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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
30 December 2009 (30.12.2009)

PCT

(10) International Publication Number
WO 2009/158628 A2

(51) International Patent Classification:
G06Q 50/00 (2006.01)

(21) International Application Number:
PCT/US2009/048874

(22) International Filing Date:
26 June 2009 (26.06.2009)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
12/163,345 27 June 2008 (27.06.2008) US

(71) Applicant *(for all designated States except US)*: **MICROSOFT CORPORATION** [US/US]; One Microsoft Way, Redmond, WA 98052-6399 (US).

(72) Inventors: **FILER, Eric, P.**; c/o Microsoft Corporation International Patents, One Microsoft Way, Redmond, WA 98052-6399 (US). **REAS, Loren Douglas**; c/o Microsoft Corporation International Patents, One Microsoft Way, Redmond, WA 98052-6399 (US). **RUBIO, Vasco**; c/o Microsoft Corporation International Patents, One Microsoft Way, Redmond, WA 98052-6399 (US). **TOM, Dennis W.**; c/o Microsoft Corporation International Patents, One Microsoft Way, Redmond, WA 98052-6399 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

Published:

— *without international search report and to be republished upon receipt of that report (Rule 48.2(g))*

(54) Title: DYNAMIC SELECTION OF SENSITIVITY OF TILT FUNCTIONALITY

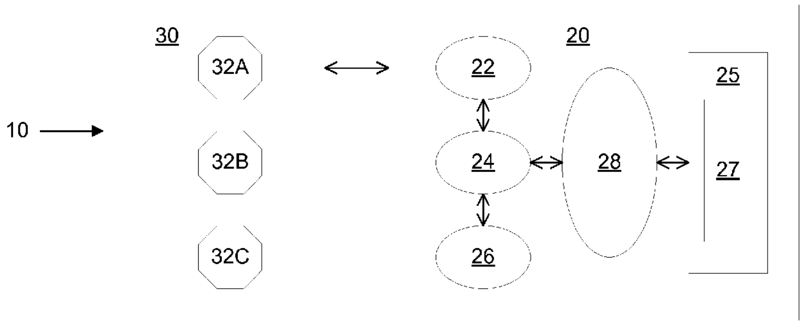


FIG. 3

(57) Abstract: Disclosed is a gaming system having a processing device and a remote input device that is operationally coupled to the processing device. The remote input device may include a motion sensor. The resolution of the motion sensor may be set dynamically from the game software, such that both gross and fine gestures can have the maximum effect. By enabling the game software to assess and control the resolution requirements, and enabling the input device to adjust and respond accordingly, relatively fine gestures, as well as relatively gross gestures, can be discerned and depicted with better accuracy and precision.

WO 2009/158628 A2

DYNAMIC SELECTION OF SENSITIVITY OF TILT FUNCTIONALITY

BACKGROUND

[0001] Gaming systems are known in which the gestures of a player are mimicked in an animated depiction of the player. As used herein, the term "gestures" may refer to movements of the player, or corresponding movements of the animated depiction of the player. Examples of such gestures include movements of all or part of a body, which may include movements of a body member, such as a hand, arm, head, face, *etc.*

[0002] In such a system, gestures are typically detected by a motion sensor in a remote gaming input device handled by the player, and communicated from the remote device to the gaming system processor. Examples of such motion sensors include gyros, magnetometers, and accelerometers. The palette of supported gestures is typically limited by the pre-set resolution of the motion sensor. That is, sensitivity to gestures is typically limited to the resolution to which the motion sensor has been set.

[0003] To achieve the fullest range of a particular gesture input for a gaming input device, the player typically needs to manually change the sensitivity of the motion sensor. However, if the player selects a fine sensor (*i.e.*, a sensor with relatively high sensitivity) and performs a gross gesture, then the sensor may tend to clip. Conversely, if the player selects a gross sensor (*i.e.*, a sensor with relatively low sensitivity) and performs a fine gesture, then depiction of the fine motion tends to get blurred in noise. In either scenario, data may be lost.

