PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06T	A2	(11) International Publication Number: **WO 97/35280**
		(43) International Publication Date: 25 September 1997 (25.09.97)

(21) International Application Number: PCT/IL97/00093

(22) International Filing Date: 13 March 1997 (13.03.97)

(30) Priority Data:
60/013,624 15 March 1996 (15.03.96) US

(71) Applicant *(for all designated States except US)*: ZAPA DIGITAL ARTS LTD. [IL/IL]; 14 Yehuda Halevi Street, 65137 Tel Aviv (IL).

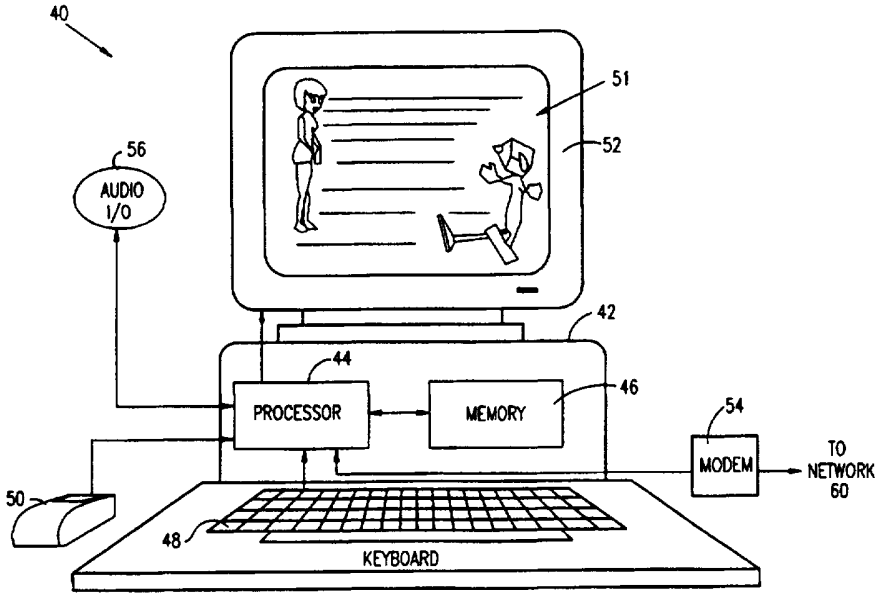
(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: GEVER, Eyal [IL/IL]; 18 Ha'kovshim Street, 68012 Tel Aviv (IL). HERMONI, Nir [IL/IL]; 35 Sokolov Street, 52564 Ramat Gan (IL). BERGMAN, Orit [IL/IL]; 11 K'far Yona Street, 69053 Tel Aviv (IL). TAYAR, Gil [IL/IL]; 69 Jabotinski Street, 53319 Giv'atayim (IL). RESHEF, Eilon [IL/IL]; 31 Melchet Street, 63114 Tel Aviv (IL). GILL, Doron [IL/IL]; 24 Arlozorov Street, 53373 Giv'atayim (IL). FEUERSTEIN, Addy [IL/IL]; 33 Balfur Street, 65211 Tel Aviv (IL). CANETI, Yaron [IL/IL]; 16 Karmiah Street, 64259 Tel Aviv (IL). OPPENHEIM, Roy [IL/IL]; 18 Ha'hashmona'im Street, 63264 Tel Aviv (IL). ETAM, Eran [IL/IL]; 1 Rekanati Street, 69494 Tel Aviv (IL). SHPEIZER, Zohar [IL/IL]; 3 La'hover Street, 69023 Tel Aviv (IL). BORER, Yoav [IL/IL]; 5 Hofyan Street, 69103 Tel Aviv (IL). LIVINE, Eyal [IL/IL]; 8 Karl Neter Street, 44421 K'far Saba (IL). KIKIN, Ruth [IL/IL]; 33 Gordon Street, 63407 Tel Aviv (IL). ZUCKERMAN, Oren [IL/IL]; 5 Herut Street, 52541 Ramat Gan (IL). KAAS, Ron [IL/IL]; 5 Moshav, 73115 Ben-Shemen (IL).

(74) Agents: SANFORD, T., Colb et al.; Sanford T. Colb & Co., P.O. Box 2273, 76122 Rehovot (IL).

(81) Designated States: AL, AM, AT, AT (Utility model), AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, CZ (Utility model), DE, DE (Utility model), DK, DK (Utility model), EE, EE (Utility model), ES, FI, FI (Utility model), GB, GE, GH, HU, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SK (Utility model), TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ARIPO patent (GH, KE, LS, MW, SD, SZ, UG), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).

Published
Without international search report and to be republished upon receipt of that report.

(54) Title: PROGRAMMABLE COMPUTER GRAPHIC OBJECTS



(57) Abstract

A method for producing an animation sequence on a graphic display driven by a computer, including defining an object that includes a geometrical description of an animated character and characteristics of social behaviour of the character and animating an image of the character responsive to the characteristics. The character is programmed with a predetermined response, such as a rule governing motion of the character, to a sensitivity condition occurring externally to the object. The geometrical description of the object includes a geometrical skeleton characterized by a hierarchy of sub-objects connected by joints, the rule governing motion defines motions of the joints.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	**ES**	Spain	**LS**	Lesotho	**SI**	Slovenia
AM	Armenia	**FI**	Finland	**LT**	Lithuania	**SK**	Slovakia
AT	Austria	**FR**	France	**LU**	Luxembourg	**SN**	Senegal
AU	Australia	**GA**	Gabon	**LV**	Latvia	**SZ**	Swaziland
AZ	Azerbaijan	**GB**	United Kingdom	**MC**	Monaco	**TD**	Chad
BA	Bosnia and Herzegovina	**GE**	Georgia	**MD**	Republic of Moldova	**TG**	Togo
BB	Barbados	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tajikistan
BE	Belgium	**GN**	Guinea	**MK**	The former Yugoslav Republic of Macedonia	**TM**	Turkmenistan
BF	Burkina Faso	**GR**	Greece			**TR**	Turkey
BG	Bulgaria	**HU**	Hungary	**ML**	Mali	**TT**	Trinidad and Tobago
BJ	Benin	**IE**	Ireland	**MN**	Mongolia	**UA**	Ukraine
BR	Brazil	**IL**	Israel	**MR**	Mauritania	**UG**	Uganda
BY	Belarus	**IS**	Iceland	**MW**	Malawi	**US**	United States of America
CA	Canada	**IT**	Italy	**MX**	Mexico	**UZ**	Uzbekistan
CF	Central African Republic	**JP**	Japan	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Netherlands	**YU**	Yugoslavia
CH	Switzerland	**KG**	Kyrgyzstan	**NO**	Norway	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	Democratic People's Republic of Korea	**NZ**	New Zealand		
CM	Cameroon			**PL**	Poland		
CN	China	**KR**	Republic of Korea	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Romania		
CZ	Czech Republic	**LC**	Saint Lucia	**RU**	Russian Federation		
DE	Germany	**LI**	Liechtenstein	**SD**	Sudan		
DK	Denmark	**LK**	Sri Lanka	**SE**	Sweden		
EE	Estonia	**LR**	Liberia	**SG**	Singapore		

PROGRAMMABLE COMPUTER GRAPHIC OBJECTS

This application claims the benefit of U.S. Provisional Patent Application No. 60/013,624, filed March 15, 1996, which is incorporated herein by reference.

FIELD OF THE INVENTION

The present invention relates generally to computer animation, and specifically to interactive generation and manipulation of moving computer graphic objects.

BACKGROUND OF THE INVENTION

Computer animation systems and methods are well known in the art, serving a wide range of applications in which a computer is used to generate and/or manipulate moving images. In recent years, the rapid advance in computing speed and reduction in cost of personal computers (PC's) has made computer animation widely accessible. Because of the relatively heavy computation load associated with three-dimensional (3D) image rendering, most of the animation programs for PC's have been able to offer only 2D imaging or very limited 3D effects. Similarly, although 2D animations are now commonly conveyed from one computer to another over the Internet and other network systems, the bandwidth limitations of the Internet have, at least until recently, made it generally impractical to convey 3D animations.

Software tools and languages are available to assist programmers in creating 3D animations. For example, "LightWave 5.0," produced by Newtek, Inc., of Topeka, Kansas, is a software package that enables users to define and manipulate geometrical models and animations of 3D graphic objects within a computer animation environment. Such objects are defined as skeletons, or hierarchies of geometrical shapes that are connected together at joints. These objects may further include methods, or program routines, that define the basic movements of the joints. Thus, an animated humanoid character may be defined by a skeleton including torso, head, arms, legs, hands, feet, and other features, with interlinking joints and methods that describe walking, running, hand-waving and other actions. Motion of the character along a desired path in a 3D virtual space may be engendered by indicating positions of the object in two or more key frames, whereupon the intermediate positions of the object between the key frames are interpolated, using the appropriate methods belonging to the object.

Based on the 3D geometrical definition of the object, an image of the animated character is rendered on a computer screen. Various software packages are available for 3D rendering, for example, "RenderWare V2.0," produced by Criterion Software Ltd. of Surrey, UK. The rendering software typically takes into account the effects of light, shade, color, surface textures, perspective and other visual elements, to create a convincingly "3D" image on the flat computer screen. Generally, multiple 3D objects are rendered together, along with a suitable background, to form a single image on screen.

U.S. Patent 5,261,041, to Susman, which is incorporated herein by reference, describes a computer-controlled animation system based on manipulation of animated objects. Each object

includes state data and methods, or functions, defining the behavior of the object. The objects also have associated affect volumes and affect agents, or methods, that determine how they will interact with each other as an animation sequence progresses. The animation sequence is governed by a set of rules, graphs or scripts. The behavior of the objects in the sequence is modeled and rendered on a display screen.

U.S. Patent 5,483,630, to Unuma et al., which is incorporated herein by reference, describes a method for representing motion of multiple-jointed objects, for use in computer animation. The bending angles of the joints of a multiple-jointed object are represented by functions expressed independently of the length between the joints. Based on the functions, contour data are produced representing the motion of the joints, which data are used in rendering the objects in computer animation sequences. Parameters of the functions can be adjusted so that the motion of the joints has a desired character, for example, running or walking motion, or motion having a manner intended to give a sense of an emotional quality.

U.S. Patent 5,267,154, to Takeuchi et al., which is incorporated herein by reference, describes a system for producing "biological images," i.e., animated computer images of human and animal characters. Images are synthesized by combining data from several databases in which shapes, motions and external features and textures are stored. The system is used to produce a variety of 3D animated characters, which take part in computer-animated image sequences.

U.S. Patent 5,577,185, to Tunnell et al., which is incorporated herein by reference, describes a method and apparatus for creating computer-animated puzzles. The puzzles are made up of animated objects, such as structural members and characters, displayed on a computer screen. Each of the objects has a set of programmed physical and behavioral attributes, as a result of which the objects appear on screen to be aware of and to react to one another. They are programmed to appear to obey physical laws, such as the laws of gravity and motion, and to exhibit natural tendencies, such as hunger and affinity. The puzzles are created by selecting the desired objects from a menu displayed on the computer screen.

Computer-animated images may be overlaid on other images and/or windows on a computer display screen. For example, U.S. Patent 5,546,518, to Blossom et al., which is incorporated herein by reference, describes a system and method for composing a display frame of multiple, layered graphic "sprites." A sprite is a graphic image that forms a part or a region of an overall computer display screen. The sprites are overlaid one above the other. The sprites preferably include transparent pixels in certain areas, through which underlying graphics can be seen. The system of sprites is used to combine video and/or animated images together with still images on a single screen.

Recently, tools have begun to emerge for sending and receiving 3D animations over computer networks, such as the Internet, for example, Version 2.0 of the Virtual Reality Modeling Language (VRML 2.0), as described in "Moving Worlds," at http://vrml.sgi.com

(Silicon Graphics Inc.), and incorporated herein by reference. VRML 2.0 was developed by a group of companies to provide a language for creating animations in a compact form that is convenient for transmission over the Internet, and for viewing the animations at the receiving end. VRML 2.0 includes conventions for a user at a source computer to define and organize elements of an interactive animated image, such as 3D object geometries, surface textures, motion and action scripts and to encapsulate these elements for Internet transmission. When the encapsulated elements are received by a target computer having VRML-compatible network browser software, the animated image created at the source computer is rendered on the target computer screen.

Viewers of 3D animations generally require user interface software to enable them to observe and manipulate the animated images. Such software is known in the art, for example, "CosmoPlayer," a VRML 2.0-compatible network browser, available from Silicon Graphics at http://vrml.sgi.com, as described above. CosmoPlayer operates as a "plug-in" to an Internet web browser, such as Netscape "Navigator" or Microsoft Internet Explorer." CosmoPlayer enables a user to view animations of 3D objects and to vary a point of view from which the animations are rendered to the user's computer screen by means of "free navigation" in a virtual 3D space. It will be appreciated that because the computer screen and user controls are essentially two-dimensional, "free navigation" is generally difficult for a non-expert user to master.

Computer-animated images may likewise be used to create an animated electronic meeting place, as described, for example, in U.S. Patent 5,347,306, to Nitta, which is incorporated herein by reference. Each participant in a meeting has a terminal with prestored local animation graphics. The terminals are connected to a network, over which the meeting takes place. The meeting is shown on the display of each of the terminals using real-time animated 3D graphic characters and sound, representing the participants' motions, expressions and speech. Each of the participants is represented in the animation by one of the animated characters, having a persona which may be chosen by the participant. The characters in the animated meeting are driven by sensors that sense the movement and speech of the actual participants. The participants may also use view controls, such as a joystick, to zoom or to alter their viewing perspectives.

SUMMARY OF THE INVENTION

It is an object of the present invention to provide improved methods and apparatus for producing an animated image of an object, preferably a three-dimensional (3D) object, in a computer graphic display.

Unlike computer animations known in the art, animated objects in accordance with preferred embodiments of the present invention are substantially autonomous, i.e., the objects include rules, preferably in the form of a scripting language, which are sufficient to determine how they will behave in a given scene. In some aspects of the present invention, the objects correspond to humanoid characters, and the rules define social behavior of the characters in the scene, imparting human personality types to the animated characters. Furthermore, the animated objects, including the rules, are modular, in that behaviors may be transferred freely among objects of a common type. The methods of implementation and the implications of this autonomy and behavioral modularity of animated objects are described below.

It is an object of some aspects of the present invention to enable the image of the 3D object to be rendered as an overlay on a display generated by another software application, not necessarily related to the Scene Manager program, for example, an Internet Web Browser.

It is a further object of some aspects of the present invention to provide methods and apparatus for representing an object by means of a modular set of parameters and/or data, which is transferred over a computer network from a source computer to a destination computer, whereupon an image of the object is produced in an animated computer graphic display at the destination computer.

Preferably, the set of parameters and/or data is encapsulated and transferred in the form of an electronic mail message.

It is another object of some aspects of the present invention to provide user interface methods and apparatus, for use in interactively viewing and controlling computer animations.

It is an additional object of some aspects of the present invention to provide visual methods and apparatus enabling a user to search conveniently through a database or library of computer graphic objects, useful particularly in creating computer animations.

In preferred embodiments of the present invention, computer animation apparatus comprises an animation generator, typically a computer, for example, a personal computer (PC). The computer has an associated video graphic display, memory and user input devices, such as a keyboard, mouse and/or joystick, as are known in the art. Preferably, the computer also includes a network connection, for example, a modem through which the computer is connected to the Internet. The computer is programmed to produce an animated graphic sequence, using methods described below, and the sequence is shown on the display.

In preferred embodiments of the present invention, the animated graphic sequence is produced using 3D Smart Objects, as described herein. In the context of the present patent application, the term "object" is used to refer to a modular, graphic and/or mathematical

description of an image element (generally a representation of a physical object, which may be naturalistic or fanciful), whose representation may appear in the animation sequence. The term "Smart Object" as used herein, in accordance with preferred embodiments of the present invention, refers to an object comprising both the graphic/mathematical description of an image element and embedded ancillary information about the element. Such ancillary information preferably comprises procedures, typically in the form of scripts, which describe attributes, motion and behavior of the element, as described hereinbelow, and may also include other information, such as textual reference data. "3D Smart Objects" include attributes of shape and motion that are used in rendering 3D images of the corresponding image elements.

Each Smart Object preferably comprises a set of sub-objects in a predetermined hierarchy, which defines the relationships between the sub-objects. The hierarchy preferably includes a "skeleton," defining the parts of the body of the image element, the joints connecting those parts, and the motions of the parts in animation. Other components of the Smart Object describe attributes such as sound and rules governing behavior of the Smart Object. The hierarchical structure of Smart Objects allows a user to exchange organs and sub-organs, as well as behaviors, between Smart Objects that share a common hierarchy. New Smart Objects may be created by inheritance of attributes from an existing Smart Object.

Image elements or characters that correspond to Smart Objects appear in a computer animation sequence to move and function autonomously, based on the behavioral rules embedded in the Smart Objects. For example, a sub-object of a Smart Object preferably specifies an "idle" behavior, i.e., a motion or other action that the corresponding image element will undertake when the Smart Object has received no other program instructions for a predetermined period of time. Preferably, the sub-objects in the hierarchy are chosen to impart a desired "personality" to the Smart Object, wherein the personality comprises a suitable combination of physical appearance, facial expressions, characteristic movement and gestures, voice and other attributes that a user will associate with a certain personality type. Further preferably, the user may design a Smart Object, using methods described below, to represent a particular character.

Preferably, Smart Objects also include sub-objects that define their response to sensitivity conditions that may arise externally to the Smart Object itself, such as the presence of other objects in the computer display. Such sub-objects may specify an interaction or response of one Smart Object to another Smart Object, for example, attraction of one Smart Object having a male persona to another Smart Object having a female persona. Additionally or alternatively, sub-objects may specify the response of the Smart Object to other image elements shown in the computer display, for example, to static or moving graphic icons generated by other software applications, which may or may not include animation and which do not themselves make use of Smart Objects. Furthermore, Smart Objects may include sub-objects that have learning

capabilities, as are known in the art, whereby the behavior of the Smart Object is modified adaptively in response to inputs from the computer-animated environment.

A Smart Object, as described herein, includes data, such as geometrical parameters and scripts, which are read by a program that generates an animated image sequence including the image element corresponding to the Smart Object. In the context of the present patent application and in the claims, such a program is referred to as a Scene Manager. The Scene Manager preferably includes program modules that read the data from the Smart Object and then, based on the Smart Object's geometrical and behavioral parameters, create an animated mathematical representation of the Smart Object. This representation is rendered to the computer display, preferably by means of a rendering module within the Scene manager, using 3D rendering techniques known in the art.

In some preferred embodiments of the present invention, the Scene Manager comprises a "plug-in" to network browser software. The Scene Manager plug-in is installed and operated by a user of the browser software in a manner similar to other animation plug-ins known in the art. In these preferred embodiments, the Scene Manager can receive Smart Objects, preferably 3D Smart Objects, over the network and then can render images of the Smart Objects within or overlaid on the browser window on the computer display.

In other preferred embodiments of the present invention, the Scene Manager comprises an application program, which runs in the framework of an operating system, preferably a windows-based operating system, as is well known in the art. In these preferred embodiments, the Scene Manager operates to create animated images, as described herein, while other, generally unrelated applications, such as a word processor program, are simultaneously running on the computer.

In some preferred embodiments of the present invention, a computer animation sequence is controlled by scripts, which control the actions of the Smart Objects in the sequence, responsive to the attributes defined by the Smart Objects themselves. The scripts are preferably embedded within the Smart Objects, as described above, but may, alternatively or additionally, belong to the Scene Manager program, which interacts with and controls the Smart Objects. The scripts preferably comprise motion scripts, which include instructions to modify the position, orientation or other attributes of the Smart Objects over time, from one animation frame to the next. Additionally or alternatively, the scripts may comprise connection scripts, which link the Smart Objects to input devices, such as a mouse or 3D pointing device, which are then used to manipulate the Smart Objects in the display. Further additionally or alternatively, the scripts may comprise trigger scripts, which are invoked when a predetermined sensitivity condition or event occurs, for example, a collision of the Smart Object with another object or selection of the Smart Object in the display.

In some preferred embodiments of the present invention, an object, preferably a 3D Smart Object, is used in rendering an image element that is overlaid over a window in the

display. For example, a suitable Smart Object may be used to produce a 3D image of a human figure, which walks across a window created by some other software application, such as an Internet browser or a word processor program. The window over which the image element is overlaid may be an Internet Web page, for example, and may include text or two- or three-dimensional graphics, or some combination of text and graphics. The software application associated with the window preferably continues to operate while the Smart Object is running. Preferably, attributes of the Smart Object and/or a script associated therewith cause the Smart Object to react to or interact with elements in the window, for example, by reading aloud text displayed in the window or pointing to an on-screen push-button control.

Preferably, the image element that is overlaid over the window obscures only the portion of the window immediately behind the element, while the remainder of the window is unobscured, and the application associated with the window continues to function. For each frame in a computer animation sequence involving the Smart Object, as the image element is rendered, a transparent mask is concomitantly created, defining the area of the display that will not be obscured by the image element. This mask is used to create a window for the Smart Object that is exactly shaped to fit a border circumscribing the image element. The shape of this window is adjusted in response to the full 3D motion of the Smart Object, unlike sprite overlaying methods known in the art, in which changes in the transparent areas of the sprites are substantially limited to 2D effects.

In some preferred embodiments of the present invention, a user interface is provided to enable a user to program and control Smart Objects and computer animation sequences using the Smart Objects. The user preferably creates animations by choosing Smart Objects from a library and/or by creating Smart Objects from libraries of sub-objects that define attributes of the appearance and behavior of the Smart Objects, as described above. Further preferably, the user interface enables the user to create scripts and otherwise define animation sequences including, for example, motion paths, sound and interactions of the Smart Objects with one another and/or with a viewer. In addition, the user interface preferably allows the user to define camera angles and simulated lighting characteristics that are applied in rendering the animation sequences on the computer display.

In one preferred embodiment of the present invention, the user creates a Smart Object which serves as an avatar for the user or for another individual. An image of the user or of another individual may preferably be scanned into the computer, as is known in the art, and overlaid on a surface of the image element corresponding to the Smart Object, preferably on the head of the image element.

In some of these preferred embodiments, the user interface enables the user to choose and navigate in virtual 3D space among predefined viewpoints. For each viewpoint, an image of an animation scene is rendered and displayed corresponding to an image of the scene that would be captured by a camera filming the scene from the location of the viewpoint. At each

viewpoint, the user is able to alter the effective camera angle, up/down and left/right, and to zoom in or out. The viewpoints may be stationary, but preferably, paths are defined between reference points in the 3D space, along which paths the viewpoints are to move, so that the virtual camera position changes accordingly. Additionally or alternatively, a user may indicate a reference point in the 3D space and instruct the viewpoint to orbit around the point and/or to move toward or away from the point. These methods for user control of the viewpoint are substantially easier for the user to conceptualize and control than are "free navigation" user interface tools known in the art, for example, "CosmoPlayer," a commonly-available VRML 2.0 browser program, described above.

In some of these preferred embodiments, the user interface includes on-screen graphic devices, with which the user interacts, preferably using a mouse or 3D pointing device, known in the art, to control the animation. These devices preferably include one or more sliders, push buttons, dubbing strips and other elements known in the art. The on-screen user interface devices are, preferably, automatically scaled, such that when the user resizes or changes the shape of the window on the computer screen in which the animation is displayed, the sizes, positions and spacing of the devices are adjusted accordingly. Adjusting the sizes and positions of the on-screen devices allows a maximal portion of the window to be used for displaying the animation and enhances the esthetic appeal of the display. The scaling of the user interface allows all of the controls to be seen simultaneously on the screen, even when the window is substantially reduced in size, unlike user interfaces known in the art, in which some or all of the controls disappear from the window under such conditions.

Among the on-screen devices, the user interface preferably includes a 3D joystick, which the user manipulates to navigate in the 3D space. The 3D joystick moves in a plane, thus controlling two degrees of freedom. Preferably, the joystick is used to control tilt and pan of a virtual camera from whose point of view the scene is rendered, while a slider controls the camera's zoom. Alternatively or additionally, the joystick may be used to control revolution of the virtual camera about a selected reference point in the scene.

Preferably, the portion of the window used for displaying the animation is not strictly rectangular, but is rather surrounded and framed by non-animated elements, including the user interface devices. The devices may penetrate into the edges of the frame containing the animated images. The non-rectangular frame is preferably designed to give the animated scene a more realistic appearance, as though it were taking place, for example, in an auditorium or a television studio.

In some preferred embodiments of the present invention, an associative visual search engine is provided to assist the user in selecting elements to be incorporated in a computer animation. The engine is used in searching through an image database that may include image representations of Smart Objects and sub-objects thereof, as well as other animations, still objects and background and border images. One or more keywords, generally a plurality of

keywords, are associated with each of the images in the database. The keywords preferably describe various aspects of the images, including image subject, content, style, color and other attributes. The keywords associated with each image are preferably ordered hierarchically, from the keyword that is most descriptive of the image to keywords that are less descriptive thereof.

To use the search engine, a user preferably enters a keyword or selects an image from an initial group of images displayed on the computer screen. The engine then searches the database for images matching the keyword that was entered or matching a keyword or keywords associated with the selected image. The user may narrow the search by specifying a characteristic of the selected image to be matched, such as its subject matter or color, for example. Images in the database that are found by the engine to match the keyword or keywords are displayed on the computer screen in a hierarchical, preferably generally concentric pattern. Preferably, images matching the keyword or keywords most closely, i.e., images with respect to which the keyword or keywords searched for occupy the highest positions in the images' respective hierarchies of associated keywords, are positioned nearest the center of the display and scaled to a relatively large size. Images matching the keyword or keywords more remotely are positioned peripherally.

