayed to a user. Information is received, at block 1202, from the user based on one or more images of the graphical interface. A main display area of the graphical interface is provided at block 1203. At block 1204, digital objects are organized in a plurality of layers. The layers include a base layer and a fixed layer. The base layer corresponds to a base-surface of the main display area. An appearance of one or more base-layer objects in the base-surface window may be selectively altered by the user. The fixed layer corresponds to a fixed-surface within the

main display area. An appearance of one or more fixed-layer objects in the fixed-surface window are fixed or pinned when an appearance of a base-layer object is altered by the user. The user is enabled to (1) selectively set a digital object to the fixed-surface as a fixed-layer object (block 1205), (2) set a digital object as a base-layer object displayed in the basic-layer (block 1206), (3) instantiate new digital objects as base layer objects (block 1207), and (4) create a group of base-layer objects, such that an appearance of the group of base-layer objects may be altered in unison (block 1208).

[00101] As can be appreciated in view of the above, the example embodiments described herein provide systems, methods, and computer program products for organizing and displaying information on a display device, such as a touch sensitive display surface, that covers a wide range of needs, and is platform independent, easy to use, and easy to expand.

[00102] While various example embodiments of the invention have been described above, it should be understood that they have been presented by way of example, and not limitation. It is apparent to persons skilled in the relevant art(s) that various changes in form and detail can be made therein. Thus, the invention should not be limited by any of the above described example embodiments, but should be defined only in accordance with the following claims and their equivalents.

[00103] In addition, it should be understood that the figures are presented for example purposes only. The architecture of the example embodiments presented herein is sufficiently flexible and configurable, such that it may be utilized and navigated in ways other than that shown in the accompanying figures.

[00104] Further, the purpose of the Abstract is to enable the general public, and especially the scientists, engineers and practitioners in the art who are not familiar with patent or legal terms or phraseology, to determine quickly from a cursory inspection the nature and essence of the technical disclosure of the application. The Abstract is not intended to be limiting as to the scope of the example embodiments presented herein in any way. It is also to be understood that the procedures recited in the claims need not be performed in the order presented.

CLAIMS

1. A system of organizing, displaying, and interacting with information on a display device, the system comprising:

a computer processor; and

a memory device accessible by the computer processor and storing at least one of:

computer code executable by the computer processor, and

data used by the computer code, wherein the computer code when executed

causes the computer processor to:

provide, to at least one display device, a graphical interface that includes a main display area;

organize digital objects in a plurality of layers, each digital object corresponding to a user interface for an interactive application, the layers including:

a base layer corresponding to a base-surface window of the main display area, wherein an appearance of one or more base-layer objects in the base-surface window may be selectively altered by a user via an input device working in conjunction with graphical information of the graphical interface, the input device enabling the user to input information to the computer processor, and

a fixed layer corresponding to a fixed-surface window of the main display area, wherein an appearance of one or more fixed-layer objects in the fixed-surface window are fixed or pinned when an appearance of a base-layer object is altered by the user via the input device;

form, based on user instructions inputted using the input device via the graphical interface, a group by overlaying a plurality of base-layer digital objects in the base-surface window of the main display area, each digital object of the group having an area that overlaps with an area of another digital object of the group; and

synchronize the digital objects of the group such that a manipulation to one of the digital objects of the group, based on a user instruction, affects a remainder of the digital objects of the group.

2. The system according to Claim 1,

wherein the computer code when executed causes the computer processor to provide the graphical interface to a plurality of display devices,

wherein the digital objects of the group are displayed on each of the display devices, and

wherein the digital objects of the group are synchronized across the display devices such that the manipulation to the one of the digital objects of the group, based on the user instruction, affects a remainder of the digital objects of the group and appears on all of the display devices.

3. The system according to Claim 1, wherein the manipulation to the one of the digital objects of the group causes a change in any one or a combination of:

an appearance of all of the digital objects of the group,

an orientation of all of the digital objects of the group, and

an operation of all of the digital objects of the group.

4. The system according to Claim 1, wherein the manipulation to the one of the digital objects of the group involves a single touchpoint on the graphical interface.

5. The system according to Claim 1, wherein the input device, which is useable to input information to the computer processor via the graphical interface, includes at least one of: a mouse, a keypad, a touchpad, an infrared device, a trackball, a stylus, a group of buttons, a group of knobs, a group of direction keys, and a microphone.