[0004] It would be desirable, therefore, to have a gaming system where the resolution could be set dynamically from the game software, such that both gross and fine gestures can have the maximum effect. By enabling the game software to assess and control the resolution requirements, and enabling the input device to adjust and respond accordingly, relatively fine gestures, as well as relatively gross gestures, could be discerned and depicted with better accuracy and precision.

SUMMARY

[0005] As described herein, a computer system, such as a gaming system, for example, may include a processing device and a remote input device. The remote input device may be operationally coupled to provide input to the processing device. The remote input device may be wirelessly coupled to the processing device.

[0006] The remote input device may include one or more motion sensors, each having one or more sensitivity ranges. For example, the remote input device may include one or more motion sensors, each having a plurality of selectable sensitivity ranges. Alternatively or additionally, the remote input device may include a plurality of motion sensors, each having at least one sensitivity range.

[0007] The processing device may include a context-determining module, a sensitivity-determining module, and a communications module. The context-determining module may be configured to ascertain a current context in an application, such as a game-playing application, for example, executing on a computing device. For example, the context-determining module may be configured to ascertain a current scripted situation within the game playing application, or to ascertain a user profile.

[0008] The sensitivity-determining module may be configured to receive information from the context-determining module, and to determine a desired sensitivity range for a remote input device. The sensitivity-determining module may be configured to determine the desired sensitivity range based at least in part on the user profile.

[0009] The communications module may be configured to communicate information indicative of the desired sensitivity range to the remote input device. For example, the processing device may signal the remote input device to select one of the sensors from the plurality of sensors, and/or to select one of the sensitivity ranges from the plurality of sensitivity ranges.

[0010] The remote input device may be configured to receive communicated information that is indicative of a desired sensitivity range. The remote input device may be configured to respond to the received information by operating in the desired sensitivity range. For example, the remote input device may be configured to respond to the received information by activating a particular physical sensor having a sensitivity range corresponding to the desired sensitivity range.

BRIEF DESCRIPTION OF THE DRAWINGS

[0011] FIG. 1 depicts an image of a gross gesture.

[0012] FIGs. 2A-C depict images of fine gestures at various times in tilt mode.

[0013] FIG. 3 is a functional block diagram of an example computing system.

[0014] FIG. 4 is a flowchart of an example method for use in a computing system as depicted in FIG. 1.

[0015] FIG. 5 is a block diagram of an example computing environment in which example embodiments and aspects of the present invention may be implemented.

[0016] FIG. 6 is an example network configuration in which aspects of the invention may be implemented.

DETAILED DESCRIPTION OF ILLUSTRATIVE EMBODIMENTS

Overview; Example Scenarios

[0017] An example scenario in which the systems and methods described herein may be used is now presented in the context of a gaming system. It should be understood, however, that a game system is described for illustrative purposes only, and that the systems and methods described herein are not limited to implementation in gaming systems.

[0018] A typically gaming system may include a game console. A processing device, on which the game's operational software application may be executed, may be housed in the game console. The gaming system may also include a remote input device, the nature of which may be based on the actual game the player is playing. The remote input device may communicate to the game console information corresponding to the player's gestures using the remote input device. The game console may cause a depiction of the player's gestures, or the effects thereof, to be presented on a video display, such as a television, computer monitor, or dedicated video display to which the game console is operationally coupled.

[0019] Consider an example scenario wherein a player is playing a game of golf. Accordingly, the remote input device may represent a golf club. The player's gestures may be characterized by the player's swinging of the golf club. The effects of the player's gestures may be characterized by the golf club's being swung.

[0020] In an example scenario, there may be three types of golf swings, *i.e.*, driving, chipping, and putting. It should be understood that, in general, a player will tend to swing harder (*i.e.*, faster and over a greater angle) when driving than when chipping. Similarly, a player will tend to swing harder when chipping than when putting. Accordingly, to present accurate and precise depictions of both driving and putting, greater motion sensitivity may be desirable during a putting gesture than during a driving gesture.