The user may browse through the images displayed on the screen by pointing to images of interest, for example, using a mouse. When the user points to an image, the image is preferably enlarged and/or expanded. If the image represents a 3D image element, it is preferably rendered in three dimensions. If the image represents an animation or animated object, for example, a Smart Object, at least a portion of the animation or a movement of the object is displayed. The user selects images from the search for inclusion in an animation sequence or, alternatively or additionally, selects an image to serve as the starting point for a new search.

It will be appreciated that the search engine and search method described above allows the user to search through an image database in a more intuitive manner than is possible with search engines and methods known in the art, and therefore enables the user to find suitable images more efficiently and quickly. Although the associative visual search engine described above is preferably used in generating computer animation sequences, as described herein, in other preferred embodiments of the present invention, the engine may be used advantageously for searching through image databases of other types and for other purposes.

In some preferred embodiments of the present invention, after the user has used a source computer to create a computer animation, the animation is encapsulated and conveyed over a network, preferably the Internet, to a destination computer. In this context, the term "computer animation" means a computer animation sequence, which is preferably created by the user as described above, or a Smart Object having attributes of motion. Encapsulating and conveying the animation comprises sending files associated with the animation over the network, preferably in the form of an electronic mail message or other network communication. Preferably, the

animation file is encapsulated as a 3D VRML-compatible animation file, as described above, or alternatively, as a JAVA applet or HTML file or in another standard file format recognized by suitable Internet browser programs. When the animation is received, it may be replayed as an animation sequence on the destination computer display, or incorporated in an animation sequence running on the destination computer.

In some of these preferred embodiments, the animation sent from the source to the destination computer comprises an animated "letter," for example, a greeting card, sent by electronic mail. The letter preferably includes one or more Smart Objects, which are programmed at the source computer to convey a desired message and/or exhibit a desired behavior when animated on the destination computer.

In other preferred embodiments of this type, the animation sent from the source to the destination computer comprises an entertainment or educational program, distributed by a network server to a plurality of client computers. Each of the clients receives the animation as created and sent by the server, but the client user may then control certain aspects of the rendering of the animated images on the client computer screen, such as varying the viewpoint, lighting and appearance of the characters, for example. Preferably, the clients include both viewers, who receive the animation from the source but are allowed relatively little control over the animation, and participants, who may actually control a character appearing in the animation. Most preferably, each participant controls a respective character by defining the appearance and behavior of a Smart Object corresponding to the character.

In still other preferred embodiments of this type, the animation sent from the source to the destination computer comprises a Smart Object, preferably a 3D Smart Object programmed as an avatar of the user, as described above. Preferably, the avatar conveys an electronic mail message to a user at the destination computer. Additionally or alternatively, the avatar may be incorporated as a participant in an interactive, animated program on the destination computer, for example, as a participant in a conference, interview or panel discussion conducted over the Internet or another network.

There is therefore provided, in accordance with a preferred embodiment of the present invention, a method for producing an animation sequence on a graphic display driven by a computer, including:

defining an object, which includes data including a geometrical description of an animated character and characteristics of social behavior of the character; and

animating an image of the character responsive to the characteristics.

Preferably, defining the object includes programming a predetermined response of the character to a sensitivity condition occurring externally to the object.

Further preferably, programming the predetermined response to the interaction condition includes programming a rule governing motion of the object responsive to the condition.

Additionally or alternatively, defining the object includes defining sufficient characteristics of the behavior of the character such that the image is animated substantially without reference to animation instructions external to the object.

Preferably, animating the image includes rendering a three-dimensional image of the character.

There is also provided, in accordance with a preferred embodiment of the present invention, a method for producing an animation sequence on a graphic display driven by a computer, including:

defining a group of objects, each such object associated with a respective animated character and including data, which includes:

a geometrical skeleton common to all the objects in the grouped, characterized by a hierarchy of sub-objects connected by joints; and

rules governing behavior of the respective character, which rules define motions of the joints and are interchangeable among any of a plurality of the objects; and

animating an image of at least one of the characters responsive to the rules.

Preferably, defining the group of objects includes defining a three-dimensional skeleton, and animating the image includes rendering a three-dimensional image.

Additionally or alternatively, defining the group of objects includes defining rules governing behavior of a predetermined personality type, associated with one or more of the animated characters.

Preferably, defining the group of objects includes defining first and second objects, wherein the second object is defined by inheritance from the first object.

There is further provided, in accordance with a preferred embodiment of the present invention, a method for providing a user interface in a computer graphic display, including:

displaying one or more user control icons in a window in the display; and

scaling the one or more icons responsive to a change of scale of the window.

There is also provided, in accordance with a preferred embodiment of the present invention, a method for producing an animation sequence on a graphic display driven by a computer, including:

producing a frame image having a non-rectangular transparent area;

generating an animation sequence on the display; and

overlaying the frame image on the animation sequence, so that the animation is framed by the transparent area.

In accordance with another preferred embodiment of the present invention, there is additionally provided a method for producing an animation sequence on a graphic display driven by a computer, including:

defining a three-dimensional geometrical description of an animation scene;

selecting first and second reference points relative to the scene and defining a geometrical path dependent on the points;

rendering a first image of the scene as seen from a viewpoint at a first position on the geometrical path;

translating the viewpoint to a second position along the geometrical path; and

rendering a second image of the scene as seen from the second position.

There is additionally provided, in accordance with a preferred embodiment of the present invention, a method for producing an animated overlay image on a graphic display driven by a computer, including:

generating a three-dimensional animated image element; and

overlaying the animated image element on a window in the graphic display driven by a software application substantially unrelated to the generation of the animated image element.

Preferably, the substantially unrelated software application continues to run substantially as though the image element was not overlaid on the window, and the only portion of the window obscured is the portion directly behind the image element on the graphic display.

There is also provided, in accordance with another preferred embodiment of the present invention, a method for conveying an animation from a source computer to a destination computer, including:

defining an object in the source computer, the object including data, which includes a graphic description of an element for inclusion in the animation and a characteristic of motion of the element;

transmitting the object from the source computer to the destination computer via a network; and

displaying an animated image on a display driven by the destination computer, wherein the image includes a graphic rendition of the element, moving in accordance with the characteristic.

Preferably, transmitting the object via the network includes encapsulating the object and transmitting the encapsulated object in the form of an electronic mail message.

Alternatively or additionally, a script is transmitted over the network, which script defines an animation sequence including the element, and wherein transmitting the object and transmitting the script include transmitting the object and the script to a plurality of destination computers.

In accordance with another preferred embodiment of the present invention, there is moreover provided a method for finding a desired image among a library of images stored by a computer, including:

displaying a first plurality of images from the library on a display associated with the computer;

selecting a first image from among the first plurality; and

searching through the library to find a second plurality of images resembling the first image in one or more characteristics thereof.

Preferably, a cursor is shown on the display and one of the images is altered when the cursor points to the one of the images, preferably by animating the image.

There is further provided, in accordance with a preferred embodiment of the present invention, a system for producing an animation sequence, including:

an animation generator, which generates an animated image of a character, responsive to an object including a geometrical description of the character and characteristics of social behavior of the character; and

a display, which is driven by the computer to display the animated image.

In addition, in accordance with another preferred embodiment of the present invention, there is provided a system for producing an animation sequence, including:

an animation generator which generates an animated image of at least one character from a group of characters, responsive to a respective one of a group of objects,

wherein the objects include a geometrical skeleton common to all the objects in the group, characterized by a hierarchy of sub-objects connected by joints, and rules governing behavior of the respective characters, which rules define motions of the joints and are interchangeable among any of a plurality of the objects; and

a display, which is driven by the computer to display the animated image.

There is also provided, in accordance with a preferred embodiment of the present invention, a system for producing graphic images, including:

an animation generator, which generates a scalable graphic window with one or more user interface icons contained therein, such that the icons are scaled responsive to a scale of the window;

a user input device for scaling the window; and

a display, which is driven by the computer to display the window with the icons.

There is moreover provided, in accordance with a preferred embodiment of the present invention, a system for producing an animation sequence, including:

an animation generator, which generates an animation sequence and which produced a frame image having a non-rectangular transparent area, such that the frame image is overlaid on the animation sequence, whereby the animation is framed by the transparent area; and

a display, which is driven by the computer to display the animation sequence framed by the transparent area.

Additionally, in accordance with another preferred embodiment of the present invention, there is provided a system for producing an animation sequence, including:

an animation generator, which renders animated images of a three-dimensional scene, each image rendered as seen from a respective viewpoint, including a first image of the scene as

seen from a first viewpoint along a predetermined geometrical path and a second image of the scene as seen from a second viewpoint along the path;

a user input device, for selecting first and second reference points, which determine the geometrical path; and

a display, which is driven by the computer to display the animated images.

There is further provided, in accordance with a preferred embodiment of the present invention, a system for producing an animated image, including:

an animation generator, which generates a three-dimensional animated image element and produces a display window driven by a software application substantially unrelated to the generation of the animated image element, and which overlays the animated image element on the window; and

a display, which is driven by the computer to display the window with the animated image overlaid thereon.

There is further provided, in accordance with a preferred embodiment of the present invention, a system for conveying an animation over a network, including a source animation generator, coupled to the network, which defines an animation file comprising an object, which includes a graphic description of an image element for inclusion in the animation and a characteristic of motion of the element and transmits the object over the network in the form of an electronic mail message.

In accordance with another preferred embodiment of the present invention, there is provided a system for conveying an animation over a network, including a network animation server, which receives a textual description of an animation object, generates an animation file based thereon, and delivers the animation file to a destination computer.

There is also provided, in accordance with another preferred embodiment of the present invention, a system for viewing an animation, including:

a destination animation generator, coupled to a network, which receives via the network a script defining the animation along with an object including a geometrical description of an image element for inclusion in the animation and a characteristic of motion of the element, and renders animated images responsive thereto;

user controls, coupled to the destination animation generator, for controlling rendition of the animated images; and

a display, which is driven by the destination animation generator to display the animated images.

There is additionally provided, in accordance with a preferred embodiment of the present invention, a system for searching a library of images, including:

image storage apparatus, which stores the library of images and selects a first plurality of the images to be displayed;

a display, which is driven by the image storage apparatus to display the images selected by the animation generator ; and

a user input device, coupled to the image storage apparatus, for pointing to and selecting a first image from among the first plurality of the images displayed,

wherein the animation generator searches through the library and selects images resembling the first image in one or more characteristics thereof, for inclusion in a second plurality of images.

The present invention will be more fully understood from the following detailed description of the preferred embodiments thereof, taken together with the drawings in which:

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a schematic representation of a computer system for use in producing and displaying animation sequences, in accordance with a preferred embodiment of the present invention;

Fig. 2 is a graphic representation of a computer image element generated on a computer display screen by a Scene Manager responsive to a 3D Smart Object, in accordance with a preferred embodiment of the present invention;

Figs. 3A and 3B are graphic representations of the computer image elements generated on the computer display screen by Smart Objects, illustrating animation of the elements in response to one another, in accordance with a preferred embodiment of the present invention;

Fig. 4 is a block diagram illustrating the program structure of a 3D Smart Object, in accordance with a preferred embodiment of the present invention;

Fig. 5 is a flow chart illustrating a method of generating a computer animation sequence including the 3D Smart Object of Fig. 4, in accordance with a preferred embodiment of the present invention;

Figs. 6A and 6B are graphic representations of an image element generated on a computer display screen by a 3D Smart Object, wherein the image element is overlaid on a display window generated by another software application, in accordance with a preferred embodiment of the present invention;

Fig. 7 is a flow chart illustrating a method of overlaying the image element on the display window of Figs. 6A and 6B, in accordance with a preferred embodiment of the present invention;

Fig. 8 is a graphic representation of a computer display screen illustrating elements of a graphic user interface presented in a window on the screen, in accordance with a preferred embodiment of the present invention;

Figs. 9A and 9B are graphic representations of the computer display screen of Fig. 8, showing changes in the shapes and sizes of elements of the graphic user interface as the size and shape of the window are varied, in accordance with a preferred embodiment of the present invention;

Figs. 10A and 10B are schematic illustrations of a computer animation scene, useful in understanding the operation of virtual cameras for rendering computer animation sequences from different viewpoints, in accordance with a preferred embodiment of the present invention;

Figs. 11A and 11B are graphic representations of the computer display screen of Fig. 8, showing changes in rendering of the scene responsive to selection and movement of the virtual cameras of Figs. 10A and 10B, in accordance with a preferred embodiment of the present invention;

Fig. 12 is a graphic representation of a 3D joystick, shown on the computer display screen of Fig. 8, for use in controlling the operation of the virtual cameras of Figs. 10A and 10B;

Figs. 13A and 13B are graphic representation of a computer display screen, illustrating a graphic user interface used in conjunction with an associative visual search engine, in accordance with a preferred embodiment of the present invention;

Fig. 14 is a flow chart illustrating the operation of the visual search engine whose user interface is shown in Figs. 13A and 13B, in accordance with a preferred embodiment of the present invention;

Fig. 15 is a block diagram illustrating a system architecture for transmission of a computer animation program over a computer network, in accordance with a preferred embodiment of the present invention;

Fig. 16 is a block diagram illustrating a system architecture for transmission of an animated message over a computer network, in accordance with another preferred embodiment of the present invention;

Fig. 17A is a graphic representation of a computer display screen illustrating the preparation of an animated message, in accordance with a preferred embodiment of the present invention;

Fig. 17B is a flow chart illustrating a method for preparation of the animated message of Fig. 17A, in accordance with a preferred embodiment of the present invention;

Fig. 18A is a graphic representation of a computer display screen illustrating playback of the animated message of Fig. 17A by a recipient thereof, in accordance with a preferred embodiment of the present invention; and

Fig. 18B is a flow chart illustrating a method for playback of the animated message of Fig. 18A, in accordance with a preferred embodiment of the present invention.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

Reference is now made to Fig. 1, which is a schematic representation of an animation generator, preferably a computer system 40, for use in producing and displaying computer animation sequences, in accordance with a preferred embodiment of the present invention. Computer 40 comprises a console 42, including a processor 44 and memory 46, preferably comprising both RAM and disk memory, as are known in the art. The computer also includes user input devices, such as a keyboard 48 and pointing device 50, for example, a mouse.

Computer 40 includes image rendering software, such as RenderWare, described above, which enables processor 44 to produce animated computer graphic images 51, which are seen on a display 52. Animated images 51 are preferably produced in response to an animation program, run by processor 44, preferably a Scene Manager program, which generates images based on 3D Smart Objects, as described herein. Preferably, the images are accompanied by sound, which is produced by an audio device 56, such as a speaker, coupled to the processor.

Computer 40 is preferably coupled via a modem 54 to a computer network 60, preferably the Internet, and sends and receives animation files over the network, as described below. Further preferably, the computer includes browser software, for example, a VRML-compatible browser, suitable for viewing animations conveyed over the network.

Fig. 2 is a graphic representation of display 52, showing an animated image produced by processor 44 based on a 3D Smart Object, representing a young female character 64. The animated character, rendered as a 3D image, is seen to walk out on screen and take a position in front of an application window 66. The application that drives window 66 is generally unrelated to the Scene Manager animation program and to the Smart Object, and the application continues to run substantially uninterrupted by the animated image. Character 64 obscures only the portion of window 66 that the character directly covers, while the rest of the window remains normally active and visible, as though character 64 were absent. The software structure and methods used in producing the animation and overlaying the character on the window are described below.

As described above, the term "Smart Object" as used herein, in accordance with preferred embodiments of the present invention, refers to an object, or module, comprising both the graphic/mathematical description of an image element and embedded ancillary information about the element. Such ancillary information preferably comprises procedures, which describe attributes, motion and behavior of the element, as described hereinbelow, and may also include textual reference data. These procedures are preferably in the form of scripts, i.e., lists of instructions that are read and executed by the Scene Manager. "3D Smart Objects" preferably include attributes of shape and motion that are used in rendering 3D images of the corresponding image elements.

Figs. 3A and 3B are graphic representations of display 52, showing animated images produced by processor 44, which illustrate further aspects of the behavior of 3D Smart Objects.

In Fig. 3A, young female character 64 has walked out onto the screen, while a male character 68 stands at the opposite side. The Smart Object corresponding to male character 68 includes animation scripts that define the character's behavior in response to young female characters. These scripts are triggered by the presence of female character 64 within a predetermined region of proximity to male character 68. For character 68, the region of proximity is determined to be the entire area of display 52.

As shown in Fig. 3B, in response to the proximity of female character 64, the scripts that are triggered cause male character 68 first to look at the female character and then to jump excitedly. The scripts that determine these behaviors are contained entirely within the Smart Object, so that the response of male character 68 to female character 64 is generated substantially autonomously by the Smart Object. There is no need for a script or program external to the Smart Object to create the behavior, although it is possible to manipulate Smart Objects using such external scripts.

Furthermore, although character 68 exhibits, in this case, behavior that would be characteristic of a young male, the behavior may be transferred in a modular fashion from one Smart Object to another, independent of their physical appearance and other characteristics. Thus, for example, any or all of the behavioral routines that make up the young, male "personality" assumed by character 68 may be transferred to another character having the appearance of an old man or woman, or of a robot. It is sufficient to transfer the specific routines that are triggered by the proximity of female character 64 to cause the other character to look at the female character and jump excitedly. Unlike animation programs known in the art, no other reprogramming or adjustment of the animation of the other character is generally needed.

3D SMART OBJECTS

Fig. 4 is a block diagram illustrating the data structure of a 3D Smart Object 70, in accordance with a preferred embodiment of the present invention. The hierarchical structure of Smart Object 70 is what makes possible the autonomy and modularity of Smart Objects, as described above with reference to Figs. 3A and 3B.

Smart Object 70 comprises a skeleton, including surfaces 76, a hierarchy of organs 78 and behaviors 74. Organs 78 are made up of sub-organs 90, which are in turn made up of 3-D objects 86, comprising 3D geometrical shapes connected by joints, as are known in the art. The term "organ," as used herein, refers to a coherent group of 3D objects 86 that form a part of the physical representation of the Smart Object. Surfaces 76 and surface details 94, as described below, relate to corresponding 3D objects 86 and define these objects' surface appearance when the Smart Object is rendered to display 52. While the skeleton of Smart Object 70 is common to all Smart Objects of a given class, for example, humanoid Smart Objects, each Smart Object in the class has its own specific geometry 72, defined by the set of 3D objects 86 and surface details 94 belonging to the Smart Object.

Behaviors 74 include 3D animations 88 of the Smart Object, defining motions of organs 78 and sub-organs 90 at joints thereof. The behaviors may also operate on surfaces 76, for example, by changing the color of a selected surface under certain predefined circumstances. Animations 88 are preferably grouped as mandatory animations 80, conditional animations 82 and optional animations 84. The meanings of these groupings are discussed below. Preferably, behaviors 74 belonging to a given Smart Object are chosen and programmed so that the Smart Object exhibits a desired personality, such as the "young male" personality of character 68 (Figs. 3A and 3B). In the context of the present patent application and in the claims, the term "personality" used in reference to a Smart Object or an animated character associated therewith refers to a set of behaviors that cause the character to move and otherwise exhibit reactions and expressions in a manner that is reminiscent of the behavior of a real character, typically a human being, of a certain type.

Although the hierarchies of geometry 72 and behaviors 74 are separate, there is a complete correspondence between the set of 3D animations 88 and the set of 3D objects 86. In other words, each animation 88 comprises a mathematical description that dictates operations to be performed in a predetermined way on certain objects 86 in the geometrical hierarchy. Each animation 88 operates on certain organs 78 and sub-organs 90 and the joints connecting them.

The same hierarchies of geometry and behavior are used for all Smart Objects of a certain type, for example, humanoid characters or four-legged creatures, or inanimate (but possibly animated) objects, such as tables or chairs. Therefore, 3D objects 86 and 3D animations 88 may be exchanged freely, in modular fashion, among all Smart Objects of the type, i.e. Smart Objects derived from the same skeleton, whether at the level of individual objects and animations or at the levels of organs 78 or groups of animations 80, 82 and 84. By transferring such groups of animations, the personality of a Smart Object corresponding to a humanoid character may be transferred to or inherited by another Smart Object. Similar conventions apply to surfaces 76 and to sub-organs 90, as described below.

The hierarchical structure of Smart Object 70 is constructed using 3D animation software tools known in the art, for example, LightWave 5.0 software, as described above. The full modularity of the behavior of Smart Objects, however, is not achieved by animation objects and systems known in the art, and files corresponding to Smart Objects that are created initially using LightWave, for example, must be converted to Smart Object form. Smart Objects have all the features of an object-oriented programming system. They enable animated characters to be created by transfer of physical and/or behavioral characteristics between different Smart Objects or between a Smart Object and a library of such characteristics, while leaving the hierarchy intact. New Smart Objects and classes of Smart Objects may be created by inheritance from existing Smart Objects, including multiple inheritance, as is known generally in areas of conventional object-oriented programming.

Grouping of 3D objects 86 into organs 78 and sub-organs 90 serves the purpose of enabling objects to be exchanged among Smart Objects in such a way that an appearance of natural motion of the animated image element corresponding to the Smart Object is preserved, and that suitable proportions are maintained between closely-related 3D objects 86 and groups of such objects. It will be understood that sub-organs 90 are themselves treated as organs, and furthermore, that a sub-organ may contain other sub-organs, below it in the hierarchy. Thus, for example, the hands of a humanoid skeleton are defined as one sub-organ, so that both hands are exchanged as a single unit. The basic hierarchy of organs 78 and sub-organs 90 and 3D objects 86 in skeleton 70 of a humanoid Smart Object is preferably defined by the following file structure, wherein the term "name" in each case refers to the reference name of a Smart Object character with which the organ, sub-organ or object is associated:

```
basenull
    cameranull
    name_abdomen
        name_body
            name_an_back
            name_hand_lt_01_upperarm
                name_hand_lt_02_lowerarm
                    name_hand_lt_03_palm
                        name_hand_lt_04_fingers
                        name_hand_lt_05_thumb
            name_hand_rt_01_upperarm
                name_hand_rt_02_lowerarm
                    name_hand_rt_03_palm
                        name_hand_rt_04_fingers
                        name_hand_rt_05_thumb
            name_neck
                name_head
                    name_an_head
                    name_an_mustache
                    name_brow_lt
                    name_brow_rt
                    name_eye_lt
                    name_eye_rt
                    name_eyelid_lt
                    name_eyelid_rt
                    name_mouth
```

name_ponytail_01
name_ponytail_02
name_ponytail_03
name_leg_lt_01_thigh
name_leg_lt_02_shin
name_leg_lt_03_shoefront
name_leg_lt_04_shoeback
name_leg_rt_01_thigh
name_leg_rt_02_shin
name_leg_rt_03_shoefront
name_leg_rt_04_shoeback
name_tail_01
name_tail_02
name_tail_03

The above hierarchy of organs 78 and sub-organs 90 is largely self-explanatory, but includes a number of unique features. The abbreviation "an" in certain of the above organs, for example, "name_an_head," refers to an "accessory null," for attaching an external accessory, such as a hat, in the case of name_an_head, to the associated organ. The expression "cameranull" refers to an optional connection point for attaching a "virtual camera" to the Smart Object, so that an image of an animation scene is rendered as seen from the point of view of the Smart Object. The expression "basenull" refers to a dummy joint, at which one Smart Object may be connected to another if desired.

Surfaces 76 comprise mathematical descriptions of the external appearance of organs 78, sub-organs 90 and 3D objects 86. A surface can include geometrical items belonging to several organs. For example, a shirt surface includes geometrical data belonging both to the body and to the arms. When the image of Smart Object 70 is rendered to computer screen 52, the surfaces are mapped onto the 3D-objects by suitable rendering software, such as RenderWare, described above, so that each 3D object 86 is shown on the screen with a surface appearance determined by the corresponding sub-surfaces and surface details.

Like 3D objects 86, surfaces 76 and surface details 94 may be exchanged and inherited freely among Smart Objects of the same type. The term "inherit" is used in the present patent application and in the claims in the sense of the term that is known in the art of object-oriented programming, to mean that 3D objects and surfaces may be taken from one Smart Object and used directly in creating a new Smart Object without additional adaptation. Animations 88 can similarly be exchanged and inherited, as described below. Preferably, surfaces 76 of a humanoid Smart Object include:

name skin
name body top (generally the shirt)
name body bottom (generally skirt or pants)
name hair
name eyes (iris)
name shoes
name lips

Surface details 94 are defined so as to enable the user to edit the colors, shapes, textures and other surface features that are mapped onto 3D objects 86 when the Smart Object is rendered to display screen 52. For example, surface details 94 of a humanoid Smart Object preferably include:

name eyes pupils
name skin face
name skin face nose
name skin face ears

and so forth. Surface details 94 may further include a logo, for example, text or an insignia, to be displayed on the shirt of a character rendered on screen 52

Every Smart Object 70 includes a set of mandatory animations 80 that define the basic behaviors of the animated character associated with the Smart Object. Preferably, mandatory animations 80 for a humanoid skeleton include the following motions:

1. Appear.
2. Disappear.
3. Advance (three variations - fast, slow and "slow motion").
4. Idle (3 variations).
5. Talk.
6. Positive.
7. Negative.
8. Greeting (2 variations).
9. Don't know (perplexity or uncertainty).
10. Yes.
11. No.
12. Special.