6. The system according to Claim 2,

wherein the display devices include a first display device and a second display device located remotely from the first display device, and

wherein a change made to an object based on a user instruction inputted via a graphical interface of the first display device appears on a graphical interface of the second display device located remotely from the first display device.

7. A system of organizing, displaying, and interacting with information on a display device, the system comprising:

a computer processor;

a memory device accessible by the computer processor and storing at least one of:

computer code executable by the computer processor, and

data used by the computer code, wherein the computer code when executed

causes the computer processor to:

provide, to at least one display device, a graphical interface that includes a main display area;

organize digital objects in a plurality of layers, the layers including:

a base layer corresponding to a base-surface window of the main display area, wherein an appearance of one or more base-layer objects in the base-surface window may be selectively altered by a user via an input device working in conjunction with graphical information of the graphical interface, the input device enabling the user to input information to the computer processor, and

a fixed layer corresponding to a fixed-surface window of the main display area, wherein an appearance of one or more fixed-layer objects in the fixed-surface window are fixed or pinned when an appearance of a base-layer object is altered by the user via the input device;

form, based on a docking instruction inputted using the input device via the graphical interface, a docked group of digital objects, each digital object of the docked group of digital objects corresponding to a user interface for an interactive application, wherein the docked group of digital objects produces a combined output that requires contributions by each interactive application corresponding to the docked group of digital objects.

8. The system according to Claim 7, wherein each digital object of the docked group of digital objects touches at least one other digital object of the docked group of digital objects on the graphical interface.

9. A system of organizing, displaying, and interacting with information on a display device, the system comprising:

a computer processor;

a memory device accessible by the computer processor and storing at least one of:

computer code executable by the computer processor, and

data used by the computer code, wherein the computer code when executed

causes the computer processor to:

provide, to each display device of a plurality of display devices, a

graphical interface that includes a main display area,

wherein, for each display device, digital objects are organized in a plurality of layers, with each digital object corresponding to a user interface for an interactive application, and

wherein, for each display device, the layers include:

a base layer corresponding to a base-surface window of the main display area, an appearance of one or more base-layer objects in the base-surface window being selectively alterable by a user via an input device working in conjunction with graphical information of the graphical interface, the input device enabling the user to input information to the computer processor, and

a fixed layer corresponding to a fixed-surface window of the main display area, an appearance of one or more fixed-layer objects in the fixed-surface window being fixed or pinned when an appearance of a base-layer object is altered by the user via the input device;

synchronize the display devices such that the display devices operate together as a single base-surface window, when the display devices are in proximity to each other; and,

based on a user instruction, drag a digital object such that the digital object appears to move from a first display device of the display devices to a second display device of the display devices as if the first and second display devices are part of a single display device.

10. The system according to Claim 9, further comprising a sensor for detecting a spatial proximity to the first and second display devices.

11. A method of organizing, displaying, and interacting with information on a display device, the method comprising steps of:

providing, from a computer processor to at least one display device, a graphical interface that includes a main display area;

organizing digital objects in a plurality of layers, each digital object corresponding to a user interface for an interactive application, the layers including:

a base layer corresponding to a base-surface window of the main display area, wherein an appearance of one or more base-layer objects in the base-surface window may be selectively altered by a user via an input device working in conjunction with graphical information of the graphical interface, the input device enabling the user to input information to the computer processor, and

a fixed layer corresponding to a fixed-surface window of the main display area, wherein an appearance of one or more fixed-layer objects in the fixed-surface window are fixed or pinned when an appearance of a base-layer object is altered by the user via the input device;

receiving, by the computer processor, user instructions inputted via the graphical interface using the input device, the user instructions causing the computer processor to form a group by causing a plurality of base-layer digital objects in the base-surface window of the main display area to be overlapped, such that each digital object of the group has an area that overlaps with an area of another digital object of the group; and

synchronizing the digital objects of the group such that a manipulation to one of the digital objects of the group, based on a user instruction, affects a remainder of the digital objects of the group.

12. The method according to Claim 11,

wherein, in the providing step, the computer processor provides the graphical interface to a plurality of display devices,

wherein the digital objects of the group are displayed on each of the display devices, and

wherein, in the synchronizing step, the digital objects of the group are synchronized across the display devices such that the manipulation to the one of the digital objects of the group, based on the user instruction, affects a remainder of the digital objects of the group and appears on all of the display devices.