[0021] The system may be capable of recognizing gestures used in such a game-play scenario, and adjusting the hardware sensitivity dynamically in response to such recognition. For example, the game software may be configured to decide when to switch resolution, and to determine the resolution to which to switch. Because a video game is typically a scripted interaction, the game software typically knows the context of the current situation. For example, in a golf scenario, the game software can recognize that, if the ball is at the tee, the player is likely to be driving rather than putting. Similarly, if the ball is on the green, the player is likely to be putting rather than driving. Or the context could be identified based on club selection. For example, if the player selects a driver, he is likely about to drive. If he selects a putter, he is likely about to putt.

[0022] The game software can recognize the context, and determine a desired sensitivity from the context. As play moves from a driving context to a chipping context to a putting context, the processing device may signal the remote input device to select progressively more sensitive sensors. Thus, driving gestures, chipping gestures, and putting gestures, along with the effects thereof, can be depicted accurately and precisely.

[0023] FIG. 1 depicts an example image of a gross gesture. As shown, an image of a person holding a microphone is presented. The gross gesture corresponds to the singer swinging her arms over about a 60-degree angle rather quickly. To produce a clear image of such a gesture, relatively low motion sensitivity would be desirable.

[0024] FIGs. 2A-C depict example images of fine gestures at various times in tilt mode. As shown, the singer is now titling the microphone relatively slowly over a relatively small angle (*e.g.*, at a rate of 10 degrees every 7 seconds). To produce a clear image of such a gesture, relatively high motion sensitivity would be desirable.

[0025] A detailed description of example systems and methods follows.

Dynamic Selection Of Sensitivity Of Tilt Functionality

[0026] FIG. 3 is a functional block diagram of an example computing system 10. As shown, the system 10 may include a computing or processing device 20, and a remote input device 30. The processing device 20 may be housed in a game console, for example. The remote input device 30 may be operationally coupled to provide input to the processing device 20. The remote input device 30 may be wired to the processing device 20, or wirelessly coupled to the processing device 20.

[0027] The remote input device 30 may include a human-interface device, such as a ball, bat, drumstick, fishing rod, or microphone, for example, including any type of game controller, such as a joystick, headset, helmet, heads-up display, or the like. The remote input device 30 may include gesture recognition hardware. The gesture recognition hardware may include one or more sensors, which may be motion sensors, thermal sensors, or pressure sensors, for example, or any combination of such sensors. The remote input device 30 may be robotic device, of a type that might be used in manufacturing, for example.

[0028] The remote input device 30 may be operable over a plurality of sensitivity ranges. The remote input device 30 may include one or more physical motion sensors 32A-C. Examples of such motion sensors include gyros, accelerometers, and magnetometers. Typically, a single motion sensor or type of motion sensor will not provide absolute positioning of a moving object. Accordingly, multiple, different sensors may be employed. For example, an accelerometer may be used for measuring movement, while an additional sensor (*e.g.*, a gyro) may be employed for determining position.

[0029] Each of the one or more physical motion sensors 32A-C may be operable over a plurality of selectable sensitivity ranges. The remote input device 30 may include a plurality of physical motion sensors 32A-C, each of which is operable in at least one sensitivity range. It should be understood that the systems and methods described herein are not limited to the use of motion sensors. For example, thermal or pressure sensors could be employed.

[0030] The processing device 20 may include a context-determining module 22, a sensitivity-determining module 24, and a communications module 26. The context-determining module 22 may be configured to ascertain a current context in an application 28 executing on the processing device 20. For example, the application 28 may be a game-playing application. The context-determining module 26 may be configured to ascertain a current scripted situation within the game-playing application 28.