Each of mandatory animations 80 comprises a mathematical description of motion of the Smart Object corresponding to a predetermined type of behavior, generally indicated by the name of the animation. "Idle" refers to behavior assumed by an animated character associated with Smart Object 70 when the Smart Object has not been selected or activated to modify its position, orientation or other attributes for a predetermined period of time. Idle behavior of character 68 (Figs. 3A and 3B) could include, for example, crossing his arms and tapping his foot if female character 64 does not appear. Each Smart Object also includes a "Special" animation, specific to its character type. Preferably, Smart Object 70 also includes optional animations 84. The optional animations can include variations on mandatory animations, for example, walking in a way that a viewer of the animation would associate with "cool" or "sexy" behavior, in addition to the three mandatory "advance" animations. Additionally or alternatively, optional animations 84 can include animations specific to a certain type of Smart Object. For example, a soccer player Smart Object can have a "kicking" animation. Preferably, Smart Object 70 also includes a set of conditional animations 82, which define the response of the Smart Object to "sensitivity conditions," i.e., events or interactions that trigger the animations. Such events and conditions may include, for example, proximity to or collision with another object (which may or may not be a Smart Object) or selection of the Smart Object by a user. The conditional animations preferably comprise scripts, which call for predetermined mandatory and/or optional animations in response to sensitivity conditions. It will thus be understood that the reaction of young male character 68 to female character 64, shown in Fig. 3B, is controlled by an appropriate conditional animation within the Smart Object associated with male character 68, responsive to the proximity of the female character thereto. As a further example, selection of character 68, by pointing to the character and/or clicking on the character using mouse 50 (Fig. 1), preferably triggers a conditional animation that causes the character to turn to face the user or, alternatively or additionally, to change color, smile, wave or exhibit any desired behavior.

It will be understood that conditional animations 82 are as much a part of Smart Object 70 as are the features of the Smart Object's geometry 72. Taken together with mandatory animations 80 and optional animations 84, the conditional animations define rules of behavior that impart to the Smart Object a substantially autonomous "personality." Although the Smart Object may be animated using a program, such as a script, that is external to the Smart Object, no such external script is required to engender the autonomous motion and response animation of the Smart Object. When an external script is used, it is generally sufficient that the script include lines for the Smart Object characters to speak, positions that they are to assume in a scene, and calls to animations 80 and 84 that the Smart Objects are to undertake.

As described above, all 3D animations 88 defining behavior 74 of Smart Objects of a certain type, such as humanoid characters, correspond strictly to the skeleton of organs 78 and sub-organs 90 belonging to the type. As long as the skeleton is unchanged, each animation 88

operates on specific organs 78, sub-organs 90 and 3D objects 86, irrespective of their shapes and sizes. In consequence, the behaviors of Smart Objects are entirely modular, just as the elements of their geometry and physical appearance are. Smart Object behaviors are themselves objects that may be exchanged and inherited freely between characters, without the need for reprogramming when the behaviors are changed or exchanged.

Fig. 5 is a flow chart illustrating a method for animating one or more Smart Objects 70 in an animation scene, based on autonomous behavior 74 of the Smart Object, in accordance with a preferred embodiment of the present invention. As described above, the method makes use of a "Scene Manager" program to coordinate the actions of Smart Object 70 and other objects in the scene. The Scene Manager reads geometrical data 72 and animations 88 from each Smart Object 70, and uses them as input data for an executable program module, controlled by the Scene Manager, including a moving geometrical representation of the Smart Object. Preferably, the Scene Manager maintains a table indicating the status of all the Smart Objects in the scene, including their positions and sensitivities, i.e., conditions and events capable of triggering conditional animations 82 belonging to each of the Smart Objects. Further preferably, the Scene Manager also includes image rendering routines, for example, executable code belonging to RenderWare, as described above, for rendering an image based on the geometrical representation of the Smart Object to display 52.

The Scene Manager preferably runs as a plug-in software module together with network browsing software known in the art, such as Netscape "Navigator," version 3.0 or higher, or Microsoft "Internet Explorer," version 3.0 or higher. Alternatively or additionally, the Scene Manager can run as a stand-alone application, preferably in a windows-based operating system, most preferably a Microsoft "Win32"-type operating system, as is known in the art.

As shown in Fig. 5, after initial animations have been selected for each Smart Object in an animation scene, each cycle of the method begins with a "tick," or incremental advance, of a system clock that drives the animation scene. Each Smart Object in the scene is indicated by an index I. At each tick, the Scene Manager reads the selected animation from each of the Smart Objects in sequence, either the initially-selected animation or another animation selected later, as described below. The Scene Manager activates the program module corresponding to the Smart Object to perform the selected animation. The animation causes the program module to alter the status of the geometrical representation of the Smart Object, for example, its position, orientation, expression or other aspects of its physical appearance. The program module, in turn, sends a message back to the Scene Manager indicating its new status, and updating the table of Smart Object status accordingly. This process continues until all of the Smart Objects in the scene have been activated. The Scene Manager then renders an image of the Smart Objects, based on the new status of the geometrical representations.

After all of the Smart Objects have been activated, the Scene Manager checks the updated status table of the Smart Objects in the animation scene, to determine whether any

sensitivity conditions have been created. Preferably, the Scene Manager also checks whether a user has provided any input that would have an impact on any of the Smart Objects, for example, selecting one of the Smart Objects using pointing device 50 (Fig. 1). If no new sensitivity condition has been created and no user input has been provided, the system clock is advanced, and the cycle repeats.

If the Scene Manager detects that a new sensitivity condition has been created for any of the Smart Objects, however, the Scene Manager informs the Smart Object of the change and triggers the appropriate conditional animation of the Smart Object. The clock is then advanced, and the cycle repeats. Thus, referring again to the example shown in Fig. 3B, when the Scene Manager detects that female character 64 has moved into sufficient proximity to male character 68, the Scene Manager triggers an appropriate "excited" animation of the male character. The subsequent behavior of the male character is controlled by the Smart Object itself, until the Scene Manager detects a significant change in the position of the female character or until some new sensitivity condition arises.

The method illustrated by Fig. 5 is simplified for the sake of clarity of explanation. It will be appreciated that the Scene Manager may fulfill other functions besides those described above. For example, as described above, the Scene Manager may apply an external animation script to the Smart Object. Similarly, other animation and rendering programs may be used to animate Smart Objects. It will also be appreciated that the method described above may be applied to a single Smart Object, as well as to a plurality of interacting Smart Objects.

TRANSPARENT 3D GRAPHIC OVERLAYS

Referring back to Fig. 2, it is observed that 3D character 64 is overlaid on application window 66, while covering only that part of the window that is directly behind the character. Character 64 appears to cast a shadow 98 on window 66, but the shadow is preferably at least partly transparent, and the window text 100 is seen through the shadow.

Character 64 is superimposed on window 66 using a method referred to herein as "Transparent 3D." Although transparent image overlays, commonly called sprites, are known in the art, they are generally limited to images produced by 2D rendering and allow only 2D motion of the characters or objects in the images. A sense of depth is achieved only when 2D sprites are overlaid one upon another. By contrast, transparent 3D image element 64 in Fig. 2 is fully three-dimensional in its rendering, as well as in the range of animated motions that it can undertake, as described below.

Figs. 6A and 6B are graphic representations of computer screen 52, further exemplifying the principles of Transparent 3D graphic overlays, in accordance with a preferred embodiment of the present invention. In these figures, a 3D Smart Object corresponding to a circling airplane 110 is used to generate a Transparent 3D overlay image of the airplane, overlaid on an application window 112, for example, an Internet Web page. Airplane 110 circles continuously above window 112 along a 3D path indicated by arrows 116. It will be observed that as the

airplane "flies" from its position in Fig. 6A to that in Fig. 6B, the shape of the airplane, as reflected by a border 114 circumscribing the airplane, changes responsive to the changing angle and perspective distance from which the airplane is seen. Yet the only part of text 100 in window 112 that is obscured is that which is directly behind the image of airplane 110, as defined by border 114.

Fig. 7 is a flow chart illustrating a method for creating Transparent 3D image overlays, in accordance with a preferred embodiment of the present invention. The method is based on the use of irregular-shaped, i.e., non-rectangular, windows on computer display screen 52, particularly as defined and supported by Microsoft Win32-type operating systems, such as Windows 95 and Windows NT 4.0. The method is also applicable, however, to computer animation systems using other operating systems that support irregular-shaped windows.

To create a Transparent 3D overlay, a 3D object is chosen, preferably a 3D Smart Object, such as that corresponding to airplane 110 in Figs. 6A and 6B, as described above. The behavior, particularly the motion, of the object is determined, and a suitable 3D view, generally comprising a viewpoint and view angle, are chosen for rendering the object. A 3D rendering engine, for example, the rendering module of the Scene Manager, preferably using RenderWare software, as described above, renders an image of the object to an off-screen bitmap. Within this bitmap, the rendering engine or another software module identifies which pixels have been written to, i.e., which pixels are occupied by the image of the object, and which have not. This information is used to create a mask exactly covering the pixels that were written to, so as to delineate a border 114 of the image and a transparent area outside the border. The mask, surrounded by border 114, thus defines a window shape that exactly fits the image of the 3D object.

This window definition is passed to the operating system running on processor 44, and is used by the operating system to shape a window for the object that exactly fits border 114, making use of a capability of Win32 and other, similar operating systems of creating irregular-shaped windows. The operating system thus draws a window on screen 52 that contains the image, for example, the image of airplane 110. This window is overlaid above other windows on the screen, such as application window 112. Since the window containing airplane 110 exactly fits the shape of border 114, the remainder of application window 112 is substantially unobscured.

The shape and position of the window containing the image of the 3D object remain unchanged for as long as the object's behavior and position and the 3D view for rendering the image of the object do not change. In general, however, Transparent 3D image overlays comprise animated images, which do change, substantially continuously. Thus, as shown in Fig. 7, the Scene Manager checks regularly to determine whether the 3D view or the object's behavior has changed. For example, a sensitivity condition may have triggered a change in the behavior of a Smart Object, as described above with reference to Fig. 5. In such a case, a new

3D view and/or new object behavior is determined, and the process of drawing a suitably-shaped window for the new image of the 3D object is repeated. Similarly, if the 3D view and object behavior are unchanged, but the animated object is in motion, as shown, for example, in Figs. 6A and 6B, then the window containing the image of the object must be continually redrawn. In either case, a new image of the object is rendered to the off-screen bitmap, overwriting the previous bitmap, and the new bitmap is used in re-shaping the object window, as described above.

Whenever the shape and/or position of the object window changes, a different portion of application window 112 will be exposed. This newly-exposed portion of display 52 is quickly redrawn, in accordance with the application driving window 112. Throughout the process of drawing and displaying the Transparent 3D overlay image of airplane 110, the application of window 112 continues to run substantially uninterrupted.

Although sprite overlays are known in the art, as described above, none has been capable of 3D image overlays with real-time animation, and none has combined 3D image rendering with Win32-type irregular-shaped windows. Transparent 3D objects allow complete freedom of animation, based on manipulation of geometrical models, including arbitrary changes in the objects' motion, behavior and rendering. Animated image overlays known in the art use only pre-programmed animations, allowing very limited variations in the appearance of the overlay image.

USER INTERFACE

Reference is now made to Fig. 8, which is a graphic representation of display 52, illustrating a computer animation application window 140, in accordance with a preferred embodiment of the present invention. Window 140 includes an animation area 142, which is contained within a frame 136 including graphic user interface controls 138. A computer animation sequence is displayed in animation area 142, including female character 64 and a host character 160, which preferably correspond to respective 3D Smart Objects. The sequence from which Fig. 8 is taken represents a television "talk show," wherein host 160 "interviews" character 64, based on a script and on the appropriate behaviors of the associated Smart Objects, as described above. The script of the talk show appears, line by line, in a dubbing strip 154, while characters 64 and 160 are also seen and can preferably also be heard to speak their appropriate lines.

Frame 136 preferably comprises a 3D still image, having a transparent region defining animation area 142. As shown in Fig. 8, the transparent region is preferably non-rectangular. Frame 136 is rendered as an overlay, above animation area 142, so that the animation is seen on display 52 inside a correspondingly non-rectangular "window," defined by the frame. The irregular shape of the animation area adds interest to the scene, and also serves to emphasize the functionality of user controls 138 in relation to the scene. Using this method, animation area 142 may be given any desired shape, and frame 136 may comprise any suitable still image. For

example, the frame may have the appearance of an auditorium, and the animation area may be shaped as a stage.

User interface controls 138 include an on-screen joystick 144, slider 156, selector push buttons 146, a start button 152, pause button 150, and user response button 148. These controls are used primarily to control the operation of virtual "cameras," defining the 3D view from which the scene in animation area 142 is rendered, as described in greater detail below. User controls 138 are preferably themselves 3D graphic objects, which are rendered as part of the image of frame 136. Thus, the sizes and shapes of the user controls and the distances between them may be altered as desired to enhance their functionality and ease of use.

Figs. 9A and 9B are graphic representations of display 52, illustrating changes in the sizes and spacing of user controls 138, responsive to scaling of animation application window 140, in accordance with a preferred embodiment of the present invention. In Fig. 9A, window 140 has been scaled down vertically. In consequence, the height of slider 156 is reduced to fit the window dimension. Similarly, in Fig. 9B, the width of window 140 is reduced, and the spaces between push-buttons 146 are concomitantly narrowed. Dubbing strip 154 is narrowed at the same time.

The variable sizes and spacing of user controls 138 in Figs. 8, 9A and 9B enable a user to operate the controls by means of pointing device 50 with greater ease and accuracy. The variations in size and spacing of the controls also give window 140 a more pleasing aesthetic appearance. By contrast, graphic user interface controls known in the art generally have fixed sizes and spacings, regardless of the scaling of the window in which the controls are contained.

Figs. 10A and 10B are schematic diagrams showing the positions of 3D objects in a 3D animated scene 170, and illustrating the operation of virtual cameras 172, 174 and 176 in rendering the scene to image area 142 of display 52 (shown in Fig. 8), in accordance with a preferred embodiment of the present invention. Fig. 10A is a top view of scene 170, and Fig. 10B is a side view. The 3D objects comprise Smart Objects 64' and 160', which are rendered respectively as images of characters 64 and 160 in image area 142, and furniture objects 162' and 164', rendered respectively as couch 162 and table 164 in the image.

Each of virtual cameras 172, 174 and 176 corresponds to and defines a 3D view of scene 170. Each camera is selected by pressing an appropriate one of push buttons 146. In a preferred embodiment of the present invention, five virtual cameras are used, corresponding to the five selection buttons 146, but for simplicity of illustration, only three cameras 172, 174 and 176 are shown in Fig. 10A, and only two of the cameras are shown in Fig. 10B. Thus, the view rendered in Fig. 8 corresponds generally to that seen by camera 176.

As illustrated in Figs. 10A and 10B, each of cameras 172, 174 and 176 is capable of a range of virtual motions, changing the cameras' viewpoints (spatial positions), zoom angles and orientations. For example, as shown in Fig. 10A, camera 174 may be panned about its viewpoint 175, as indicated by an arrow 178. The camera's zoom angle, indicated by an arrow 180, may be

varied between wide-angle and close-up views. The zoom is preferably controlled by shifting slider 156.

Camera 174 may also be moved along a straight or curved path in space, as shown, for example, by an arrow 182, indicating that the camera is translating from a starting point 184 to an end point 186. Preferably, these paths are predefined, to give a well-controlled range of camera positions, analogous to movement of real video cameras in an actual television studio. When camera 174 is selected, a user initiates motion of the camera along the path indicated by arrow 182 by pressing start button 152. To stop the camera's motion, the user presses pause button 150.

Other camera motions are illustrated in Fig. 10B. For example, camera 172 tilts up and down about its viewpoint 195, as indicated by an arrow 190. A point in space may be chosen, such as a point 196, and camera 176 may be controlled to revolve about the point in an orbit beginning at a position 193, for example, and continuing as indicated by an arrow 194. The cameras' tilt, pan and revolution are preferably controlled using joystick 144, as is further described below. Likewise, camera 172 may be made to close in toward point 196, along a linear path indicated by an arrow 192, and may similarly be made to pull out, away from scene 170, along an opposite path. Point 196 may be chosen to be anywhere in the 3D space of scene 170, including in or adjacent to one of the objects in the scene.

It will be understood that although certain of these camera motions are described herein with reference to a specific one of cameras 172, 174 and 176, the motions may be applied equally to all of the cameras, as well as to additional virtual cameras, not shown in the figures, which may be provided.

Figs. 11A and 11B are graphic representations of display 52, illustrating the effects of the camera motions and camera selection described above on images rendered in image area 142. Fig. 11A shows a view corresponding to camera 174, whose general position and zoom angle are shown schematically in Fig. 10A.

Fig. 11A also includes a schematic, outline representation of camera 172, superimposed on the animated scene. Ordinarily, this representation of the camera is not shown in the animated image area. The user may choose to display the camera outline, however, to assist in visualizing and shifting the camera's coordinates. Fig. 11B shows a view "captured" by camera 172 in the position shown in Fig. 11A, i.e., Fig. 11B shows the view of camera 172, after the camera has closed in toward point 196 (Fig. 10B), as described above.

It will be appreciated that the user interface described above, comprising on-screen controls 138 and virtual cameras 172, 174 and 176, allows a user of computer animation system 40 to choose and manipulate the viewpoints from which scene 170 is rendered with substantial precision and flexibility and with relative ease. In interactive computer animation systems known in the art, a user is generally limited either to a fixed viewpoint, giving relatively little flexibility, or must maneuver a virtual camera by "free navigation" in 3D space. Free navigation is difficult

to control, since pointing devices and on-screen controls typically have only two degrees of freedom, while the virtual camera has three. In accordance with the principles of the present invention, however, navigation in 3D space is based on selection from among predefined viewpoints, on zoom, pan and tilt angles centered on these viewpoints, and on movement of the viewpoints along predetermined, linear and arcuate paths. The viewpoints and paths are preferably pre-programmed and supplied to a user as part of an animation software package. Alternatively or additionally, the user may independently designate viewpoints and paths, preferably using pointing device 50 (Fig. 1).

Fig. 12 is a graphic representation showing a portion of display 52, illustrating the operation of on-screen joystick 144. The joystick is operated by selecting it, preferably using pointing device 50, and then dragging it in a desired direction by a desired amount. In Fig. 12, joystick 144 has been dragged to the left.

Joystick 144 has two alternative functions: Either pan and tilt of one of the virtual cameras that has been selected, or revolution of the selected camera about a point in scene 170. When the joystick is used to control camera revolution, left-right movement of the joystick drives the revolution in a horizontal plane, while up-down movement drives the revolution in a vertical plane. Preferably, user response button 148 is used to toggle between the tilt/pan and revolutionary movements of the selected camera.

ASSOCIATIVE VISUAL SEARCH

Figs. 13A and 13B are graphic representations of display 52, schematically illustrating a user interface window 200 associated with a visual search engine, in accordance with a preferred embodiment of the present invention. The search engine is preferably used to search through a library of animations and/or animated objects, most preferably, 3D Smart Objects, and it is particularly useful in creating computer animations. But it may, alternatively or additionally, be used in searching through databases of still images, particularly including background and border images for use in creating computer animations.

Fig. 14 is a flow chart, illustrating a method of operation of the visual search engine, in accordance with a preferred embodiment of the present invention.

Each item in the library or database to be searched must have at least one, and preferably a plurality, of keywords associated therewith. The keywords preferably describe different aspects of the item, such as its name and type, subject matter, character, appearance, style, color, size, etc. The keywords for each item are hierarchically ranked, with one of the keywords chosen to be the premier keyword for the item. Preferably, each item also has a unique name by which it is identified.

As shown in Fig. 13A, to begin a search, a user preferably enters a starting keyword or name in an entry field 202 in window 200, and then selects a "search" button 204. Based on the starting keyword or name, a primary search is conducted of the library or database. The search engine finds keywords or names associated with items in the library that match the keyword that

is entered, and displays the matching keywords or names in a match field 218. Preferably, the search engine includes a thesaurus, which enables the engine to select and search for other keywords synonymous with the starting keyword, particularly when the starting keyword is not found among the keywords of the library or database.

Once the primary search is complete, the results are displayed in an image area 210 of window 200. An image of an item, in this case a bouquet 208, that most closely matched the starting keyword is displayed in a central portion 216 of the image area. Images of other items that matched the starting keyword are also displayed within area 210, in a generally concentric arrangement around central portion 216. Images of items that matched the starting keyword more closely, such as an image of a flower 212, are displayed in closest proximity to central portion 216, while those that matched less closely, such as a banner 214, are displayed farther away.

In the context of the present patent application and the claims, an item is considered to match closely a keyword that is searched for when the keyword searched for occupies a high rank in the hierarchy of keywords associated with the item. Generally, the higher the rank of the keyword searched for is in the hierarchy, the closer is the match. It will be appreciated, however, that in other preferred embodiments of the present invention, the principles of the search engine described herein and particularly the graphic user interface associated therewith, as exemplified by window 200, may be applied using other search algorithms known in the art.

Alternatively, the user may begin a search by selecting a "random" button 206. In this case, the search engine performs a random search through the database or library and presents a random assortment of images in window 200. This random search feature is useful when the user wishes to browse through the library or database so as to intuitively seek associations, rather than using the more directed keyword search.

Once the primary search or random search has been performed, the user visually reviews the images in window 200. Preferably, as shown in Fig. 13B, when the user points to one of the images 222 with a cursor 220 (without necessarily selecting the image), the image, in this case a man with a bouquet, is enlarged. If image 222 represents an animated object, then when the user points to the image, an animation associated therewith is run. Similarly, if image 222 represents a 3D object, then the enlarged image is preferably rendered as a 3D image.

It will thus be appreciated that the properties of the search engine exemplified by Fig. 13B make it particularly useful for searching through a library of Smart Objects, and most preferably 3D Smart Objects. On the other hand, it will also be understood that the unique aspects of the search engine and the user interface associated therewith will be useful for searching through other image libraries and databases, as well.

Returning to the flow chart in Fig. 14, after the primary or random search results are displayed, the user selects an item shown in image area 210, for example, image 222, preferably by "clicking on" the item using pointing device 50. The selected item is displayed in central

portion 216 of window 200, instead of the image that was displayed there previously. The search engine conducts a secondary search, preferably based on the principles of keyword matching described above in reference to the primary search. Since the item associated with image 222 will generally have a different set of keywords from that used in the primary search, the secondary search will find new items in the library or database. The new items are displayed in image area 210 in the same concentric arrangement as described above.

The user reviews the new items found in the secondary search. If the user finds a desired item, for example, an object to be used in an animation, that item is selected, and the search ends. If the desired item is not found, the user preferably selects a new one of the items in image area 210 that has some association with a type of item or concept that the user has in mind. This new item then serves as the starting point for another search, like the secondary search described above. This iterative cycle of association (by the user) and search (by the search engine) may be repeated indefinitely until the desired item is found. It will be appreciated that an associative search of this sort is useful both for finding a specific item or type of item in a library or database and for browsing through the library or database in search of a new, as-yet-unformulated idea.

TRANSMITTING ANIMATION OVER A NETWORK

Reference is now made to Fig. 15, which is a block diagram illustrating a system architecture 231 for transmission of an interactive computer animation over a computer network 236, preferably over the Internet, in accordance with a preferred embodiment of the present invention. The animation, for example, a talk show, as described above with reference to Fig. 8, is produced by a server 230 and is distributed over network 236 to multiple clients 232 and 234. The clients include viewers 234, who use computers such as computer 40, shown in Fig. 1, to receive the animation from the server and view the animation on local display 52. The clients may also include participants 232, who not only view the animation, but also participate in the animation, for example, by controlling respective avatars in the talk show. Preferably, the animation is based on Smart Objects, most preferably 3D Smart Objects, as described above.

For the sake of clarity of explanation, the following description of Fig. 15 assumes that the animation transmitted over network 236 comprises a talk show, as illustrated and described above. It will be appreciated, however, that the principles of the present invention, as described with reference to the talk show, may similarly be applied to transmit other types of animated information, entertainment and advertising programs over a network. Thus, the animation may comprise, for example, a game show, a soap opera, or an interactive shopping program, which are preferably realized using Smart Objects.

In order to transmit the talk show of Fig. 8 to one of viewers 234, the Smart Objects representing female character 64 and host 160 are transmitted by server 230 over network 236 and are preferably stored in local memory 46 of the viewer's computer 40. The Smart Objects need be transmitted only once. In fact, assuming host 160 appears in a series of talk show

programs transmitted by server 230, the host Smart Object may be recalled from local memory and need not be retransmitted over the network for each new talk show transmission.

As a result of this use of Smart Objects in the show, the time and bandwidth needed to transmit each show over network 236 is substantially reduced. All that is necessary is to transmit an animation script. The animated Smart Objects speak their lines and move between appropriate, predetermined positions within an animation scene, such as scene 170 shown in Figs. 10A and 10B, in accordance with the script. On account of the substantial autonomy of the Smart Objects, substantial portions of the appearance, gestures and expressions of the characters are filled in by the Smart Objects themselves, and need not be transmitted over the network. The animated talk show is rendered to display 52 by a rendering engine running on computer 40, as described above.

Preferably, the Smart Objects and the animation script are compatible with animation interchange tools and conventions known in the art, most preferably with VRML 2.0, as described above. Such compatibility allows viewers 234 to receive and view animations using commonly-available Internet browsing software, with suitable animation plug-ins, as are known in the art.