13. The method according to Claim 11, wherein the manipulation to the one of the digital objects of the group causes a change in any one or a combination of:

- an appearance of all of the digital objects of the group,
- an orientation of all of the digital objects of the group, and
- an operation of all of the digital objects of the group.

14. The method according to Claim 11, wherein the manipulation to the one of the digital objects of the group involves a single touchpoint on the graphical interface.

15. The method according to Claim 11, wherein, in the receiving step, the computer processor receives via the graphical interface the user instructions inputted using an input device that includes at least one of: a mouse, a keypad, a touchpad, an infrared device, a trackball, a stylus, a group of buttons, a group of knobs, a group of direction keys, and a microphone.

16. The method according to Claim 12,

- wherein the display devices include a first display device and a second display device located remotely from the first display device, and

- wherein, in the synchronizing step, a change made to an object based on a user instruction inputted via a graphical interface of the first display device appears on a graphical interface of the second display device located remotely from the first display device.

17. A method of organizing, displaying, and interacting with information on a display device, the method comprising steps of:

- providing, from a computer processor to at least one display device, a graphical interface that includes a main display area;

organizing digital objects in a plurality of layers, the layers including:

a base layer corresponding to a base-surface window of the main display area, wherein an appearance of one or more base-layer objects in the base-surface window may be selectively altered by a user via an input device working in conjunction with graphical information of the graphical interface, the input device enabling the user to input information to the computer processor, and

a fixed layer corresponding to a fixed-surface window of the main display area, wherein an appearance of one or more fixed-layer objects in the fixed-surface window are fixed or pinned when an appearance of a base-layer object is altered by the user via the input device;

receiving, by the computer processor, a docking instruction inputted using the input device via the graphical interface, the docking instruction causing the computer processor to form a docked group of digital objects, each digital object of the docked group of digital objects corresponding to a user interface for an interactive application, wherein the docked group of digital objects produces a combined output that requires contributions by each interactive application corresponding to the docked group of digital objects.

18. The method according to Claim 17, wherein each digital object of the docked group of digital objects touches at least one other digital object of the docked group of digital objects on the graphical interface.

19. A method of organizing, displaying, and interacting with information on a display device, the method comprising steps of:

providing, by a computer processor to each display device of a plurality of display devices, a graphical interface that includes a main display area,

wherein, for each display device, digital objects are organized in a plurality of layers, with each digital object corresponding to a user interface for an interactive application, and

wherein, for each display device, the layers include:

a base layer corresponding to a base-surface window of the main display area, an appearance of one or more base-layer objects in the base-surface window being

selectively alterable by a user via an input device working in conjunction with graphical information of the graphical interface, the input device enabling the user to input information to the computer processor, and

a fixed layer corresponding to a fixed-surface window of the main display area, an appearance of one or more fixed-layer objects in the fixed-surface window being fixed or pinned when an appearance of a base-layer object is altered by the user via the input device;

synchronizing the display devices such that the display devices operate together as a single base-surface window, when the display devices are in proximity to each other; and

receiving, by the computer processor, a user instruction that causes the computer processor to move a digital object such that the digital object appears to be dragged from a first display device of the display devices to a second display device of the display devices as if the first and second display devices are part of a single display device.

20. The method according to Claim 19, further comprising a step of detecting a spatial proximity to the first and second display devices.

21. A non-transitory computer-readable medium having stored thereon sequences of instructions that, when executed by a computer system, cause the computer system to perform method of organizing, displaying, and interacting with information on a display device, wherein the method comprises steps of:

providing, from a computer processor to at least one display device, a graphical interface that includes a main display area;

organizing digital objects in a plurality of layers, each digital object corresponding to a user interface for an interactive application, the layers including:

a base layer corresponding to a base-surface window of the main display area, wherein an appearance of one or more base-layer objects in the base-surface window may be selectively altered by a user via an input device working in conjunction with graphical information of the graphical interface, the input device enabling the user to input information to the computer processor, and

a fixed layer corresponding to a fixed-surface window of the main display area, wherein an appearance of one or more fixed-layer objects in the fixed-surface window are fixed or pinned when an appearance of a base-layer object is altered by the user via the input device;

receiving, by the computer processor, user instructions inputted via the graphical interface using the input device, the user instructions causing the computer processor to form a group by causing a plurality of base-layer digital objects in the base-surface window of the main display area to be overlapped, such that each digital object of the group has an area that overlaps with an area of another digital object of the group; and

synchronizing the digital objects of the group such that a manipulation to one of the digital objects of the group, based on a user instruction, affects a remainder of the digital objects of the group.