[0031] The context-determining module 22 may be configured to ascertain a user profile 27. The processing device 20 may include a memory 25 in which the user profile 27 is stored. An example user profile may include one or more predefined preferences of a specific user. Examples of such preferences include presets of default settings for gain, sensitivity, and personalization. This may be accessed through a password or biometric sensor, for example. Multiple profiles may be stored at once.

[0032] The sensitivity-determining module 24 may be configured to receive information from the context-determining module 22, and to determine a desired sensitivity range for the remote input device 30. The desired sensitivity range may be determined based at least in part on the current context in the application 28 executing on the processing device 20. For example, the desired sensitivity range may be determined based at least in part on the current scripted situation within a game-playing application. The sensitivity-determining module 24 may be configured to determine the desired sensitivity range based at least in part on the user profile 27.

[0033] The communications module 26 may be configured to communicate information indicative of the desired sensitivity range to the remote input device 30, for use in selecting a sensitivity range for the remote input device 30. The processing device 20 may signal the remote input device 30 to operate in the desired sensitivity range. The signal may be transmitted over a wired or wireless connection between the processing device 20 and the remote input device 30. Thus, the processing device 20 can send a control signal to the sensor in the remote input device 30 to set the sensitivity of the sensor.

[0034] Such a signal may include a field that informs the remote input device 30 of the desired sensitivity range. For example, the signal may include a number of bits (*e.g.*, two) that correspond to the desired scale setting. The number of bits (and, accordingly, the range of sensitivities) may be a parametric value that may be adjustable via the processing device.

[0035] The remote input device 30 may receive the signal from the processing device 20, and thus receive from the processing device 20 information indicative of the desired sensitivity range. The remote input device 30 may respond to receiving the information communicated from the processing device 20 by operating in the desired sensitivity range.

[0036] For example, the remote input device 30 may respond to receiving the communicated information by causing a selected one of the physical motion sensors to operate in a selected one of the plurality of sensitivity ranges. Where the remote input device 30 includes a plurality of physical motion sensors, the remote input device 30 may cause to operate a selected one of the physical motion sensors that is operable in the desired sensitivity range. Where the remote input device 30 includes a physical motion sensor that is operable over a plurality of selectable sensitivity ranges, the remote input device 30 may cause the physical motion sensor to operate in the desired sensitivity range by selecting the desired sensitivity range from the plurality of sensitivity ranges over which the motion sensor is operable.

[0037] To summarize, FIG. 4 provides a flowchart of an example method 60 for use in a computing system as depicted in FIG. 3. At 62, a desired sensitivity range for the remote input device may be determined at the processing device. As shown at 64, the determination at 62 may be based, at least in part, on a current context in an application executing on the system. As shown at 66, the determination at 62 may be based, at least in part, on a user profile.

[0038] At 68, the processing device may signal the remote input device to operate in the desired sensitivity range. The processing device may signal the remote input device to cause a motion sensor in the remote input device to operate in a selected sensitivity range, as at 70. Alternatively or additionally, the processing device may signal the remote input device to cause a selected one of a plurality of motion sensors to operate in the desired sensitivity range, as at 72.

Example Computing Environment

[0039] FIG. 5 shows an exemplary computing environment in which example embodiments and aspects may be implemented. The computing system environment 100 is only one example of a suitable computing environment and is not intended to suggest any limitation as to the scope of use or functionality. Neither should the computing environment 100 be interpreted as having any dependency or requirement relating to any one or combination of components illustrated in the exemplary operating environment 100.

[0040] Numerous other general purpose or special purpose computing system environments or configurations may be used. Examples of well known computing systems, environments, and/or configurations that may be suitable for use include, but are not limited to, personal computers, server computers, hand-held or laptop devices, multiprocessor systems, microprocessor-based systems, set top boxes, programmable consumer electronics, network PCs, minicomputers, mainframe computers, embedded systems, distributed computing environments that include any of the above systems or devices, and the like.