Further preferably, viewers 234 are provided with user interface controls, such as on-screen controls 138 shown in Fig. 8. As described above, controls 138 are used to vary the selection and viewpoints of virtual cameras 172, 174 and 176 (Figs. 10A and 10B), so that the viewer may, for example, choose between the wide angle view of Fig. 11A and the close-up of Fig. 11B. User controls may also be provided to change other aspects of the animation, such as lighting or background.

Viewers 234 have only limited opportunity, however, to interact with and affect the animation script. Such interaction is typically limited to indicating a selection which may be transmitted back to the server, for example, answering a multiple choice question put to them by host 160 of the talk show, by "clicking on" user response button 148.

Participants 232 are clients of server 230 who are given a more active role in the animation. For example, the participants may control respective Smart Object characters appearing in the talk show, wherein the Smart Objects preferably comprise the participants' respective avatars. Each participant 232 preferably inputs text, along with a suitable animation from the set of animations belonging to the participant's respective Smart Object. The text and animation are conveyed over network 236 to server 230 and sent out to viewers 234. Participants 232 may further control their avatars' motions, preferably using a pointing device, such as pointing device 50 (Fig. 1), or by other means known in the art.

The local computer hardware and animation rendering software required by participant clients 232 and by viewer clients 234 are substantially the same. Participants 232 generally have additional software enabling them to participate in the animation. A further feature of animation

programs transmitted over network 236 by server 230 is that one or more of viewers 234 may be invited to become participants in such a program.

Fig. 16 is a block diagram illustrating another system architecture 239 for transmission of computer animations over network 236, in accordance with another preferred embodiment of the present invention. System architecture 239 includes a network server 246, which is connected via network 236 to a source client 240 and a recipient client 254. The system is used by source client 240 to prepare an animation, for example, an animated message as described below, and send the animation in the form of electronic mail to recipient 254 for viewing.

In the preferred embodiment illustrated by Fig. 16, the animation is prepared at source 240 and sent to recipient 254 using conventions that are well known in the art. Source 240 and recipient 254 are preferably equipped with computers 40 as shown in Fig. 1, or with similar equipment. Preferably, as shown in Fig. 16, the animation sent to recipient 254 is based on the JAVA language and HTML file format. It will be appreciated that the principles of the present invention, as exemplified by Fig. 16 and described with reference thereto, may similarly be applied to prepare and transmit electronic mail that includes 3D animations, using VRML, for example.

Source client 240 uses a message editor 242 and assets 244, comprising a library of animation elements, as described below, to prepare an animated message for recipient 254. The message is conveyed to server 246 via network 236, addressed to recipient 254, in the form of an encapsulated textual description, rather than as an animation, and thus substantially reduces transmission time and memory needed to contain the message.

Server 246 includes a JAVA message generator program 248. The server maintains in its memory JAVA classes 250, including program routines for producing animations, sound, and text, as are known in the art, and assets 252, generally matching the set of assets 244 of source client 240. Message generator 248 converts the source client's message to HTML. Message generator 248 receives the textual description of the animation from source client 240 and uses it to generate an HTML document. Server 246 transmits a URL reference to the HTML document, as is known in the art, in the form of an electronic mail message over network 236 to recipient 254. When recipient 254 opens the message, the HTML document activates a JAVA applet on server 246. The applet causes the animation prepared by source client 240 to be replayed on the recipient's computer screen, thus conveying the animated message. System 239 allows a user of source computer 240 to compose and send an animated message to recipient 254 in a manner analogous to sending an electronic mail letter, without the need to understand or follow the conventions of the JAVA language and HTML files.

Fig. 17A is a graphic representation of display 52, illustrating an on-screen window 260 used by editor 242 (Fig. 16) to create an animated message, in accordance with a preferred embodiment of the present invention. Window 260 includes an animation area 262, which displays an animated scene that includes an animated character 264, text 266 and other elements

to be included in the message. The window also includes menus 272 and user controls 268, which are used in producing and editing the message.

Fig. 17B is a flow chart illustrating a method for creating and editing the message of Fig. 17A, in accordance with a preferred embodiment of the present invention. A user of editor 242 uses menus 272 and user controls 268 to compose the animated scene. The user selects tiles and colors for the background and the border of the scene. Preferably, at least one animation object is selected, such as character 264, along with a message, such as text 266. The background, border and animation object are preferably chosen from among assets 244 using an associative search engine, as described above with reference to Figs. 13A, 13B and 14. It will be understood, however, that the steps of choosing the background and border are optional, and in the event that the user does not choose a background and/or border, editor 242 provides a default background and border.

The user next inputs characteristics of the motion of the animation object and voice and/or sound effects to be included with the message. Preferably the animation object comprises a Smart Object, and the user needs only to define a path within the scene along which the object is to move. Further preferably, the user may assign to the animation object certain responses to events and user interactions, for example, defining a response of the object that is invoked when a recipient selects the object. In the case of Smart Objects, such responses may be assigned by adding conditional and/or optional animations to the Smart Object, as described above with reference to Fig. 4.

Once the elements of the scene in animation area 262 have been composed, the user may observe and edit the scene and make any desired changes. The user may also add additional scenes, in sequence, so as to create a multi-scene message.

When the animated message is complete, the user may preview the animation using an on-screen control bar 270, shown in Fig. 17A. The message may also be saved to disk as an HTML file. When the message is ready to be sent, the user selects a "mail" button 272, and the textual description of the message is conveyed to server 246, as described above.

Fig. 18A is a graphic representation of display 52 belonging to recipient 254 (Fig. 16), showing the message conveyed from source client 240, in accordance with a preferred embodiment of the present invention. A window 280 on the screen of display 52 is created by a well-known Internet browser. Animated character 264 and text message 266, as created by the source client, appear within an animation area 282 in window 280.

Fig. 18B is a flow chart illustrating a method by which the message is presented on display 52, in accordance with a preferred embodiment of the present invention. Recipient 254 receives the electronic mail message conveyed by server 246. When the message is opened, a "hot link" appears in the message, as is known in the art, providing a link to a URL created by server 246 for displaying the animated message created by source client 240. When the hot link is selected, the Internet browser associated with window 280 is opened (if it is not open already),

and the animated message is displayed in the window, as shown in Fig. 18A. Alternatively, the recipient user may himself determine that the message contains an HTML file and open the browser to display the animation. When other, non-HTML electronic mail messages are received by recipient 254, they are simply read as text.

Although the animated electronic mail message of Fig. 18A comprises a JAVA applet, which drives an animation that is displayed in a dedicated animation area 282, the principles of the present invention may be employed to encapsulate and send animated messages of other types, as well. For example, a 3D Smart Object may be created in a VRML-compatible format to serve as the animated avatar of a user. This avatar may be incorporated into an mail message sent by the user, so that when recipient opens the message using a VRML-compatible browser, the animated 3D avatar will appear on the recipient's computer display. If desired, the avatar can speak (or even sing) the user's message. The 3D Smart Object is preferably rendered as a Transparent 3D overlay on the browser window, as shown, for example, in Fig. 2.

It will be appreciated that Smart Objects are particularly well suited for encapsulation and transmission over computer networks. The modularity of Smart Objects allows them to be easily customized, to present a desired animated image, such as an avatar, and/or to convey a message chosen by a user. Furthermore, the substantial autonomy of Smart Objects allows animation files that include Smart Objects to be relatively compact, since beyond the Smart Objects themselves, such files must generally contain only a script, and need not include details of the animation.

It will moreover be appreciated that the preferred embodiments described above are cited by way of example, and the full scope of the invention is limited only by the claims.

CLAIMS

1. A method for producing an animation sequence on a graphic display driven by a computer, comprising:

defining an object comprising data, including a geometrical description of an animated character and characteristics of social behavior of the character; and

animating an image of the character responsive to the characteristics.

2. A method according to claim 1, wherein defining the object comprises programming a predetermined response of the character to a sensitivity condition occurring externally to the object.

3. A method according to claim 2, wherein programming the predetermined response to the interaction condition comprises programming a rule governing motion of the object responsive to the condition.

4. A method according to claim 3, wherein defining the object comprises defining a geometrical skeleton of the object characterized by a hierarchy of sub-objects connected by joints, and wherein programming the rule governing motion comprises defining motions of the joints.

5. A method according to any of claims 2-4, and comprising assigning the character a human personality type and programming a set of responses appropriate to the personality type, which set of responses includes the predetermined response to the condition.

6. A method according to any of claims 2-5, wherein programming the predetermined response comprises programming a response of the character to another item whose image is shown in the display.

7. A method according to claim 6, wherein programming the response to the other item comprises programming a response of the character to the other item that is triggered in response to proximity of the item to the character in the display.

8. A method according to claim 6 or 7, wherein programming the response of the character to the other item comprises programming a response such that the character exhibits a response to the other item indicative of a human emotion.

9. A method according to any of claims 6-8, wherein programming the response of the character to the other item comprises programming a response of the character to an item whose image is rendered on the display substantially under the control of another program running on the computer substantially unrelated to the object.

10. A method according to any of claims 2-9, wherein programming the predetermined response comprises programming a response of the character to being selected by a user of the computer graphic display.

11. A method according to any of claims 2-10, wherein programming the predetermined response comprises programming an idle behavior undertaken by the character when a predetermined time has elapsed without another sensitivity condition having occurred.

12. A method according to any of claims 2-11, wherein animating the image of the character comprises animating the image responsive to a script, wherein the script includes the sensitivity condition that elicits the predetermined response of the character.

13. A method according to any of claims 2-12, and comprising transferring the object over a network from a source computer to a destination computer, wherein animating the image comprises animating an image of the character on a display associated with the destination computer, in accordance with the predetermined response of the character to a sensitivity condition created at the destination computer.

14. A method according to any of the preceding claims, wherein defining the object comprises defining sufficient characteristics of the behavior of the character such that the image is animated substantially without reference to animation instructions external to the object.

15. A method according to any of the preceding claims, wherein animating the image comprises rendering a three-dimensional image of the character.

16. A method according to claim 15, wherein rendering the three-dimensional image comprises overlaying the image on a window in the display generated by a program running on the computer substantially independently of the object.

17. A method according to claim 16, wherein overlaying the image on the window comprises overlaying an image such that substantially the only portion of the window that is obscured is the portion directly behind the image of the character.

18. A method according to any of the preceding claims, wherein defining the object comprises defining a first object comprising a geometrical description and characteristics of social behavior of a first animated character, and comprising defining a second object comprising a geometrical description of a second animated character and characteristics of social behavior thereof by transferring at least some elements of the first object to the second object.

19. A method for producing an animation sequence on a graphic display driven by a computer, comprising:

defining a group of objects, each such object associated with a respective animated character and comprising data including:

a geometrical skeleton common to all the objects in the grouped, characterized by a hierarchy of sub-objects connected by joints; and

rules governing behavior of the respective character, which rules define motions of the joints and are interchangeable among any of a plurality of the objects; and

animating an image of at least one of the characters responsive to the rules.

20. A method according to claims 19, wherein defining the group of objects comprising the geometrical skeleton comprises defining a three-dimensional skeleton, and wherein animating the image comprises rendering a three-dimensional image.

21. A method according to claim 19 or 20, wherein defining the group of objects comprises defining an accessory null within the geometrical skeleton, at which null an accessory is coupled to at least one of the objects.

22. A method according to any of claims 19-21, wherein defining the group of objects comprises defining a camera null within the geometrical skeleton, for coupling a virtual camera to at least one of the objects so as to render an image of a scene from a point of view associated with the at least one object.

23. A method according to any of claims 19-22, wherein defining the group of objects comprising the rules governing behavior comprises defining rules governing behavior of a predetermined personality type, associated with one or more of the animated characters.

24. A method according to any of claims 19-23, wherein defining the group of objects comprises defining first and second objects, wherein the second object is defined by inheritance from the first object.

25. A method for providing a user interface in a computer graphic display, comprising:
displaying one or more user control icons in a window in the display; and
scaling the one or more icons responsive to a change of scale of the window.

26. A method according to claim 25, wherein displaying the one or more icons comprises displaying two icons in mutual proximity, and wherein scaling the one or more icons comprises changing a distance between the two icons.

27. A method according to claim 25 or 26, wherein displaying the one or more icons comprises displaying an elongate icon of predetermined length, and wherein scaling the one or more icons comprises changing the length of the elongate icon.

28. A method according to any of claims 25-27, wherein displaying the one or more user control icons comprises producing a frame image including the icons and having a transparent area, and overlaying the frame image on an animation generated on the display, such that the animation is framed within the transparent area.

29. A method according to claim 28, wherein producing the frame image having the transparent area comprises producing a frame image having a non-rectangular transparent area.

30. A method for producing an animation sequence on a graphic display driven by a computer, comprising:
producing a frame image having a non-rectangular transparent area;
generating an animation sequence on the display; and

overlaying the frame image on the animation sequence, so that the animation is framed by the transparent area.

31. A method for producing an animation sequence on a graphic display driven by a computer, comprising:

defining a three-dimensional geometrical description of an animation scene;

selecting first and second reference points relative to the scene and defining a geometrical path dependent on the points;

rendering a first image of the scene as seen from a viewpoint at a first position on the geometrical path;

translating the viewpoint to a second position along the geometrical path; and

rendering a second image of the scene as seen from the second position.

32. A method according to claim 31, wherein defining the geometrical path comprises defining a substantially linear path connecting the first and second reference points.

33. A method according to claim 31, wherein defining the geometrical path comprises defining a generally arcuate path connecting the first and second reference points.

34. A method according to claim 31, wherein defining the geometrical path comprises defining a path of revolution generally centered at the second reference point and passing through the first reference point.

35. A method according to any of claims 31-34, wherein selecting the second reference point comprises selecting a point corresponding generally to the position of a character in the scene.

36. A method according to any of claims 31-35, wherein translating the viewpoint comprises translating the viewpoint using an on-screen joystick control.

37. A method according to any of claims 31-36, wherein rendering the first image of the scene as seen from the viewpoint at the first position on the geometrical path comprises rendering the image of the scene as seen from a first viewpoint, and comprising defining a second viewpoint and providing an on-screen control such that when the control is activated, an image of the scene is rendered from the second viewpoint.

38. A method for producing an animated overlay image on a graphic display driven by a computer, comprising:

generating a three-dimensional animated image element; and

overlaying the animated image element on a window in the graphic display driven by a software application substantially unrelated to the generation of the animated image element.

39. A method according to claim 38, wherein overlaying the image element on the window comprises overlaying an image element such that the substantially unrelated software application continues to run substantially as though the image element was not overlaid on the window.

40. A method according to claim 38 or 39, wherein overlaying the image element on the window comprises overlaying an image element such that substantially the only portion of the window obscured is the portion directly behind the image element on the graphic display.

41. A method according to any of claims 38-40, wherein overlaying the animated image element comprises finding a border circumscribing the image element, defining an animation window which is shaped to fit the border, and altering the shape of the animation window substantially whenever the border changes.

42. A method according to any of claims 38-41, wherein generating the animated image element comprises controlling a motion of the element responsive to a characteristic of the display driven by the substantially unrelated software application.

43. A method according to claim 42, wherein controlling the motion responsive to the characteristic of the display comprises controlling a motion responsive to a graphic icon in the display.

44. A method according to claim 42 or 43, wherein generating the animated image element comprises defining an object corresponding to the image element and including a rule governing motion of the object, and wherein controlling the motion of the element comprises controlling a motion responsive to the rule.

45. A method according to any of claims 38-44, and comprising transferring a data module corresponding to the image element over a network from a source computer to a destination computer, wherein generating and overlaying the image element comprise generating and overlaying an image on a display associated with the destination computer.

46. A method for conveying an animation from a source computer to a destination computer, comprising:

defining an object in the source computer, the object comprising a graphic description of an element for inclusion in the animation and a characteristic of motion of the element;

transmitting the object from the source computer to the destination computer via a network; and

displaying an animated image on a display driven by the destination computer, wherein the image includes a graphic rendition of the element, moving in accordance with the characteristic.

47. A method according to claim 46, wherein transmitting the object via the network comprises encapsulating the object and transmitting the encapsulated object in the form of an electronic mail message.

48. A method according to claim 47, wherein encapsulating the object comprises generating a textual description of the object, and wherein transmitting the object comprises transmitting the

textual description to a server computer, which generates an animation file based on the textual description and delivers the animation file to the destination computer.

49. A method according to claim 47 or 48, wherein encapsulating the object comprises generating a HTML document that invokes a JAVA applet.

50. A method according to claim 46, and comprising transmitting a script over the network, which script defines an animation sequence including the element, and wherein transmitting the object and transmitting the script comprise transmitting the object and the script to a plurality of destination computers.

51. A method according to claim 50, wherein displaying the animated image on the display driven by the destination computer comprises providing user controls on the destination computers for controlling rendition of the animated image.

52. A method according to claim 51, wherein providing user controls comprises providing on-screen control icons on the display.

53. A method according to claim 52, and comprising scaling the icons responsive to a change in scale of a window on the display containing the animated image.

54. A method according to any of claims 51-53, wherein controlling rendition of the animated image comprises changing a viewpoint with respect to which the image is rendered.

55. A method for finding a desired image among a library of images stored by a computer, comprising:

displaying a first plurality of images from the library on a display associated with the computer;

selecting a first image from among the first plurality; and

searching through the library to find a second plurality of images resembling the first image in one or more characteristics thereof.

56. A method according to claim 55, and comprising displaying the first image and the second plurality of images, such that the images of the second plurality most closely resembling the first image in the one or more characteristics are displayed in closest proximity to the first image.

57. A method according to claim 56, wherein displaying the first image comprises displaying the first image in a central position on the display, and wherein displaying the second plurality of images comprises displaying the images of the second plurality in a generally concentric pattern around the first image.

58. A method according to any of claims 55-57, wherein displaying the images comprises displaying a cursor on the display and altering one of the images when the cursor points to the one of the images.

59. A method according to claim 58, wherein altering one of the images comprises animating the image.

60. A method according to claim 58 or 59, wherein altering one of the images comprises enlarging the image.

61. A method according to any of claims 58-60, wherein altering one of the images comprises displaying a three-dimensional rendition of the image.

62. A method according to any of claims 55-61, and comprising repeating the steps of selecting and searching with respect to a second image from among the second plurality.

63. A method according to any of claims 55-62, and comprising assigning keywords to the images in the library, wherein searching through the library to find the second plurality of images resembling the first image comprises searching to find images having at least one common keyword with the first image.

64. A method according to claim 63, and comprising choosing a starting keyword, wherein displaying the first plurality of images comprises displaying images matching the starting keyword.

65. A system for producing an animation sequence, comprising:

an animation generator , which generates an animated image of a character, responsive to an object comprising a geometrical description of the character and characteristics of social behavior of the character; and

a display, which is driven by the animation generator to display the animated image.

66. A system according to claim 65, wherein the characteristics of social behavior of the character comprise a predetermined response of the character to a sensitivity condition occurring externally to the object.

67. A system according to claim 66, wherein the predetermined response comprises a rule governing motion of the object.

68. A system according to claim 67, wherein the geometrical description comprises a geometrical skeleton of the object characterized by a hierarchy of sub-objects connected by joints, and wherein the rule governing motion defines motions of the joints.

69. A system according to any of claims 66-68, wherein the character has a human personality type associated therewith, and wherein the predetermined response belongs to a set of responses appropriate to the personality type.

70. A system according to any of claims 66-69, wherein the predetermined response comprises a response of the character to an item whose image is shown on the display.

71. A system according to claim 70, wherein the predetermined response is triggered responsive to proximity of the item to the character on the display.

72. A system according to claim 70 or 71, wherein the predetermined response is such that the character exhibits an attraction to the item.

73. A system according to any of claims 70-72, wherein the animation generator renders the image of the item on the display responsive to another program running on the animation generator substantially independently of the object.

74. A system according to any of claims 66-73, and comprising a user input device for selecting the character, wherein the predetermined response comprises a response of the character to being selected.

75. A system according to any of claims 66-74, wherein the predetermined response comprises an idle behavior undertaken by the character when a predetermined time has elapsed without another sensitivity condition having occurred.

76. A system according to any of claims 66-75, wherein the animation generator is programmed to produce the animation sequence according to a script, which includes the sensitivity condition that elicits the predetermined response of the character.

77. A system according to any of claims 66-76, wherein the animation generator is connected to a network and receives the object from a source animation generator through the network.

78. A system according to any of claims 65-77, wherein the object comprises sufficient characteristics of the behavior of the character so that the animation generator generates the animated image substantially without reference to program instructions external to the object.

79. A system according to any of claims 65-78, wherein the animation generator renders a three-dimensional image of the character.

80. A system according to claim 79, wherein the image is overlaid on a window in the display generated by the animation generator in response to another program substantially independent of the object.

81. A system according to claim 80, wherein the image is overlaid so that substantially the only portion of the window that is obscured is the portion directly behind the image of the character.

82. A system according to any of claims 65-81, wherein the object comprises a first object, and wherein the animation generator generates an animated image of another character responsive to a second object, comprising a graphic description of the other character and characteristics of social behavior thereof, defined by transferring at least some elements of the first object to the second object.

83. A system for producing an animation sequence, comprising:

an animation generator, which generates an animated image of at least one character from a group of characters, responsive to a respective one of a group of objects,

the objects comprising data that includes a geometrical skeleton common to all the objects in the group, characterized by a hierarchy of sub-objects connected by joints, and rules governing behavior of the respective characters, which rules define motions of the joints and are interchangeable among any of the objects; and

a display, which is driven by the animation generator to display the animated image.

84. A system according to claim 83, wherein the skeleton comprises a three-dimensional skeleton and the animation generator renders a three-dimensional image to the display.

85. A system according to claim 83 or 84, wherein the skeleton comprises an accessory null, at which null an accessory is coupled to at least one of the objects.

86. A method according to any of claims 83-85, wherein the skeleton comprises a camera null, for coupling a virtual camera to at least one of the objects so as to render an image of a scene from a point of view associated with the at least one object.

87. A system according to any of claims 83-86, wherein the rules governing behavior comprise rules governing behavior of a predetermined personality type associated with one or more of the animated characters.

88. A system according to any of claims 83-87, wherein the group of objects comprises a first object and a second object, which is defined by inheritance from the first object.

89. A system for producing graphic images, comprising:

an animation generator, which generates a scalable graphic window with one or more user interface icons contained therein, such that the icons are scaled responsive to a scale of the window;

a user input device for scaling the window; and

a display, which is driven by the animation generator to display the window with the icons.

90. A system according to claim 89, wherein the one or more icons comprise two icons in mutual proximity, and wherein a distance between the two icons is changed when the icons are scaled.

91. A system according to claim 89 or 90, wherein the one or more icons comprise an elongated icon of a predetermined length, which length changes when the icon is scaled.

92. A system according to any of claims 89-91, wherein the animation generator generates a frame image including the icons and having a transparent area, and wherein the frame image is

overlaid on an animation generated by the animation generator on the display, such that the animation is framed within the transparent area.

93. A system according to claim 92, wherein the transparent area is non-rectangular.

94. A system for producing an animation sequence, comprising:

an animation generator, which generates an animation sequence and which produced a frame image having a non-rectangular transparent area, such that the frame image is overlaid on the animation sequence, whereby the animation is framed by the transparent area; and

a display, which is driven by the animation generator to display the animation sequence framed by the transparent area.

95. A system for producing an animation sequence, comprising:

an animation generator, which renders animated images of a three-dimensional scene, each image rendered as seen from a respective viewpoint, including a first image of the scene as seen from a first viewpoint along a predetermined geometrical path and a second image of the scene as seen from a second viewpoint along the path;

a user input device, for selecting first and second reference points, which determine the geometrical path; and

a display, which is driven by the animation generator to display the animated images.

96. A system according to claim 95, wherein the geometrical path comprises a substantially linear path connecting the first and second reference points.

97. A system according to claim 95, wherein the geometrical path comprises a generally arcuate path connecting the first and second reference points.

98. A system according to claim 95, wherein the geometrical path comprises a path of revolution generally centered at the second reference point and passing through the first reference point.

99. A system according to any of claims 95-98, wherein the second reference point corresponds generally to the position of a character in the scene.

100. A system according to any of claims 95-99, wherein the animation generator generates an on-screen joystick control that is activated using the input device for translating the viewpoint along the geometrical path.

101. A system according to any of claims 95-100, wherein the animation generator generates an on-screen control that is activated using the input device to switch the image rendered by the animation generator between at least two different viewpoints.

102. A system for producing an animated image, comprising:

an animation generator, which generates a three-dimensional animated image element and produces a display window driven by a software application substantially unrelated to the

generation of the animated image element, and which overlays the animated image element on the window; and

a display, which is driven by the animation generator to display the window with the animated image overlaid thereon.

103. A system according to claim 102, wherein the software application continues to run substantially as though the image element was not overlaid on the window.

104. A system according to claim 102 or 103, wherein substantially the only portion of the window obscured by the overlay is the portion directly behind the image element.

105. A system according to any of claims 102-104, wherein the animation generator finds a border circumscribing the image element, defines an animation window shaped to fit the borders, and alters the shape of the animation window whenever the borders change.

106. A system according to any of claims 102-105, wherein the animation generator controls a motion of the image element responsive to a characteristic of the display window driven by the substantially independent software application.

107. A system according to claim 106, wherein the animation generator controls the motion of the image element response to a graphic icon in the display.

108. A system according to claim 106 or 107, wherein the animation generator generates the animated image element responsive to an object corresponding to the image element, wherein the object includes a rule governing motion of the object, and the animation generator controls the motion of the image element responsive to the rule.

109. A system according to any of claims 102-108, wherein the animation generator is connected to a network and receives therefrom a file corresponding to the image element.