22. The non-transitory computer-readable medium according to Claim 21,

wherein, in the providing step, the computer processor provides the graphical interface to a plurality of display devices,

wherein the digital objects of the group are displayed on each of the display devices, and

wherein, in the synchronizing step, the digital objects of the group are synchronized across the display devices such that the manipulation to the one of the digital objects of the group, based on the user instruction, affects a remainder of the digital objects of the group and appears on all of the display devices.

23. The non-transitory computer-readable medium according to Claim 21, wherein the manipulation to the one of the digital objects of the group causes a change in any one or a combination of:

an appearance of all of the digital objects of the group,

an orientation of all of the digital objects of the group, and

an operation of all of the digital objects of the group.

24. The non-transitory computer-readable medium according to Claim 21, wherein the manipulation to the one of the digital objects of the group involves a single touchpoint on the graphical interface.

25. The non-transitory computer-readable medium according to Claim 21, wherein, in the receiving step, the computer processor receives via the graphical interface the user instructions inputted using an input device that includes at least one of: a mouse, a keypad, a touchpad, an infrared device, a trackball, a stylus, a group of buttons, a group of knobs, a group of direction keys, and a microphone.

26. The non-transitory computer-readable medium according to Claim 22,
wherein the display devices include a first display device and a second display device located remotely from the first display device, and

wherein, in the synchronizing step, a change made to an object based on a user instruction inputted via a graphical interface of the first display device appears on a graphical interface of the second display device located remotely from the first display device.

27. A non-transitory computer-readable medium having stored thereon sequences of instructions that, when executed by a computer system, cause the computer system to perform method of organizing, displaying, and interacting with information on a display device, wherein the method comprises steps of:

providing, from a computer processor to at least one display device, a graphical interface that includes a main display area;

organizing digital objects in a plurality of layers, the layers including:

a base layer corresponding to a base-surface window of the main display area, wherein an appearance of one or more base-layer objects in the base-surface window may be selectively altered by a user via an input device working in conjunction with graphical information of the graphical interface, the input device enabling the user to input information to the computer processor, and

a fixed layer corresponding to a fixed-surface window of the main display area, wherein an appearance of one or more fixed-layer objects in the fixed-surface window are fixed or pinned when an appearance of a base-layer object is altered by the user via the input device;

receiving, by the computer processor, a docking instruction inputted using the input device via the graphical interface, the docking instruction causing the computer processor to form a docked group of digital objects, each digital object of the docked group of digital objects corresponding to a user interface for an interactive application, wherein the docked group of digital objects produces a combined output that requires contributions by each interactive application corresponding to the docked group of digital objects.

28. The non-transitory computer-readable medium according to Claim 27, wherein each digital object of the docked group of digital objects touches at least one other digital object of the docked group of digital objects on the graphical interface.

29. A non-transitory computer-readable medium having stored thereon sequences of instructions that, when executed by a computer system, cause the computer system to perform method of organizing, displaying, and interacting with information on a display device, wherein the method comprises steps of:

providing, by a computer processor to each display device of a plurality of display devices, a graphical interface that includes a main display area,

wherein, for each display device, digital objects are organized in a plurality of layers, with each digital object corresponding to a user interface for an interactive application, and

wherein, for each display device, the layers include:

a base layer corresponding to a base-surface window of the main display area, an appearance of one or more base-layer objects in the base-surface window being selectively alterable by a user via an input device working in conjunction with graphical information of the graphical interface, the input device enabling the user to input information to the computer processor, and

a fixed layer corresponding to a fixed-surface window of the main display area, an appearance of one or more fixed-layer objects in the fixed-surface window being fixed or pinned when an appearance of a base-layer object is altered by the user via the input device;

synchronizing the display devices such that the display devices operate together as a single base-surface window, when the display devices are in proximity to each other; and

receiving, by the computer processor, a user instruction that causes the computer processor to move a digital object such that the digital object appears to be dragged from a first display device of the display devices to a second display device of the display devices as if the first and second display devices are part of a single display device.

30. The non-transitory computer-readable medium according to Claim 29, wherein the method further comprises a step of detecting a spatial proximity to the first and second display devices.