110. A system for conveying an animation over a network, comprising a source animation generator, coupled to the network, which defines an animation file comprising an object, which includes a graphic description of an image element for inclusion in the animation and a characteristic of motion of the element and transmits the object over the network in the form of an electronic mail message.

111. A system for conveying an animation over a network, comprising a network animation server, which receives a textual description of an animation object, generates an animation file based thereon, and delivers the animation file to a destination computer.

112. A system according to claim 110 or 111, wherein the animation file comprises a set of parameters to be executed by a JAVA applet.

113. A system for conveying an animation over a network, comprising a network animation server, which transmits over the network a script defining the animation to a plurality of

destination computers, along with an object that comprises a geometrical description of a character for inclusion in the animation and characteristics of social behavior of the character.

114. A system according to any of claims 110-113, and comprising a network, over which the animations are conveyed.

115. A system for viewing an animation, comprising:

a destination animation generator , coupled to a network, which receives via the network a script defining the animation along with an object comprising a geometrical description of an image element for inclusion in the animation and a characteristic of motion of the element, and renders animated images responsive thereto;

user controls, coupled to the destination animation generator , for controlling rendition of the animated images; and

a display, which is driven by the destination animation generator to display the animated images.

116. A system according to claim 115, wherein the user controls comprise on-screen control icons generated by the animation generator on the display.

117. A system according to claim 116, wherein the animated images are contained in a window on the display, and wherein the icons are scaled responsive to a change in scale of the window.

118. A system according to any of claims 115-117, wherein the user controls are used to change a viewpoint with respect to which the images are rendered.

119. A system for searching a library of images, comprising:

image storage apparatus, which stores the library of images and selects a first plurality of the images to be displayed;

a display, which is driven by the image storage apparatus to display the images selected by the apparatus; and

a user input device, coupled to the image storage apparatus, for pointing to and selecting a first image from among the first plurality of the images displayed,

wherein the image storage apparatus searches through the library and selects images resembling the first image in one or more characteristics thereof, for inclusion in a second plurality of images.

120. A system according to claim 119, wherein the images included in the second plurality that most closely resemble the first image are displayed in closest proximity thereto on the display.

121. A system according to claim 120, wherein the first image is displayed in a central position on the display, and wherein the images included in the second plurality are displayed in a generally concentric pattern around the first image.

122. A system according to any of claims 119-121, wherein when the user input device is used to point to one of the images, the image is altered.

123. A system according to claim 122, wherein when the user input device is used to point to one of the images, the image is animated.

124. A system according to claim 122 or 123, wherein when the user input device is used to point to one of the images, the image is enlarged.

125. A system according to any of claims 122-124, wherein when the user input device is used to point to one of the images, the image is rendered in three dimensions.

126. A system according to any of claims 119-125, wherein the user input device is used to select a second image from among the second plurality of the images displayed, and the image storage apparatus searches through the library and selects images resembling the second image in one or more characteristics thereof.

127. A system according to any of claims 119-126, wherein the image storage apparatus stores keywords assigned to each of the images in the library, and wherein the image storage apparatus searches through the library and selects images having at least one keyword in common with the image that is selected.

128. A system according to claim 127, wherein the image storage apparatus selects the first plurality of images by finding images matching a starting keyword entered by a user.

FIG. 1

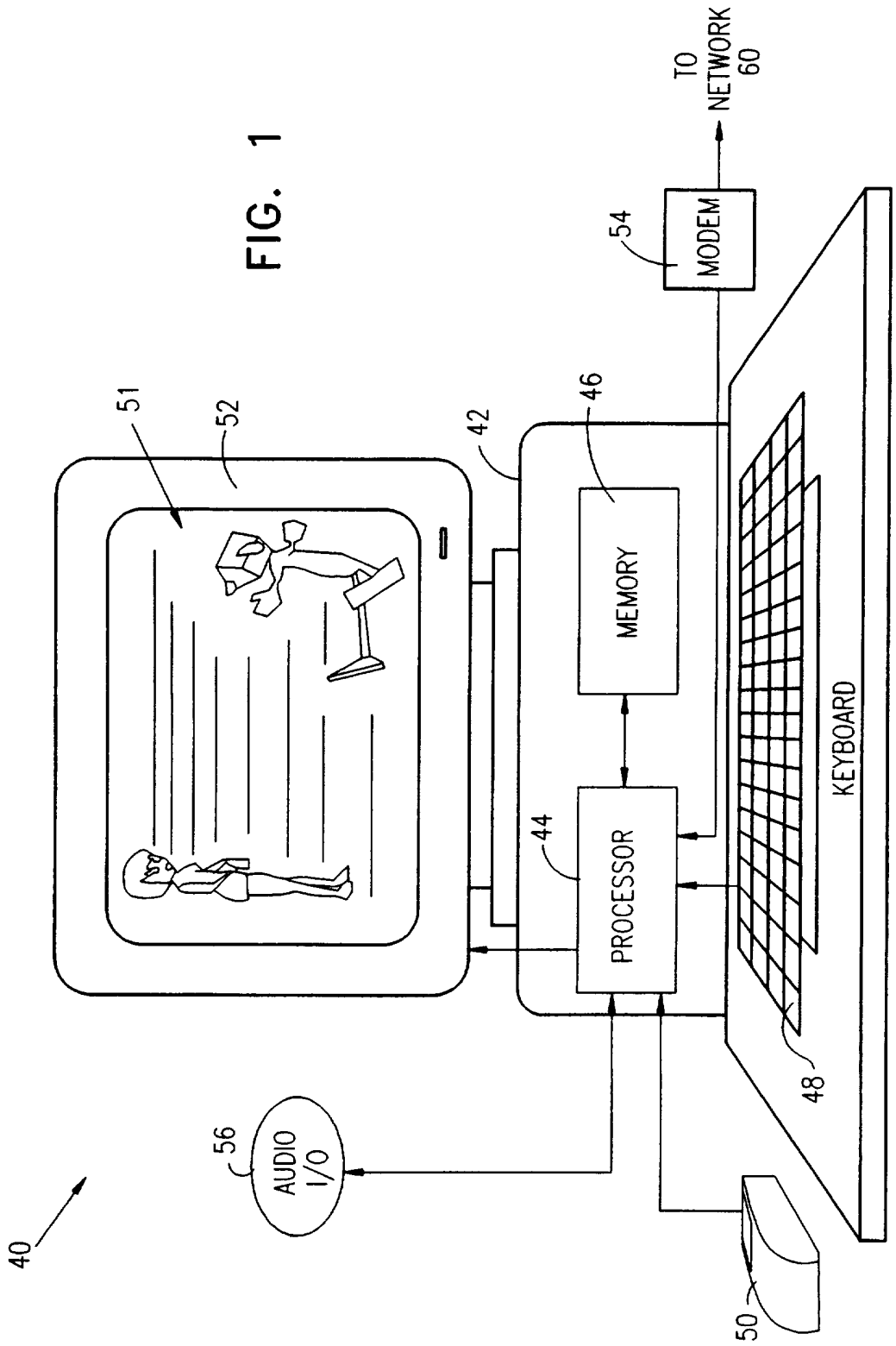

TO NETWORK 60
54
MODEM
51
52
42
46
MEMORY
44
PROCESSOR
KEYBOARD
48
56
AUDIO I/O
40
50

FIG. 2

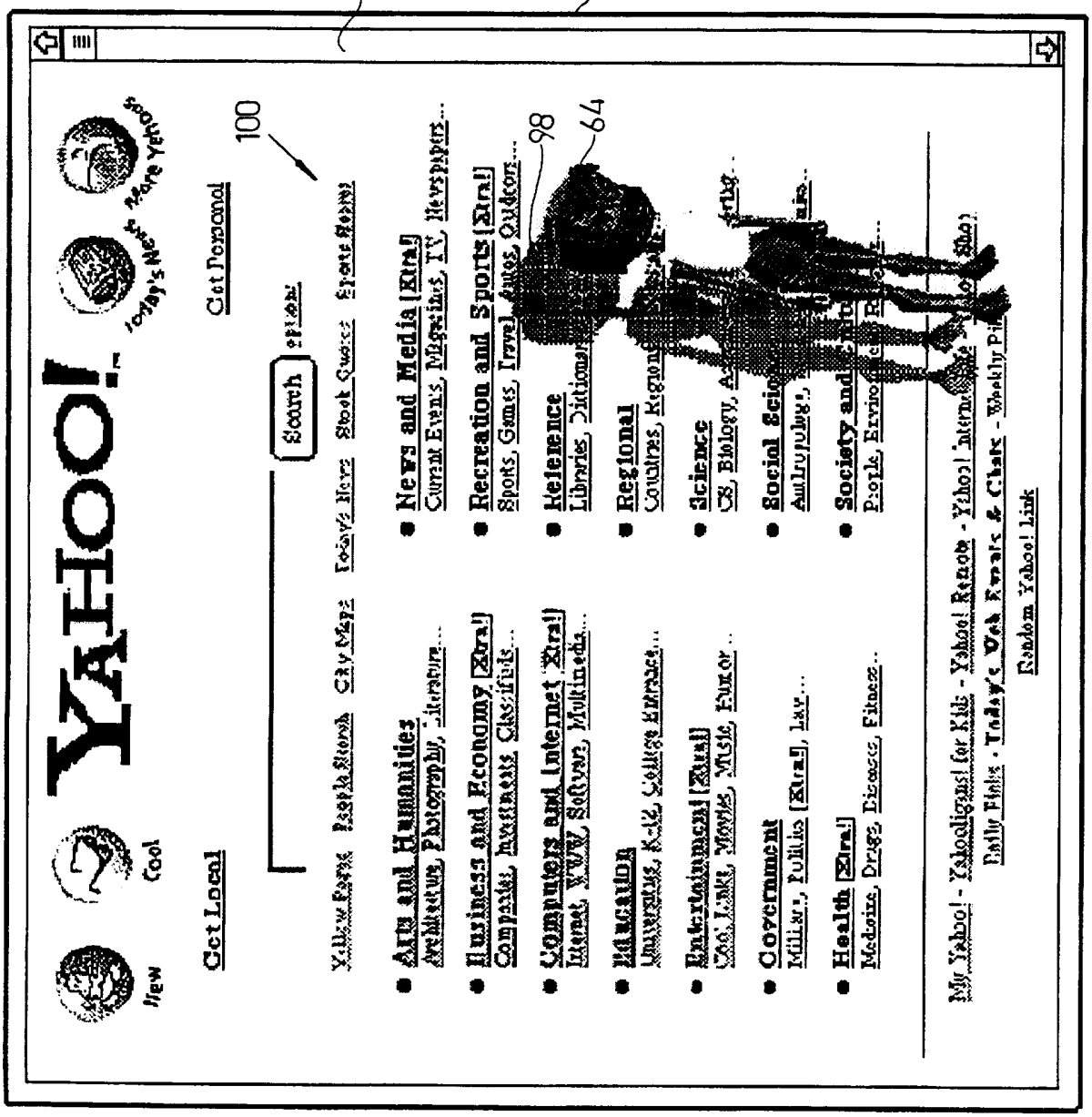

66
52
100
98
64
YAHOO!
New
Cool
Today's News
More Yahoos
Get Local
Get Personal
Search
Yellow Pages
People Search
City Maps
Today's News
Stock Quotes
Sports Scores
Arts and Humanities
Architecture, Photography, Literature...
Business and Economy [Xtra!]
Companies, Investments, Classifieds...
Computers and Internet [Xtra!]
Internet, WWW, Software, Multimedia...
Education
Universities, K-12, College Entrance...
Entertainment [Xtra!]
Cool Links, Movies, Music, Humor...
Government
Military, Politics [Xtra!], Law...
Health [Xtra!]
Medicine, Drugs, Diseases, Fitness...
News and Media [Xtra!]
Current Events, Magazines, TV, Newspapers...
Recreation and Sports [Xtra!]
Sports, Games, Travel, Autos, Outdoors...
Reference
Libraries, Dictionaries...
Regional
Countries, Regions...
Science
CS, Biology, A...
Social Scie...
Anthropology, ...
Society and ...
People, Environment, Religion...
My Yahoo! - Yahooligans! for Kids - Yahoo! Remote - Yahoo! Internet Life - Yahoo! Shop
Daily Picks - Today's Web Events & Chats - Weekly Pi...
Random Yahoo! Link

FIG. 3A

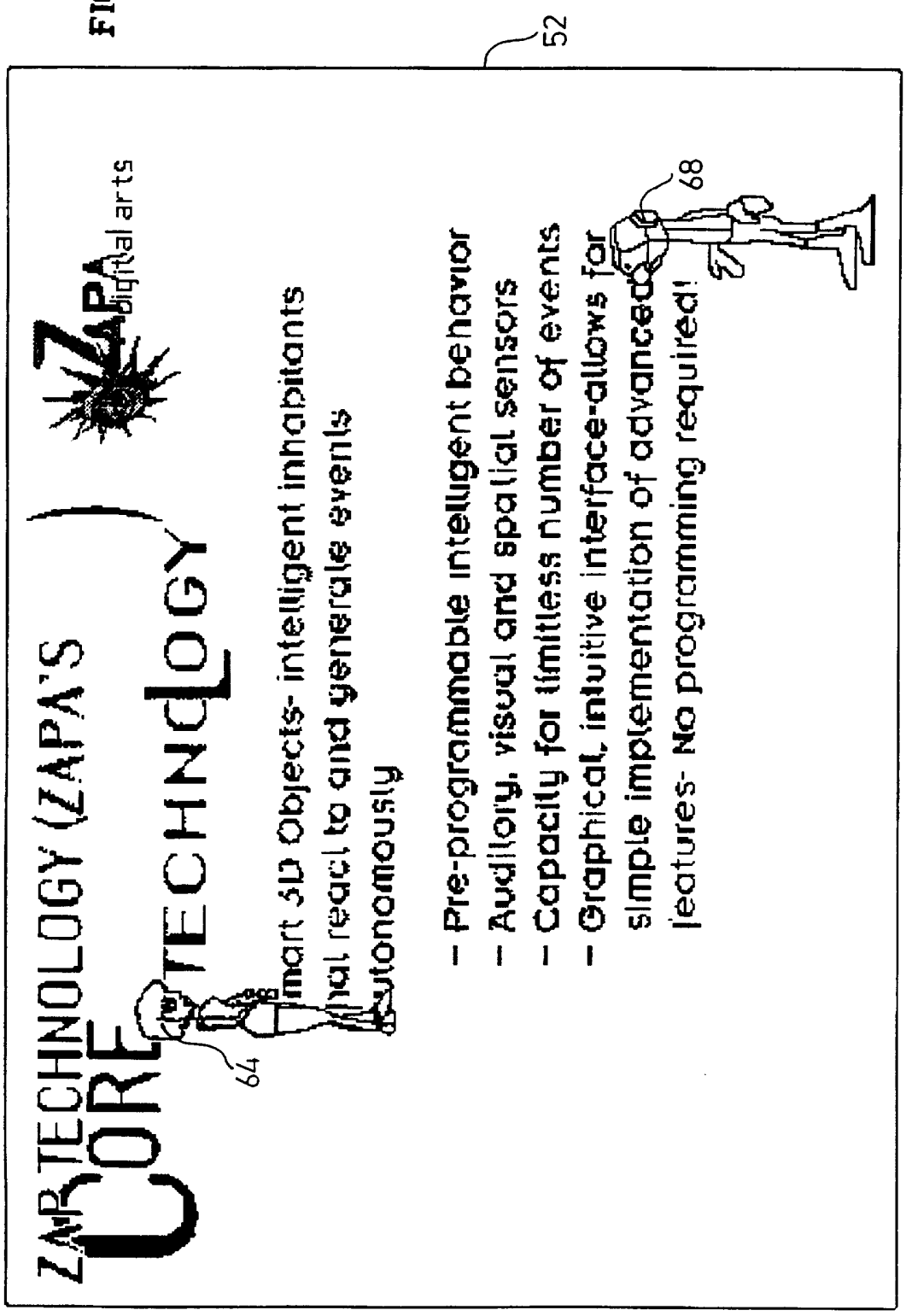

52
ZAPA digital arts
ZAPA'S TECHNOLOGY
CORE TECHNOLOGY
64
mart 3D Objects- intelligent inhabitants
hat react to and generate events
utonomously
- Pre-programmable intelligent behavior
- Auditory, visual and spatial sensors
- Capacity for limitless number of events
- Graphical, intuitive interface-allows for
simple implementation of advanced
features- No programming required!
68

FIG. 3B

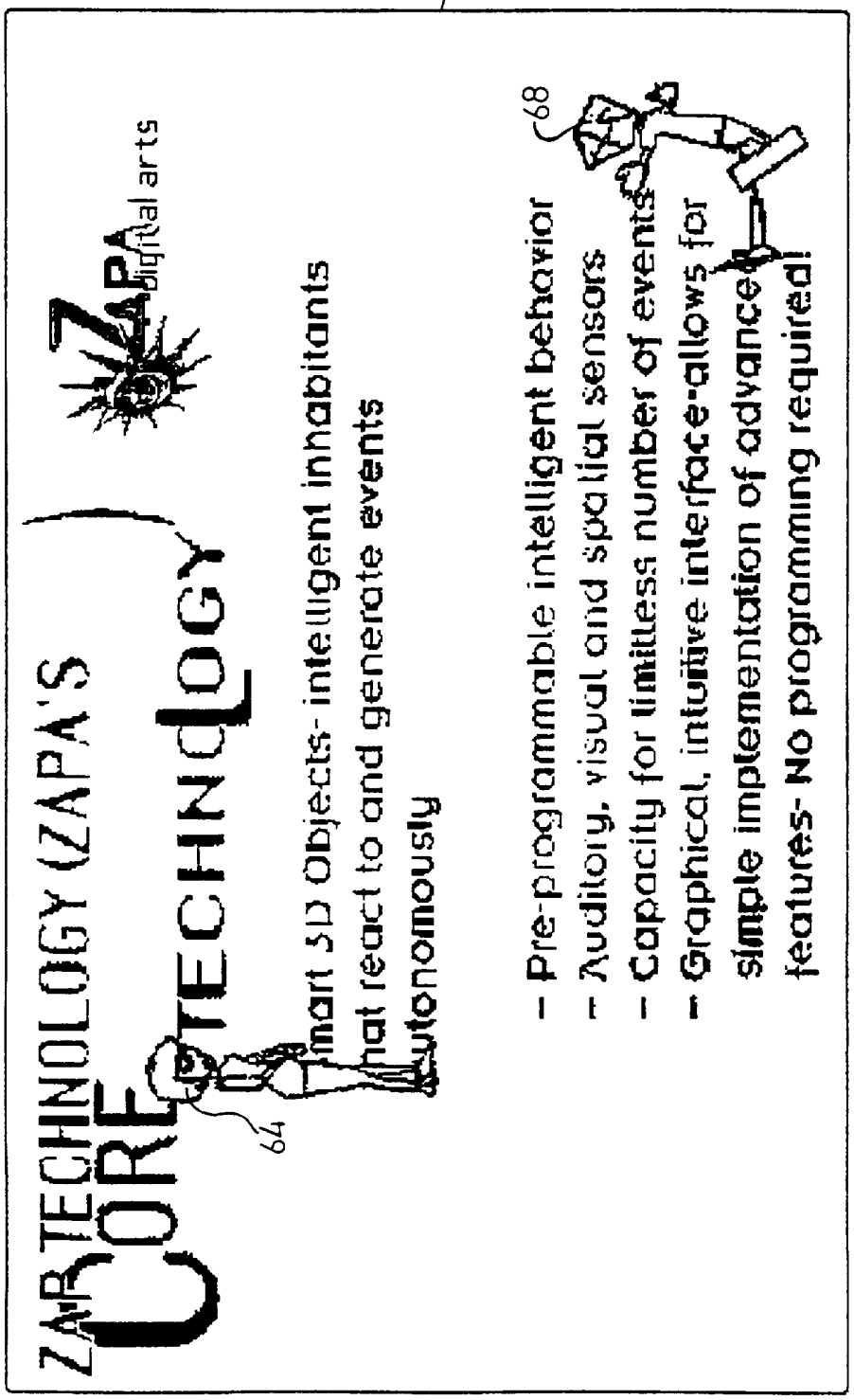

52
ZAP TECHNOLOGY (ZAPA'S
CORE TECHNOLOGY
ZAPA digital arts
64
Smart 3D Objects- intelligent inhabitants
that react to and generate events
autonomously
68
- Pre-programmable intelligent behavior
- Auditory, visual and spatial sensors
- Capacity for limitless number of events
- Graphical, intuitive interface-allows for
simple implementation of advanced
features- No programming required!

FIG. 4

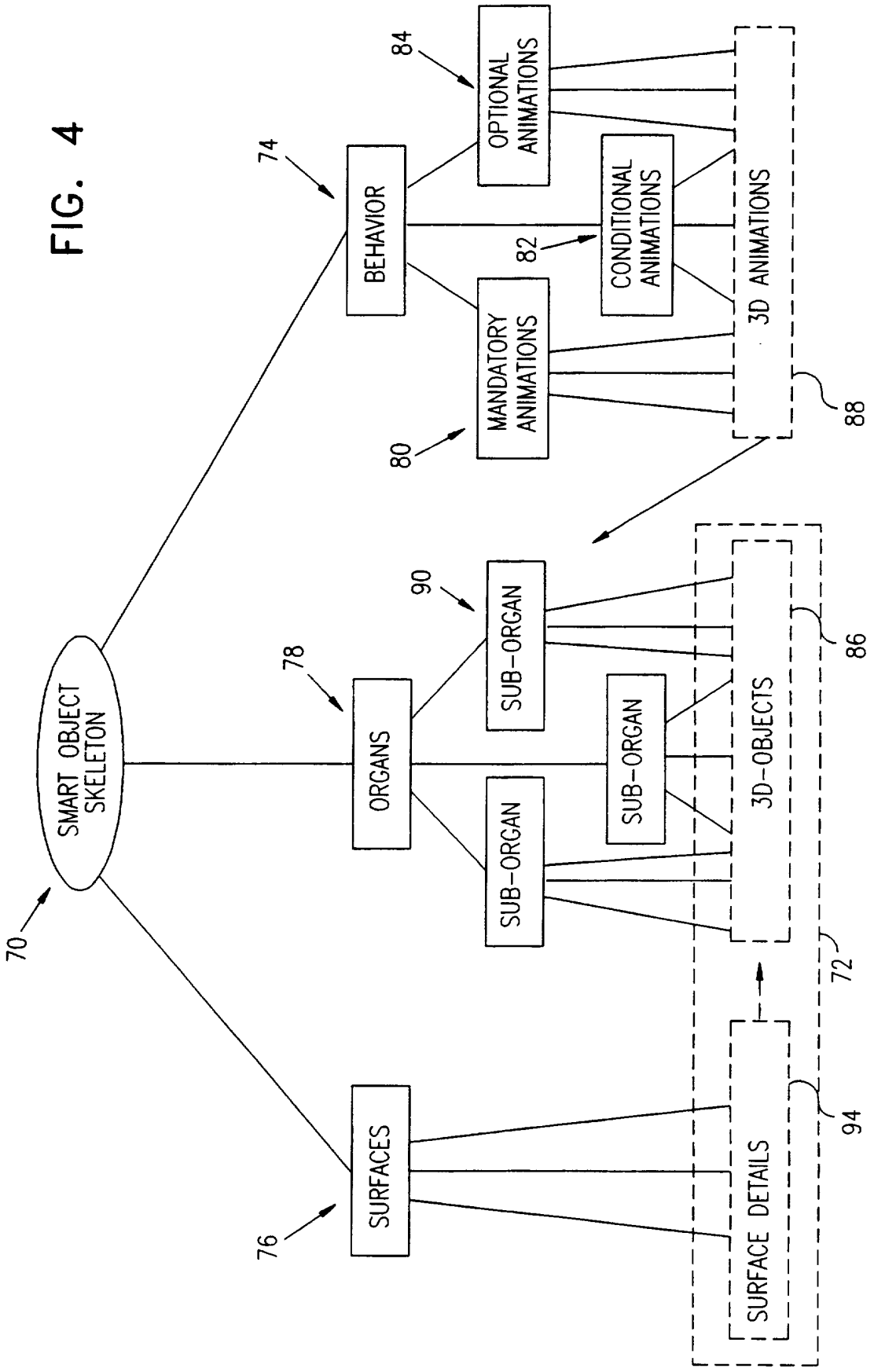

SMART OBJECT SKELETON
70
BEHAVIOR
74
ORGANS
78
SURFACES
76
OPTIONAL ANIMATIONS
84
CONDITIONAL ANIMATIONS
82
MANDATORY ANIMATIONS
80
3D ANIMATIONS
88
SUB-ORGAN
90
SUB-ORGAN
SUB-ORGAN
3D-OBJECTS
86
72
SURFACE DETAILS
94

FIG. 5

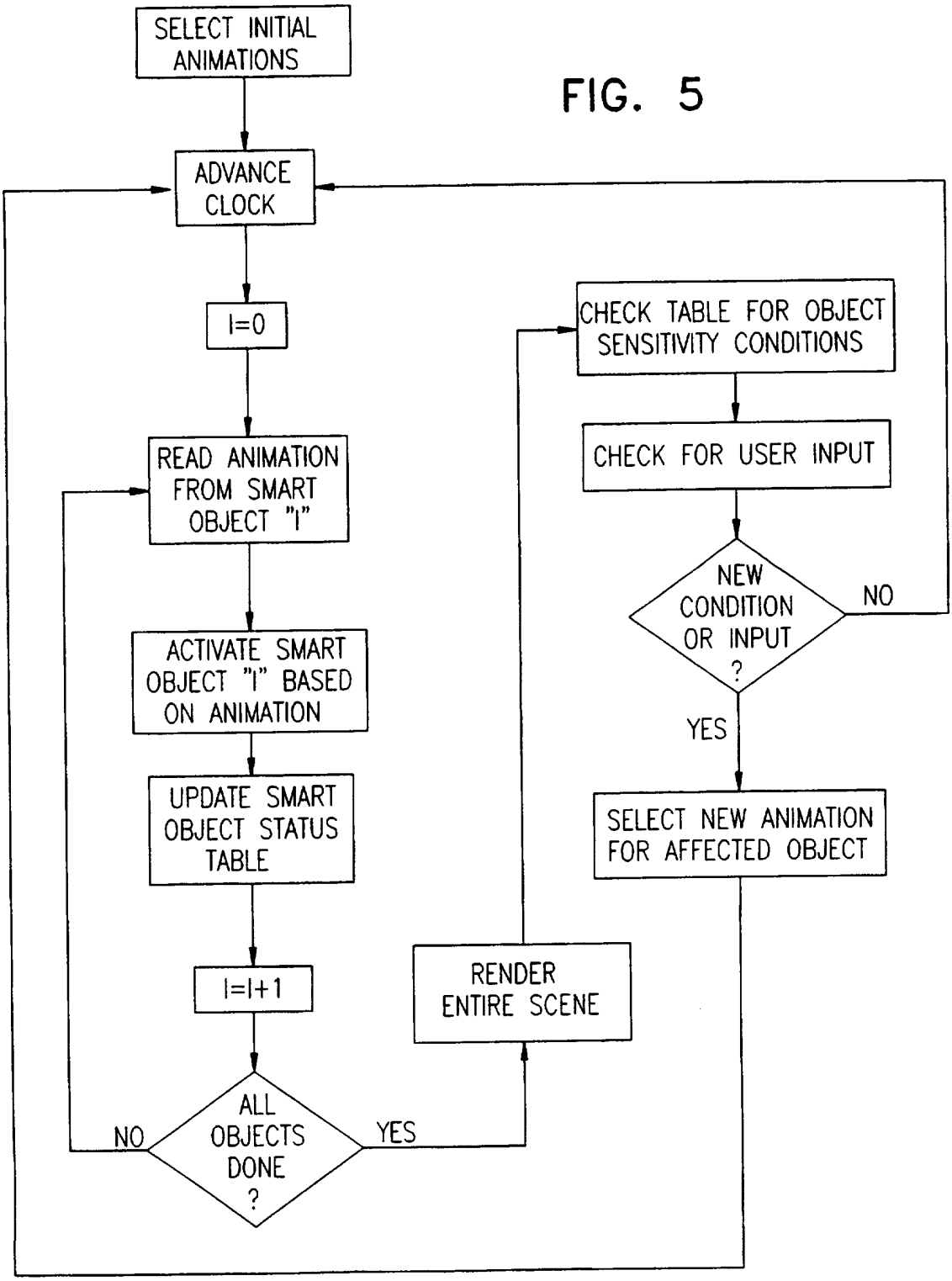

SELECT INITIAL ANIMATIONS
ADVANCE CLOCK
I=0
READ ANIMATION FROM SMART OBJECT "I"
ACTIVATE SMART OBJECT "I" BASED ON ANIMATION
UPDATE SMART OBJECT STATUS TABLE
I=I+1
ALL OBJECTS DONE ?
NO
YES
RENDER ENTIRE SCENE
CHECK TABLE FOR OBJECT SENSITIVITY CONDITIONS
CHECK FOR USER INPUT
NEW CONDITION OR INPUT ?
NO
YES
SELECT NEW ANIMATION FOR AFFECTED OBJECT

FIG. 6A

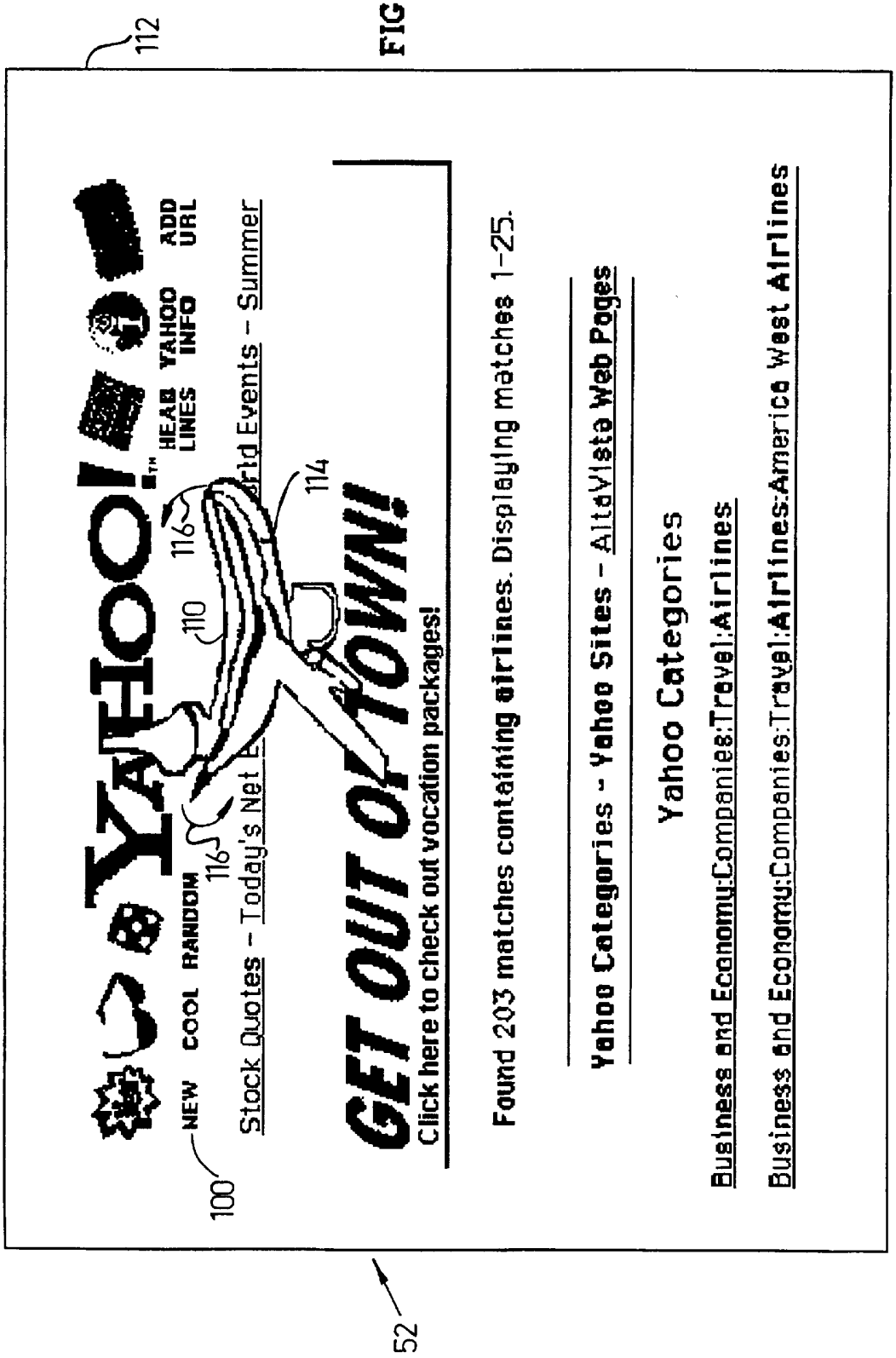
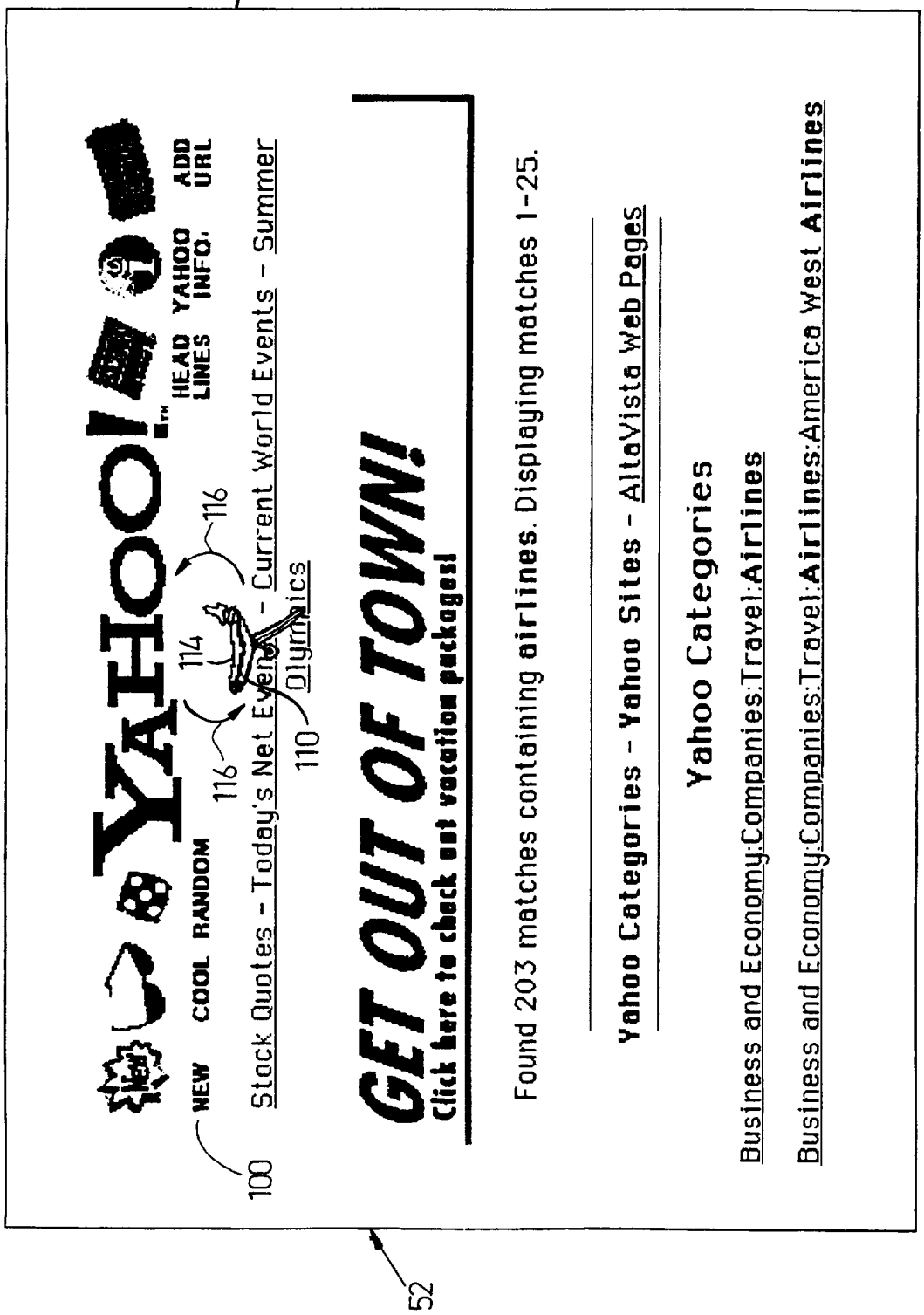

112
NEW COOL RANDOM
HEAD LINES
YAHOO INFO
ADD URL
100
110
116
116
114
Stock Quotes - Today's Net E
rld Events - Summer
GET OUT OF TOWN!
Click here to check out vocation packages!
Found 203 matches containing airlines. Displaying matches 1-25.
Yahoo Categories - Yahoo Sites - AltaVista Web Pages
Yahoo Categories
Business and Economy:Companies:Travel:Airlines
Business and Economy:Companies:Travel:Airlines:America West Airlines
52

FIG. 6B


112
NEW COOL RANDOM
YAHOO!
HEAD LINES
YAHOO INFO
ADD URL
100
Stock Quotes - Today's Net Events - Current World Events - Summer Olympics
110
114
116
116
GET OUT OF TOWN!
Click here to check out vacation packages!
Found 203 matches containing airlines. Displaying matches 1-25.
Yahoo Categories - Yahoo Sites - AltaVista Web Pages
Yahoo Categories
Business and Economy:Companies:Travel:Airlines
Business and Economy:Companies:Travel:Airlines:America West Airlines
52

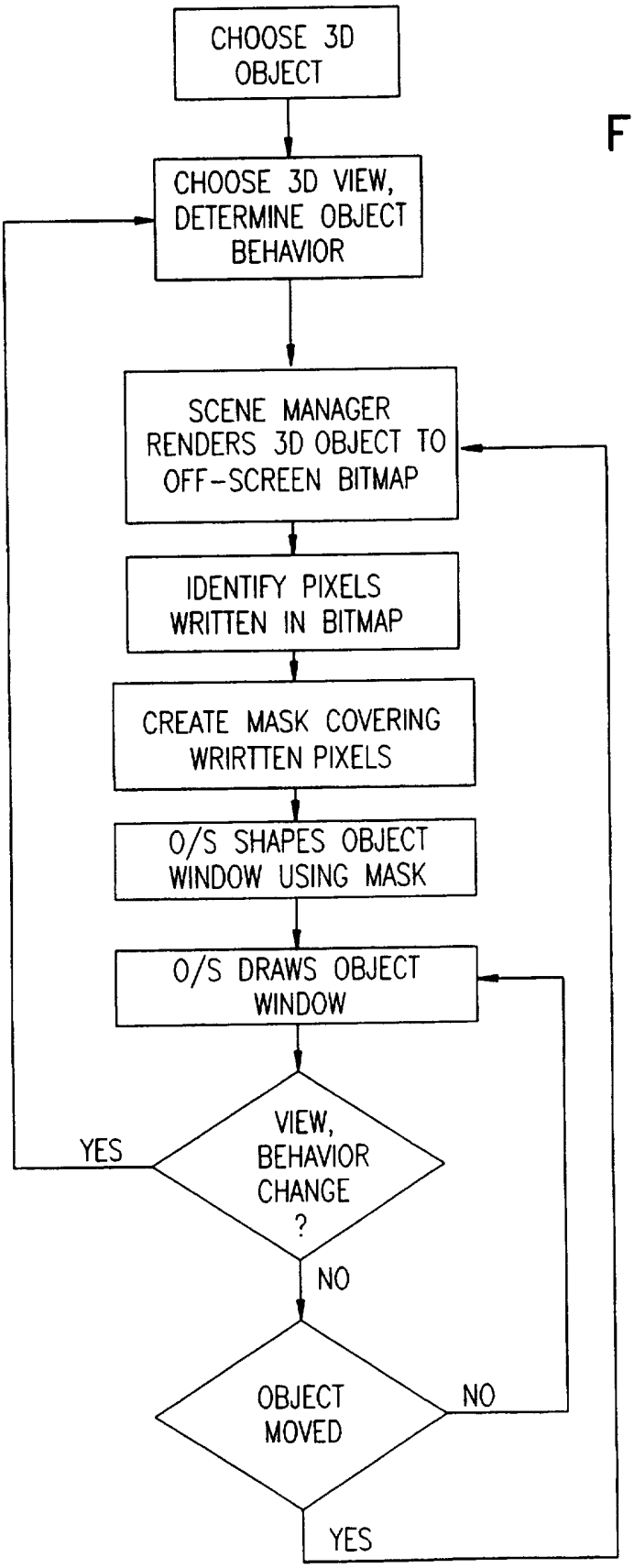

CHOOSE 3D OBJECT
CHOOSE 3D VIEW, DETERMINE OBJECT BEHAVIOR
SCENE MANAGER RENDERS 3D OBJECT TO OFF-SCREEN BITMAP
IDENTIFY PIXELS WRITTEN IN BITMAP
CREATE MASK COVERING WRIRTTEN PIXELS
O/S SHAPES OBJECT WINDOW USING MASK
O/S DRAWS OBJECT WINDOW
VIEW, BEHAVIOR CHANGE ?
YES
NO
OBJECT MOVED
NO
YES

FIG. 7

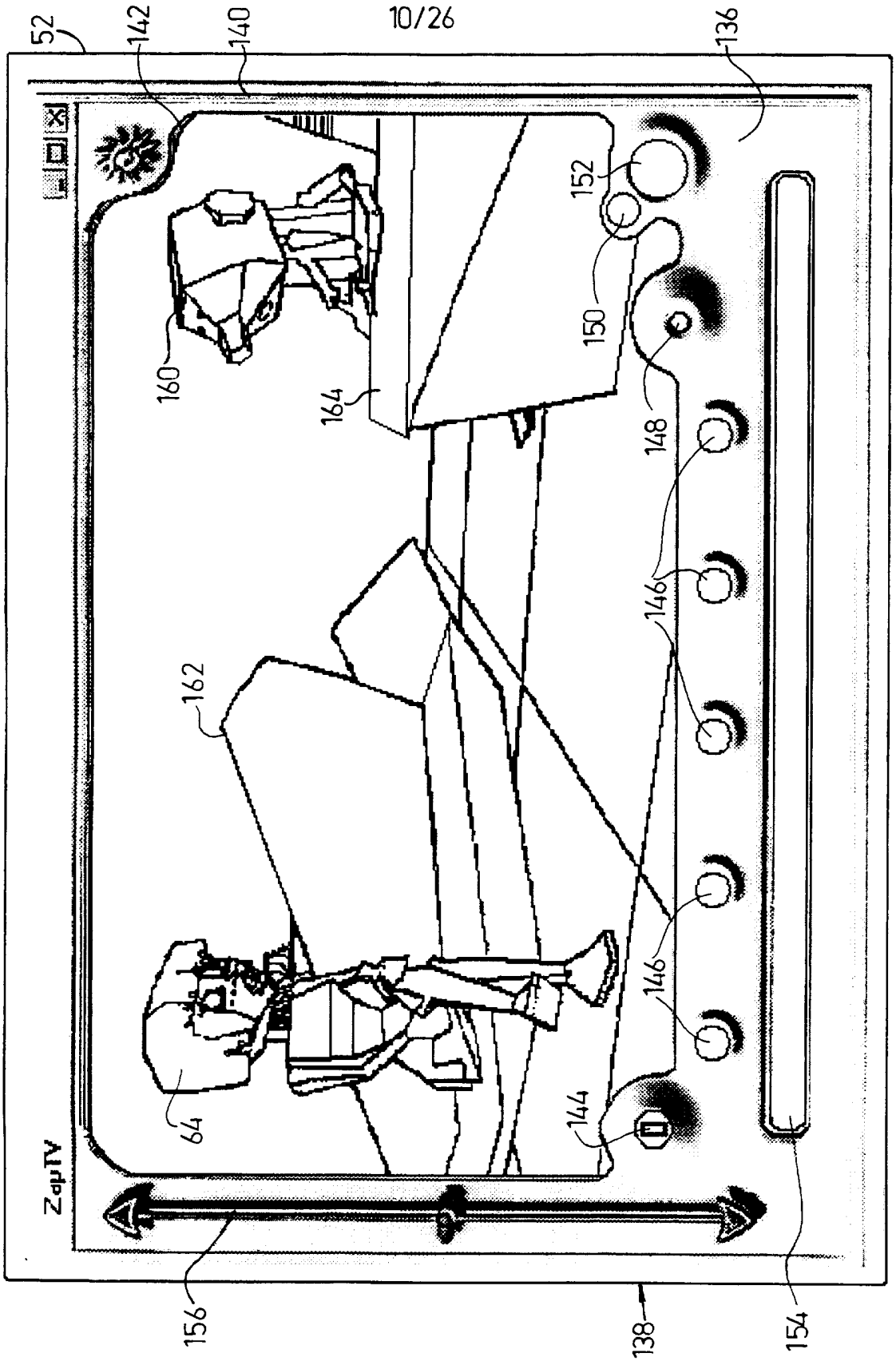

52
142
140
136
152
150
160
164
148
146
162
146
64
144
146
156
138
154

FIG. 8

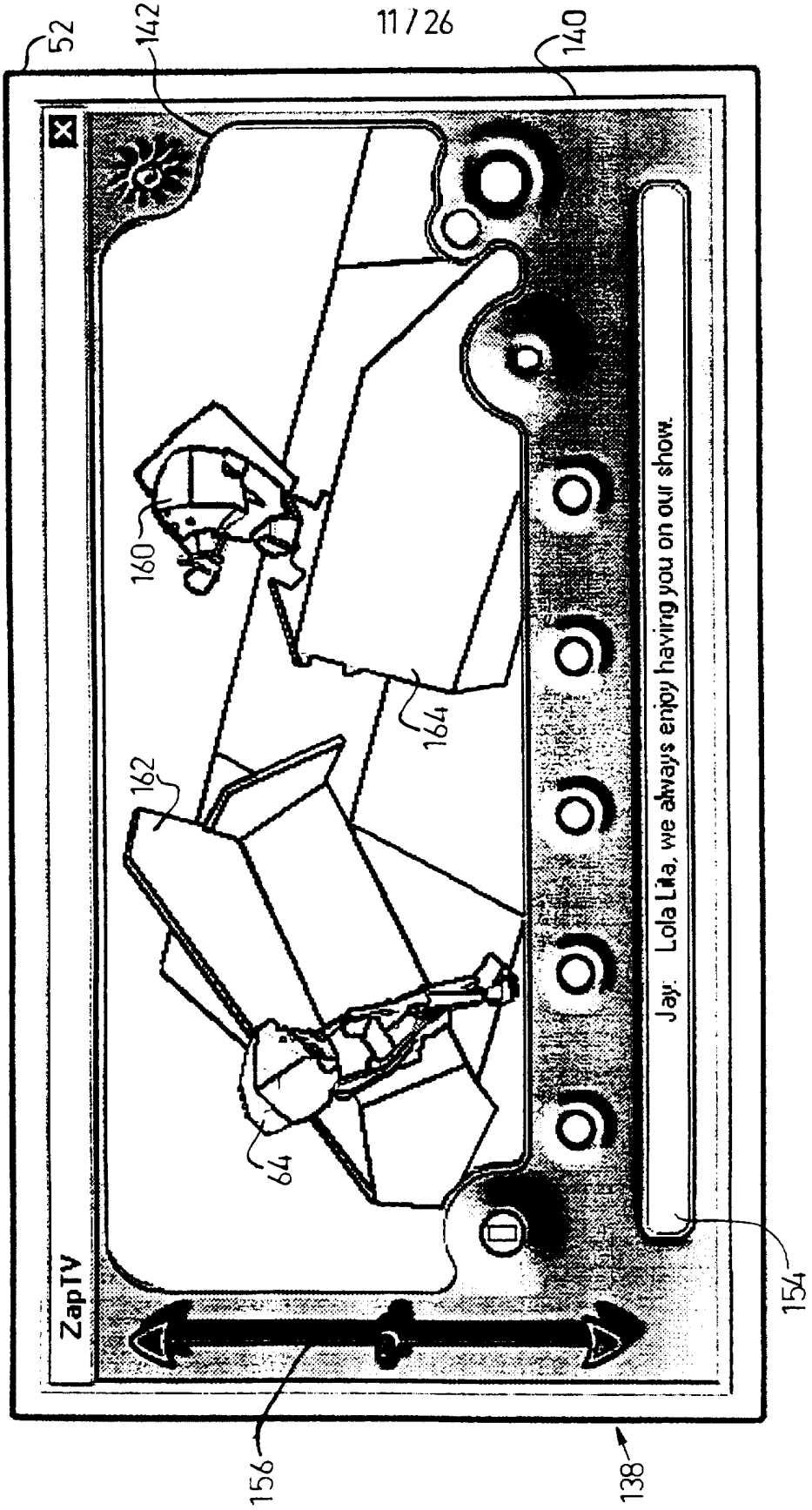
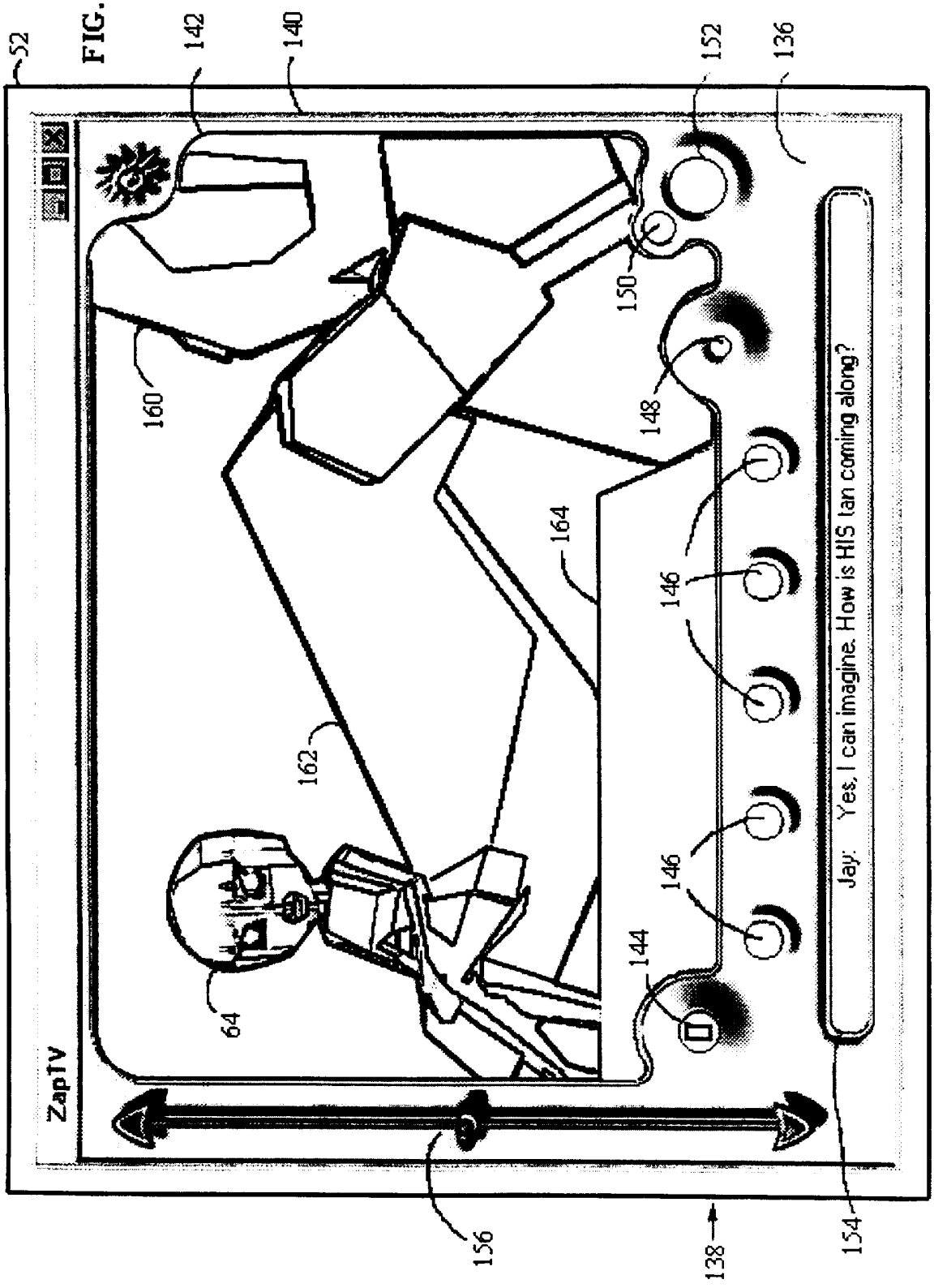

52
142
140
160
162
164
64
ZapTV
Jay: Lola Lita, we always enjoy having you on our show.
154
156
138

FIG. 9A

FIG. 9B


52
142
140
152
136
150
160
148
164
146
162
146
64
144
ZapTV
Jay: Yes, I can imagine. How is HIS lan coming along?
156
138
154

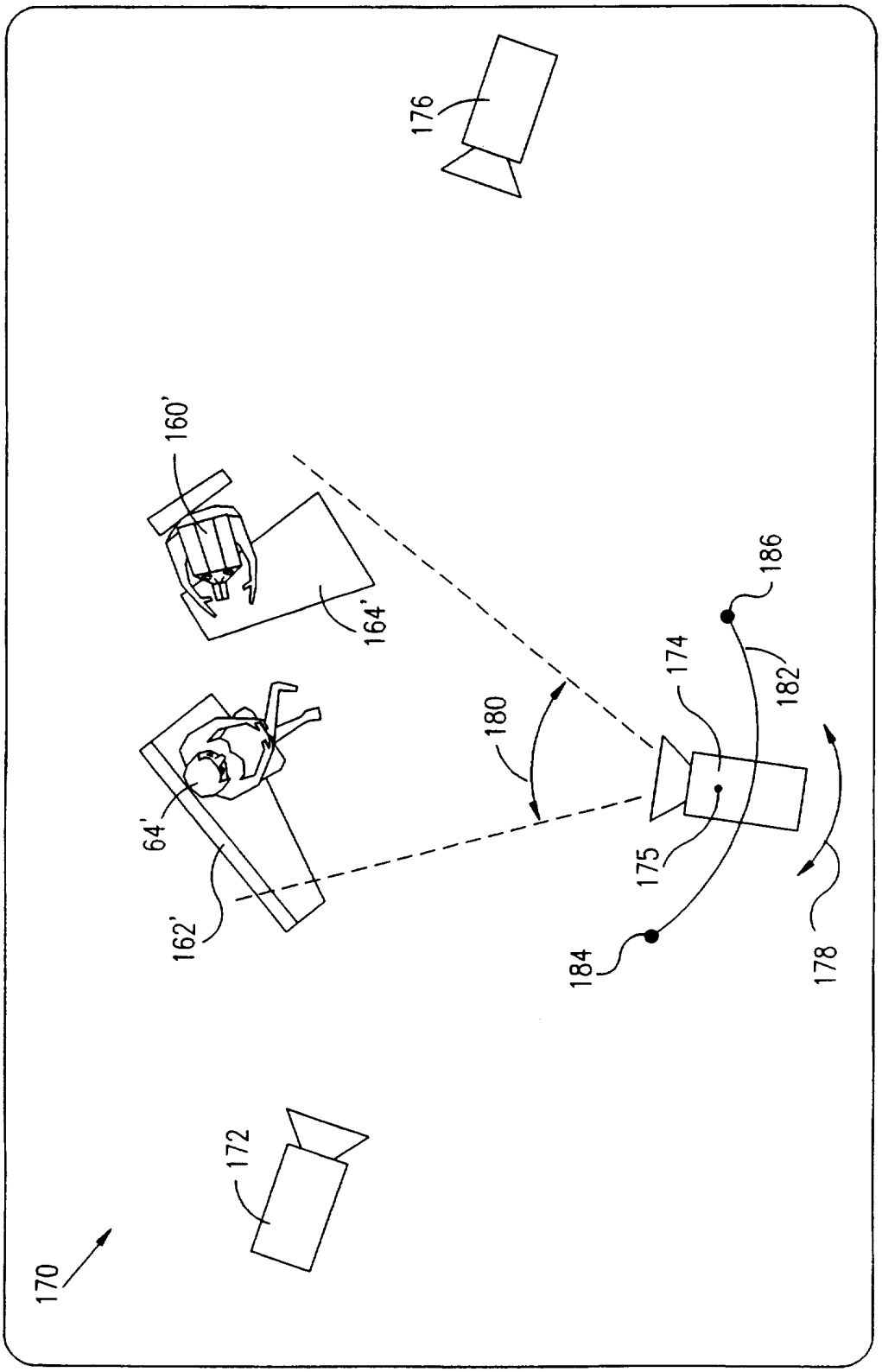


FIG. 10A

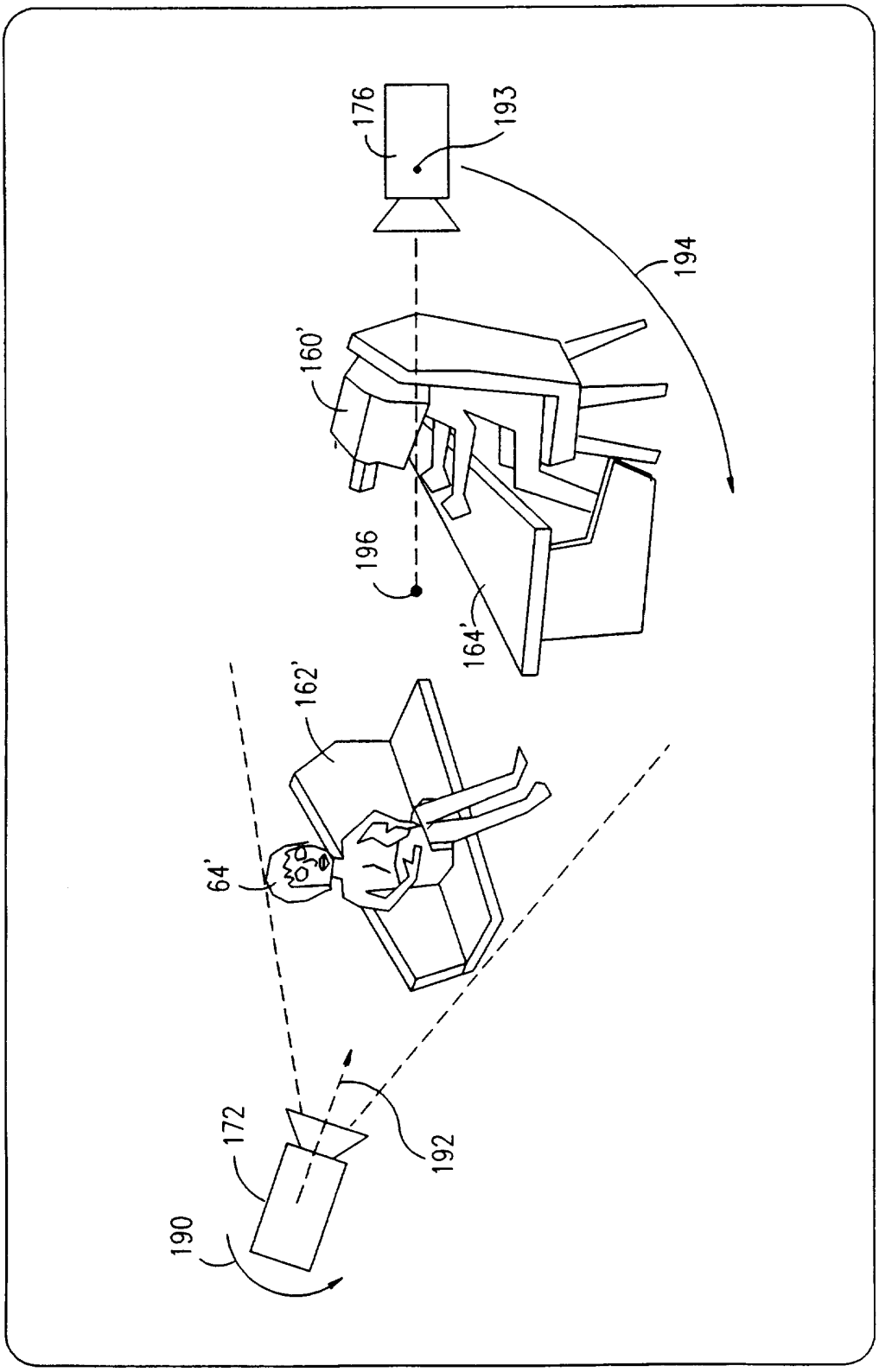
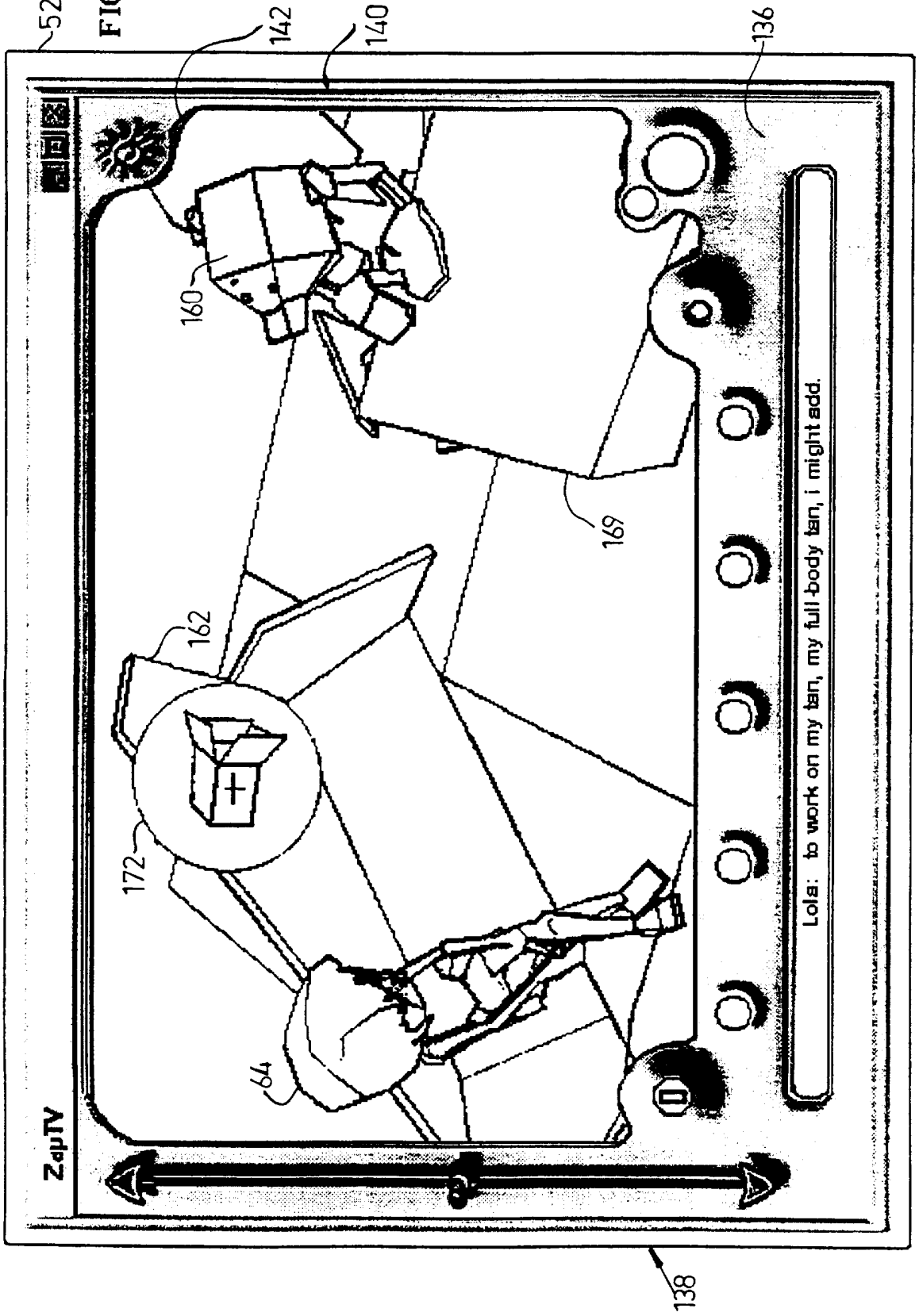


FIG. 10B

FIG. 11A


52
142
140
136
160
169
162
172
64
138
ZapTV
Lola: to work on my tan, my full body tan, i might add.

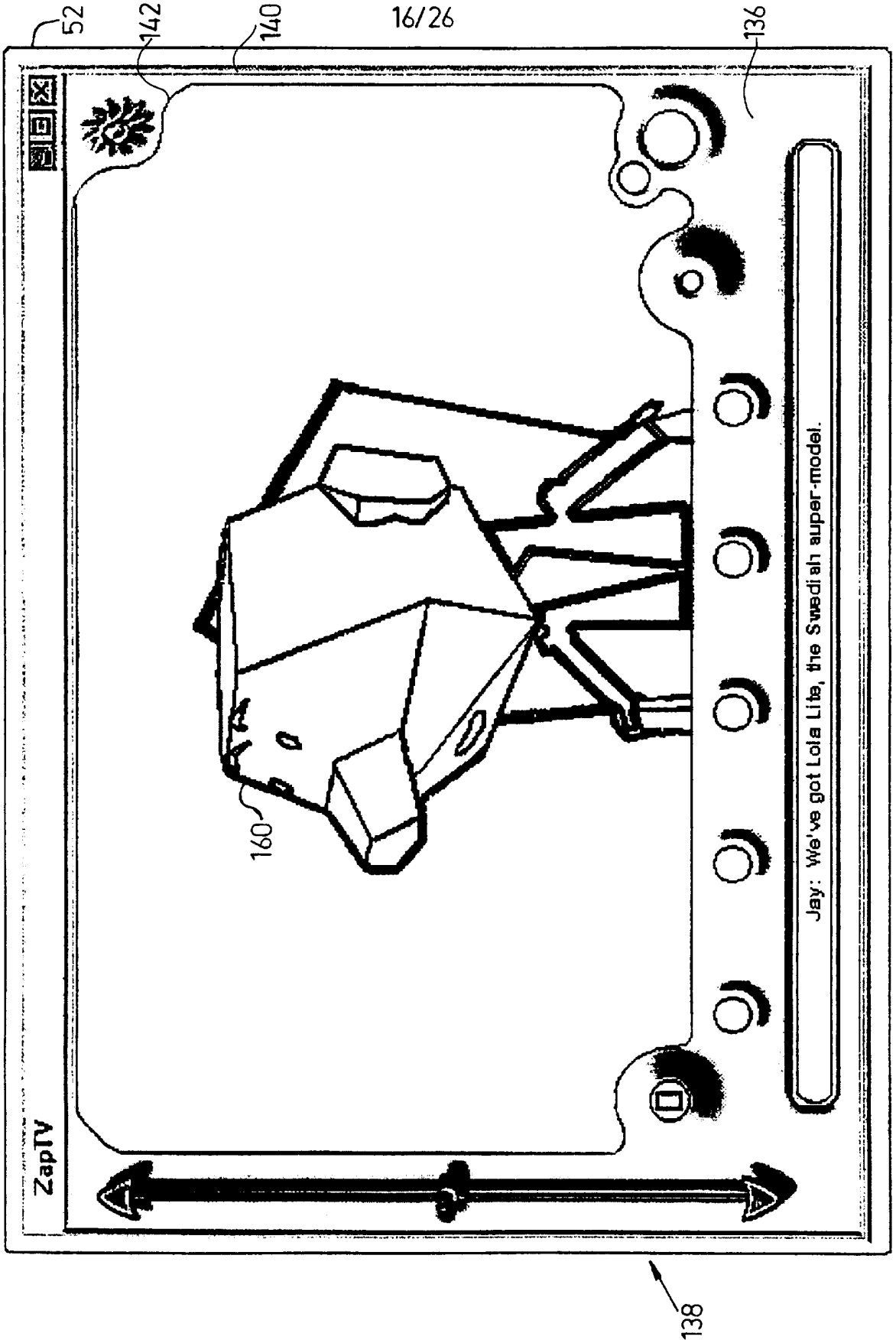

52
142
140
136
160
138
ZapTV
Jay: We've got Lola Lita, the Swedish super-model.

FIG. 11B

FIG. 12

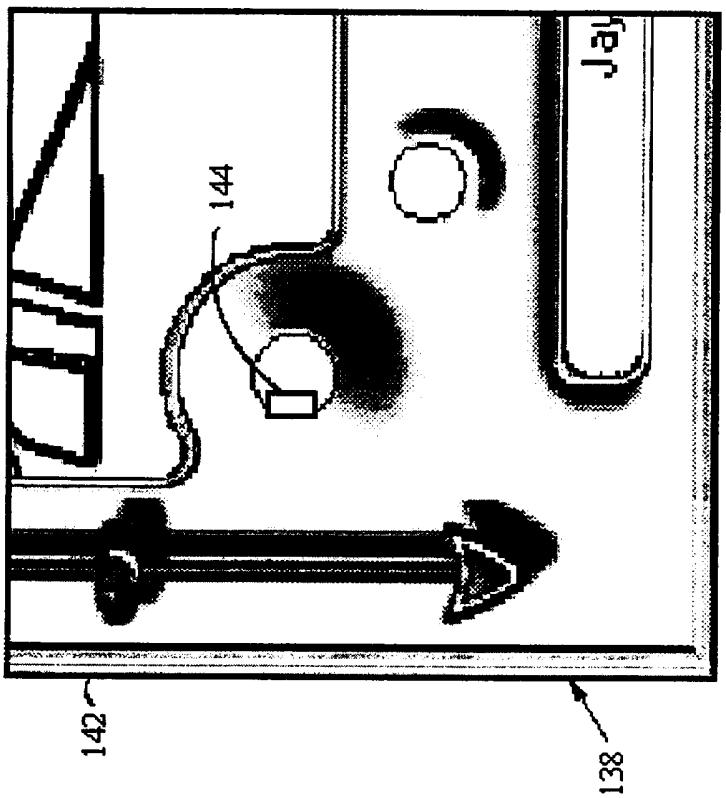


FIG. 13A

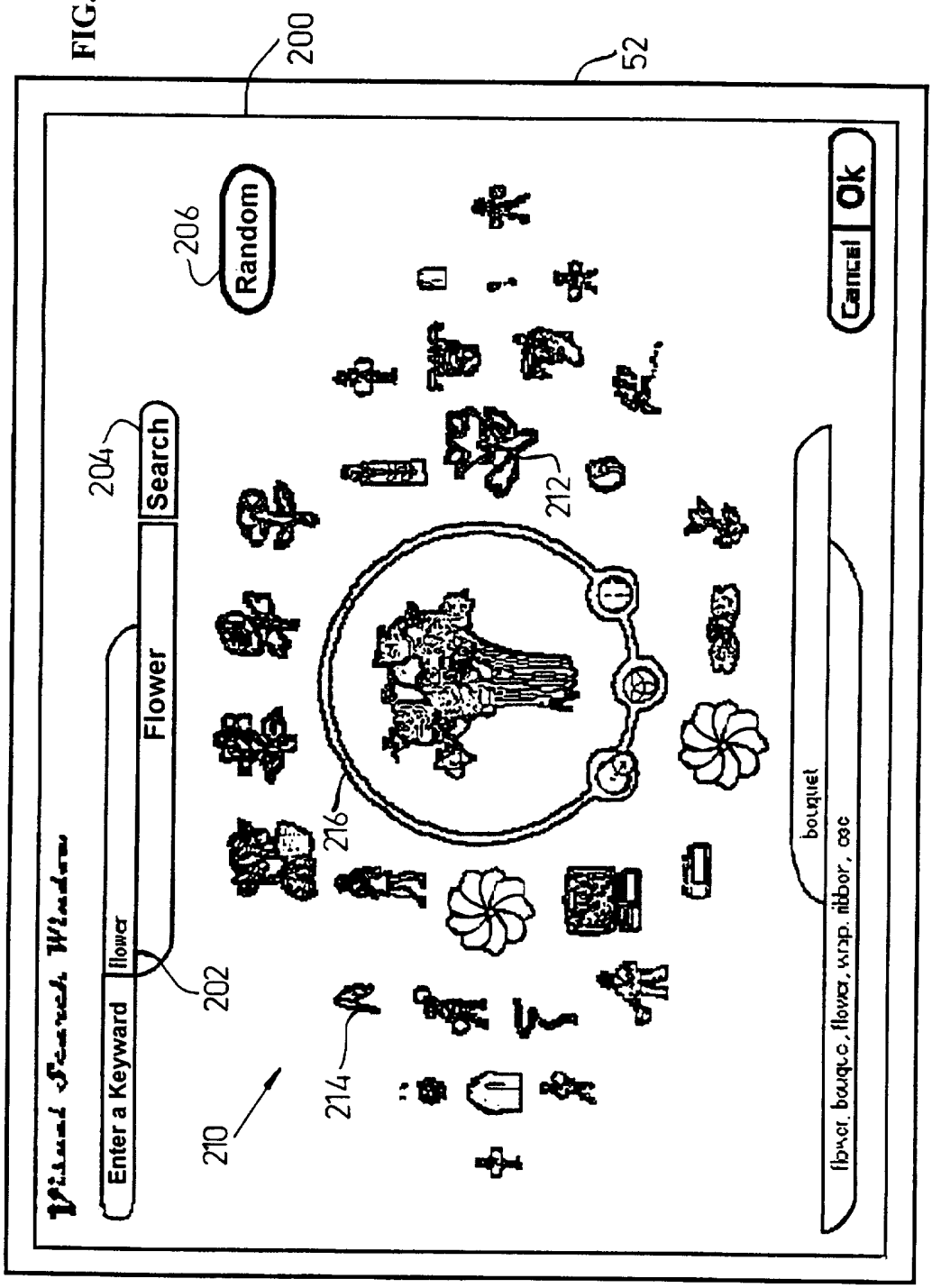
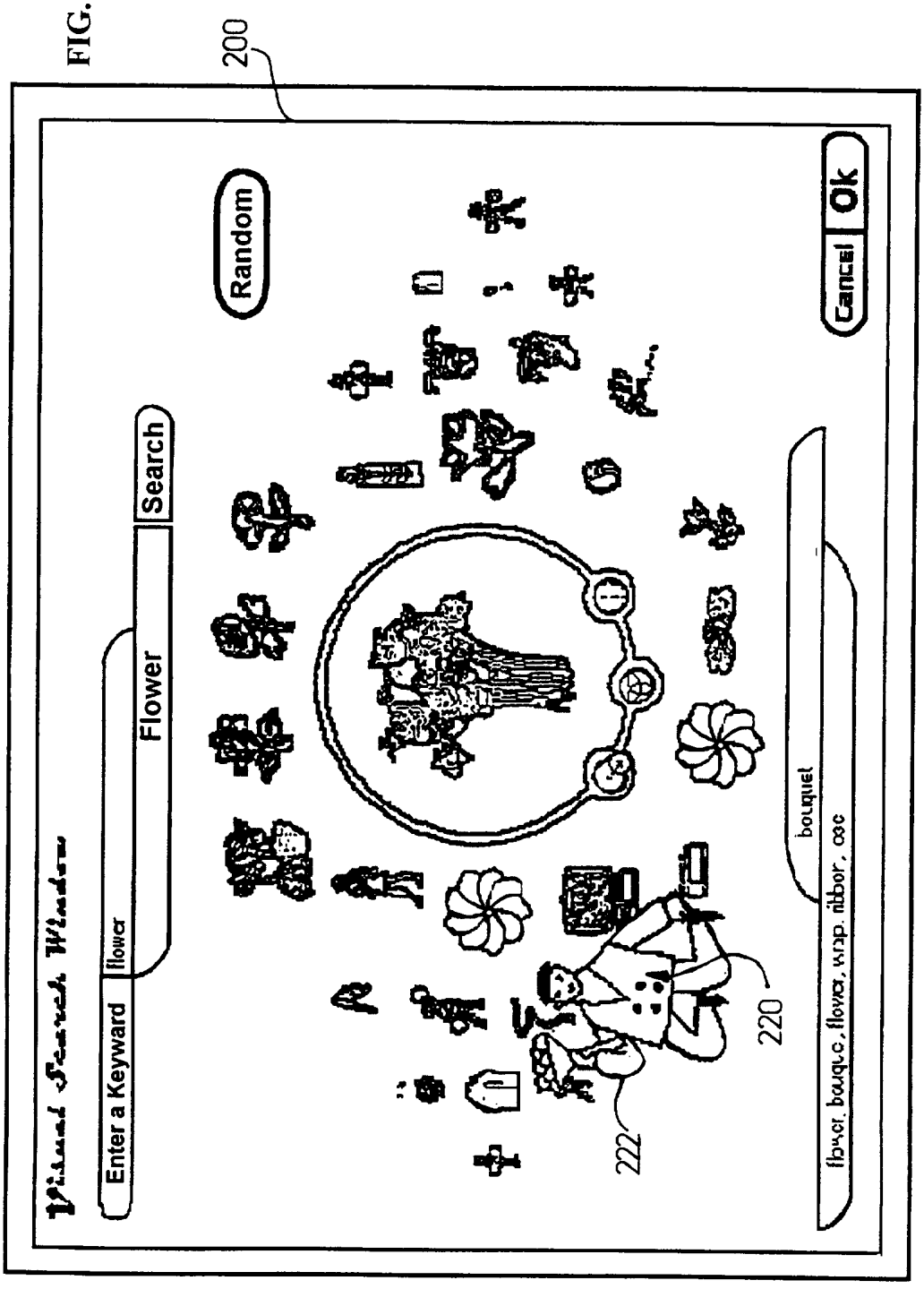

200
52
206
Random
204
Search
Flower
Enter a Keyword
flower
202
210
214
216
212
Cancel
Ok
bouquet
flower, bouque, flowe, wrap, ribbor, oac

FIG.13B


200
Visual Search Window
Enter a Keyword
flower
Flower
Search
Random
Cancel
Ok
bouquet
flower, bouque, flower, wrap, ribbor, cac
220
222

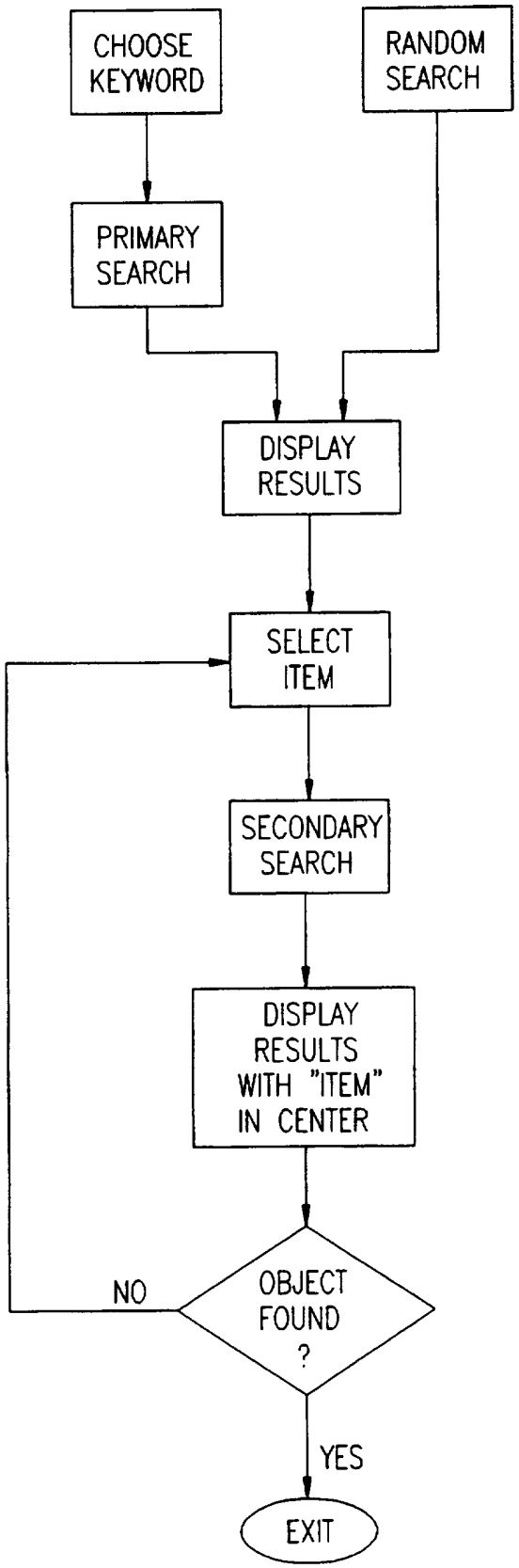

CHOOSE KEYWORD
RANDOM SEARCH
PRIMARY SEARCH
DISPLAY RESULTS
SELECT ITEM
SECONDARY SEARCH
DISPLAY RESULTS WITH "ITEM" IN CENTER
OBJECT FOUND ?
NO
YES
EXIT

FIG. 14

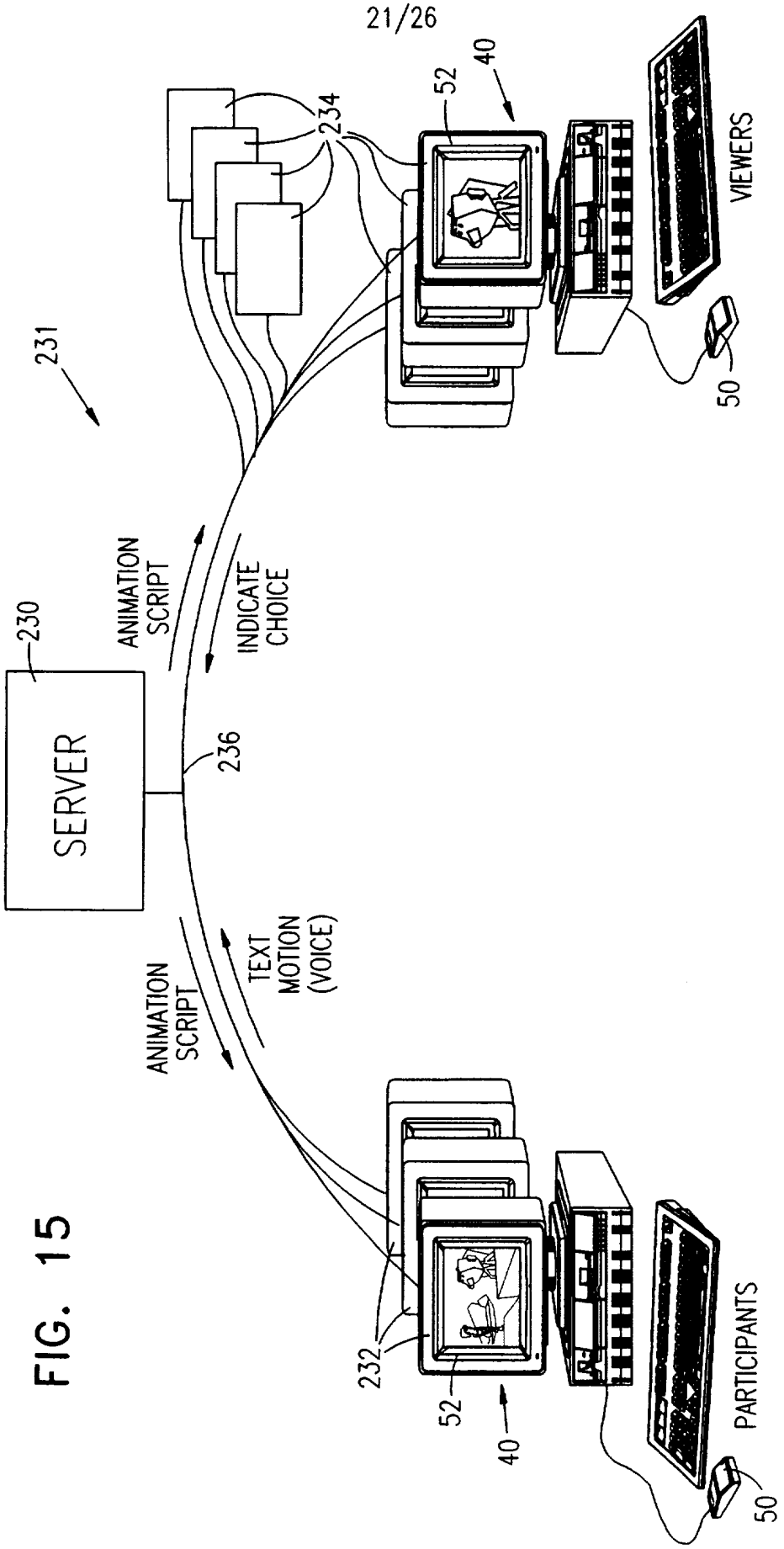

231
234
52
40
VIEWERS
50
ANIMATION SCRIPT
INDICATE CHOICE
230
SERVER
236
TEXT MOTION (VOICE)
ANIMATION SCRIPT
232
52
40
PARTICIPANTS
50

FIG. 15

FIG. 16

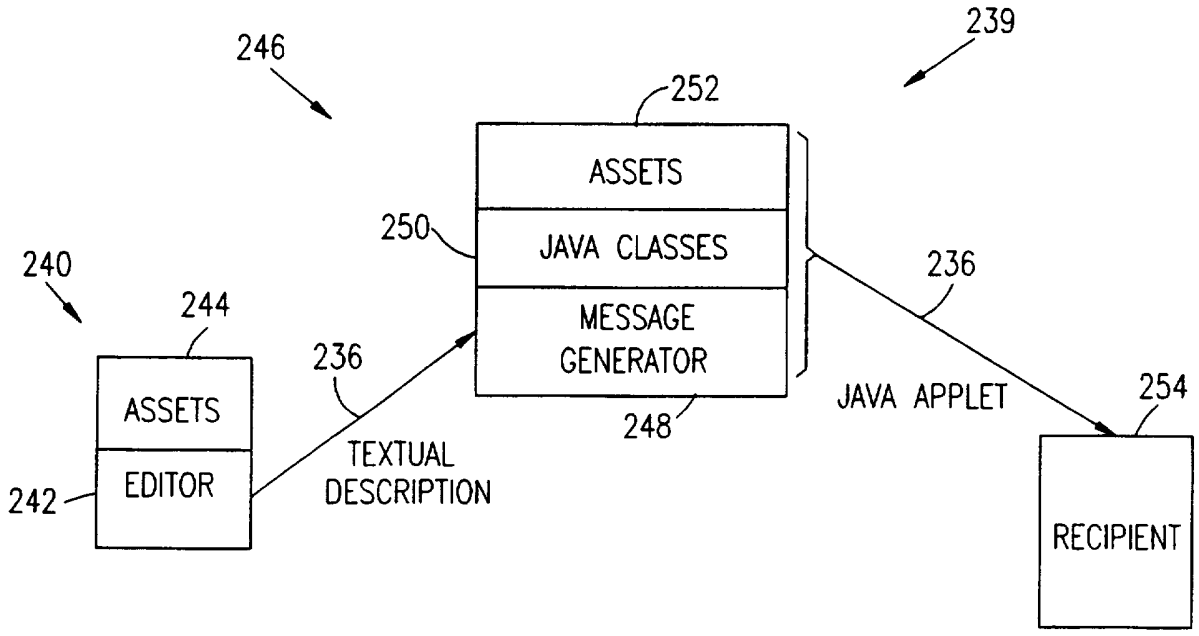
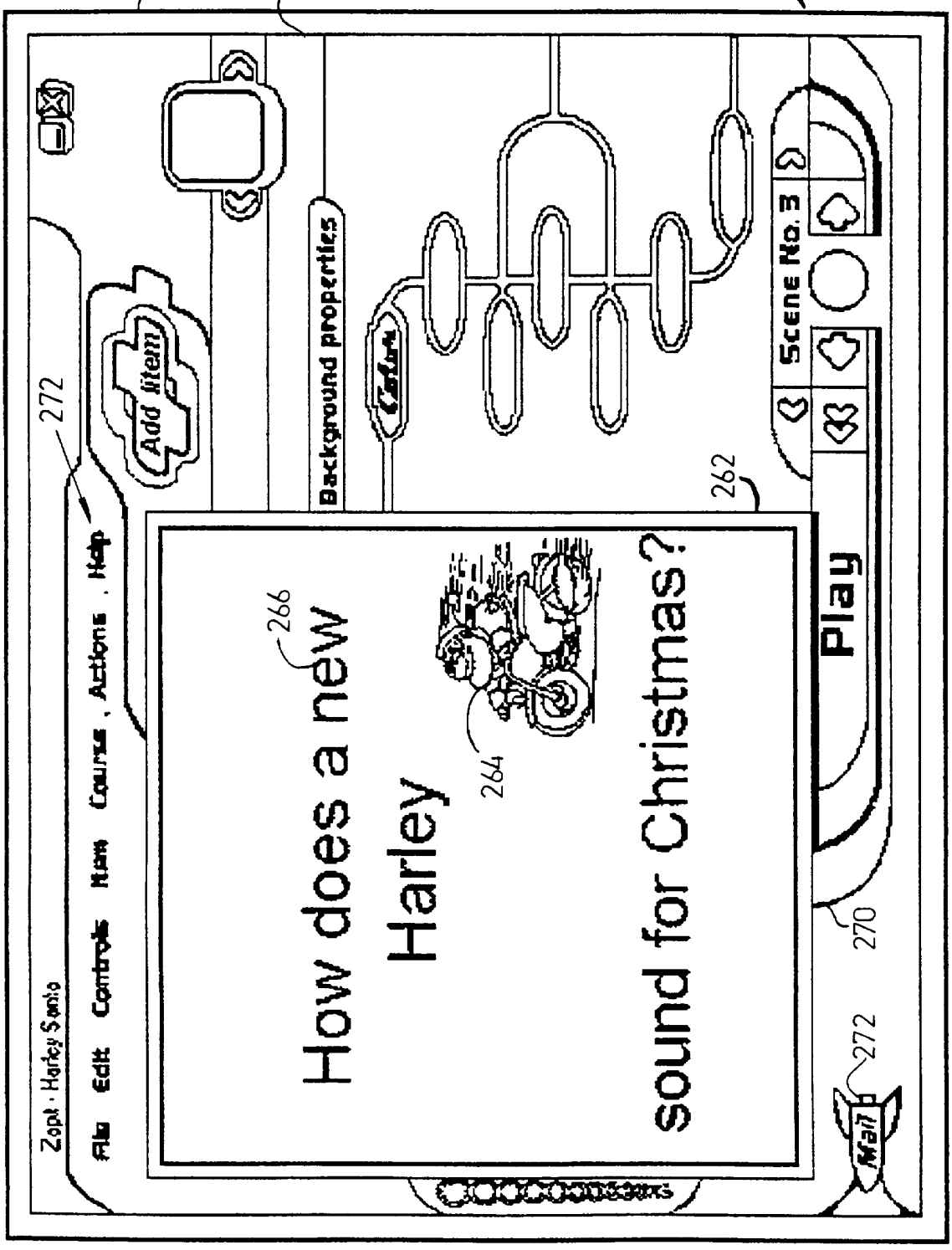

246
239
252
ASSETS
250
JAVA CLASSES
240
236
244
MESSAGE
GENERATOR
236
ASSETS
JAVA APPLET
254
248
TEXTUAL
DESCRIPTION
EDITOR
242
RECIPIENT

FIG. 17A


52
260
268
272
Add item
Background properties
Scene No. 3
262
Play
266
How does a new
Harley
sound for Christmas?
264
270
272
Mail
Zapt · Harley Santa
File Edit Controls New Course, Actions, Help

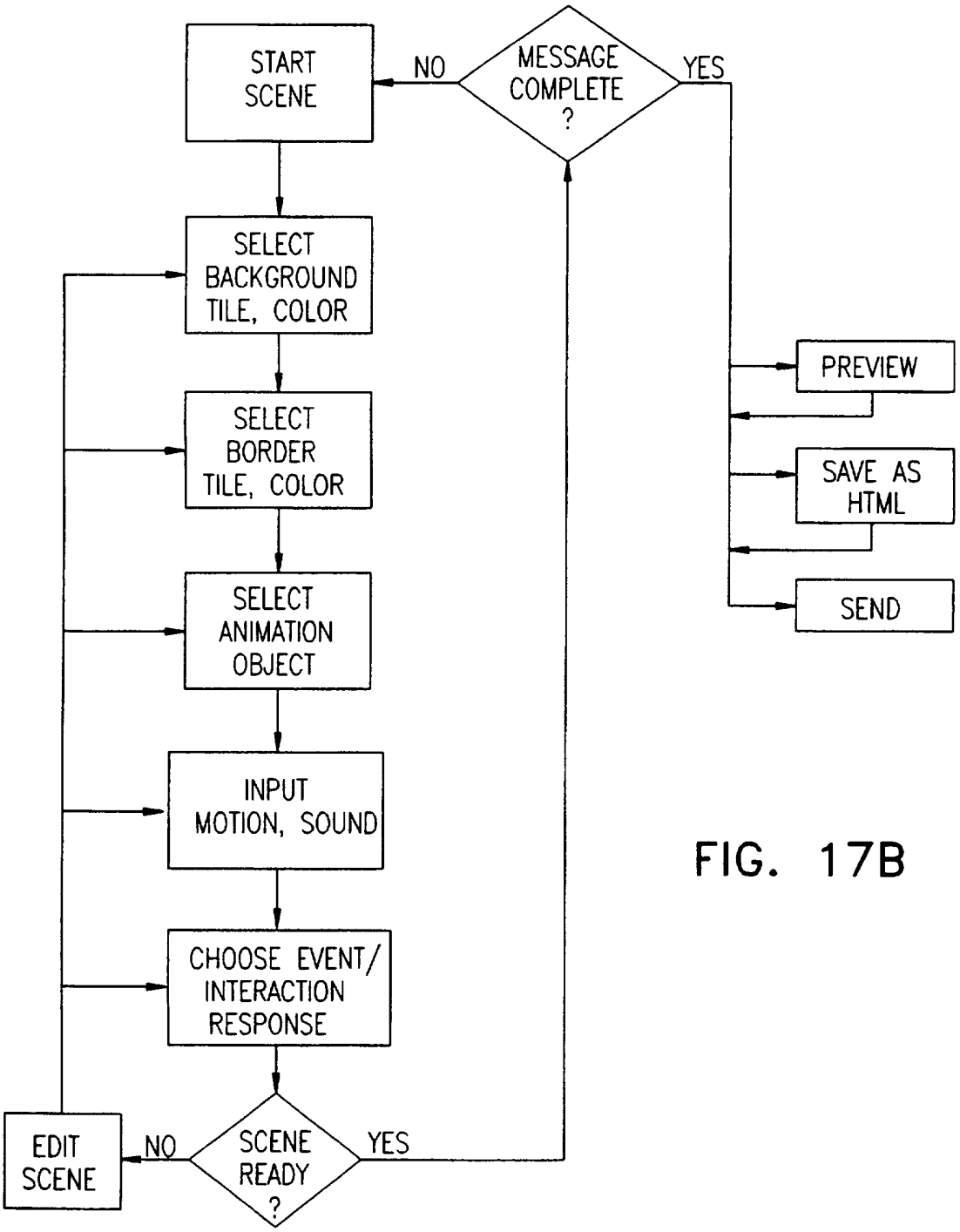

START SCENE
NO
MESSAGE COMPLETE ?
YES
SELECT BACKGROUND TILE, COLOR
SELECT BORDER TILE, COLOR
SELECT ANIMATION OBJECT
INPUT MOTION, SOUND
CHOOSE EVENT/ INTERACTION RESPONSE
EDIT SCENE
NO
SCENE READY ?
YES
PREVIEW
SAVE AS HTML
SEND

FIG. 17B

FIG. 18A

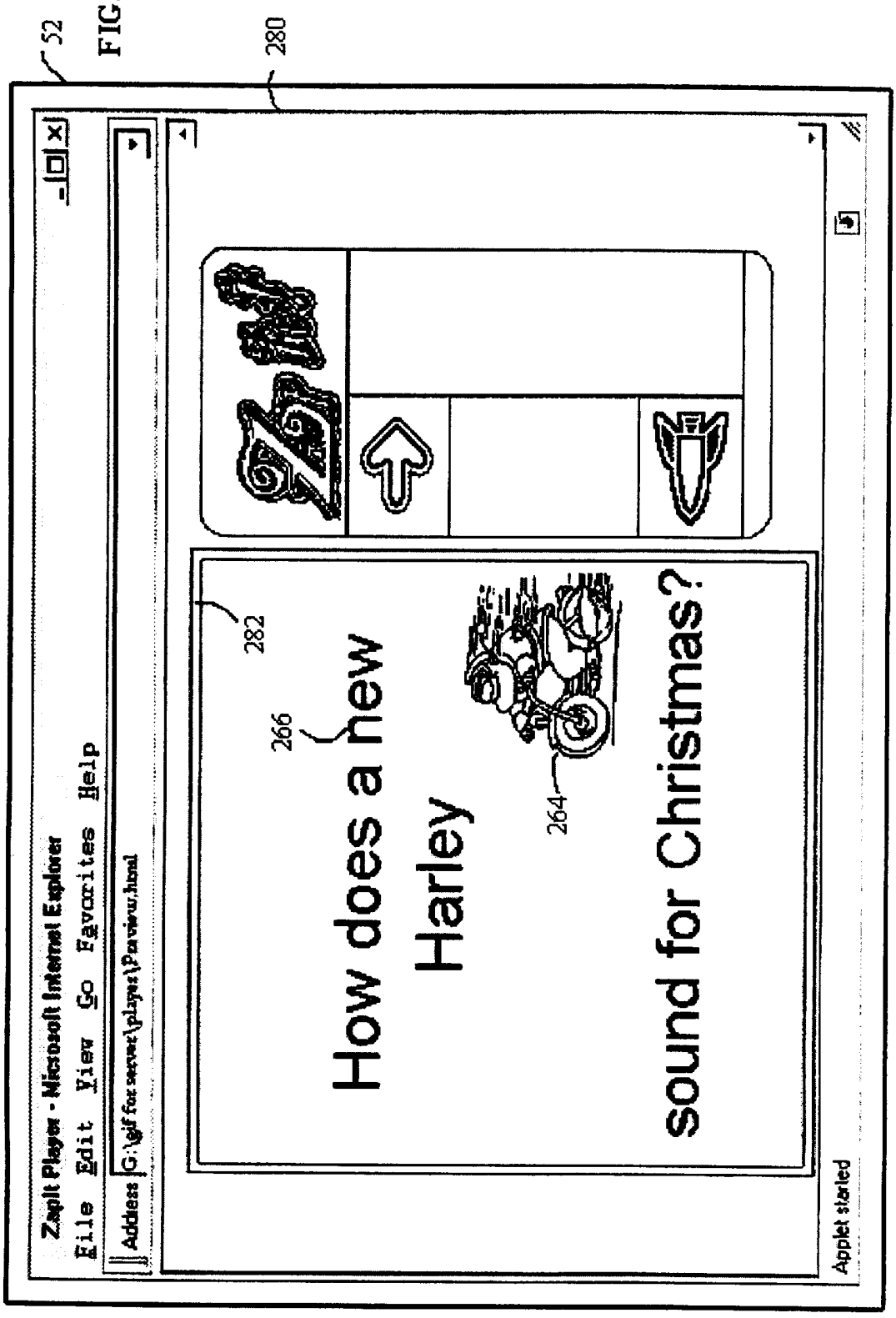
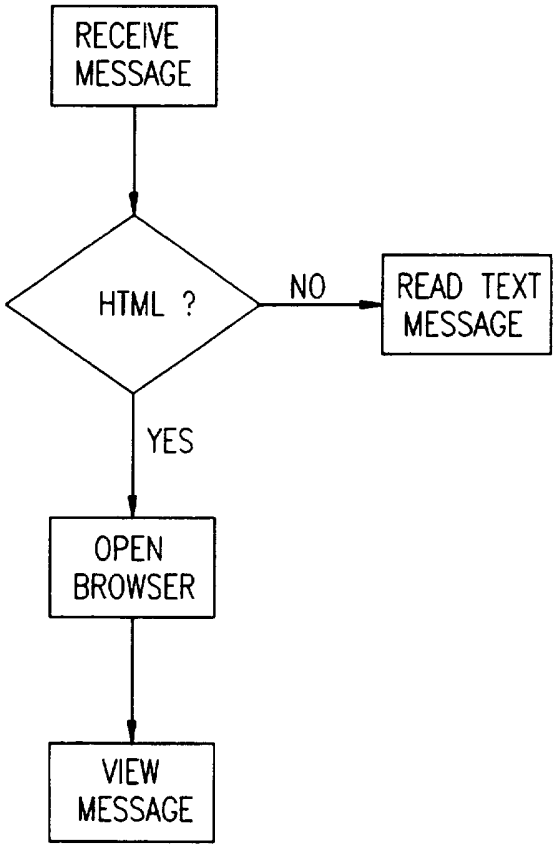

52
280
Zapit Player - Microsoft Internet Explorer
File Edit View Go Favorites Help
Address G:\gif for server\player1\Preview.html
282
266
How does a new
Harley
264
sound for Christmas?
Applet started

FIG. 18B


RECEIVE MESSAGE
HTML ?
NO
READ TEXT MESSAGE
YES
OPEN BROWSER
VIEW MESSAGE