[0041] Computer-executable instructions, such as program modules, being executed by a computer may be used. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. Distributed computing environments may be used where tasks are performed by remote processing devices that are linked through a communications network or other data transmission medium. In a distributed computing environment, program modules and other data may be located in both local and remote computer storage media including memory storage devices.

[0042] With reference to FIG. 5, an exemplary system includes a general purpose computing device in the form of a computer 110. Components of computer 110 may include, but are not limited to, a processing unit 120, a system memory 130, and a system bus 121 that couples various system components including the system memory to the processing unit 120. The processing unit 120 may represent multiple logical processing units such as those supported on a multi-threaded processor. The system bus 121 may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures. By way of example, and not limitation, such architectures include Industry Standard Architecture (ISA) bus, Micro Channel Architecture (MCA) bus, Enhanced ISA (EISA) bus, Video Electronics Standards Association (VESA) local bus, and Peripheral Component Interconnect (PCI) bus (also known as Mezzanine bus). The system bus 121 may also be implemented as a point-to-point connection, switching fabric, or the like, among the communicating devices.

[0043] Computer 110 typically includes a variety of computer readable media. Computer readable media can be any available media that can be accessed by computer 110 and includes both volatile and nonvolatile media, removable and non-removable media. By way of example, and not limitation, computer readable media may comprise computer storage media and communication media. Computer storage media includes both volatile and nonvolatile, removable and non-removable media implemented in any method or technology for storage of information such as computer readable instructions, data structures, program modules or other data. Computer storage media includes, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CDROM, digital versatile disks (DVD) or other optical disk storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can accessed by computer 110. Communication media typically embodies computer readable instructions, data structures, program modules or other data in a modulated data signal such as a carrier wave or other transport mechanism and includes any information delivery media. The term “modulated data signal” means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media includes wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, RF, infrared and other wireless media. Combinations of any of the above should also be included within the scope of computer readable media.

[0044] The system memory 130 includes computer storage media in the form of volatile and/or nonvolatile memory such as read only memory (ROM) 131 and random access memory (RAM) 132. A basic input/output system 133 (BIOS), containing the basic routines that help to transfer information between elements within computer 110, such as during start-up, is typically stored in ROM 131. RAM 132 typically contains data and/or program modules that are immediately accessible to and/or presently being operated on by processing unit 120. By way of example, and not limitation, FIG. 5 illustrates operating system 134, application programs 135, other program modules 136, and program data 137.

[0045] The computer 110 may also include other removable/non-removable, volatile/nonvolatile computer storage media. By way of example only, FIG. 5 illustrates a hard disk drive 140 that reads from or writes to non-removable, nonvolatile magnetic media, a magnetic disk drive 151 that reads from or writes to a removable, nonvolatile magnetic disk 152, and an optical disk drive 155 that reads from or writes to a removable, nonvolatile optical disk 156, such as a CD ROM or other optical media. Other removable/non-removable, volatile/nonvolatile computer storage media that can be used in the exemplary operating environment include, but are not limited to, magnetic tape cassettes, flash memory cards, digital versatile disks, digital video tape, solid state RAM, solid state ROM, and the like. The hard disk drive 141 is typically connected to the system bus 121 through a non-removable memory interface such as interface 140, and magnetic disk drive 151 and optical disk drive 155 are typically connected to the system bus 121 by a removable memory interface, such as interface 150.

[0046] The drives and their associated computer storage media discussed above and illustrated in FIG. 5, provide storage of computer readable instructions, data structures, program modules and other data for the computer 110. In FIG. 5, for example, hard disk drive 141 is illustrated as storing operating system 144, application programs 145, other program modules 146, and program data 147. Note that these components can either be the same as or different from operating system 134, application programs 135, other program modules 136, and program data 137. Operating system 144, application programs 145, other program modules 146, and program data 147 are given different numbers here to illustrate that, at a minimum, they are different copies. A user may enter commands and information into the computer 20 through input devices such as a keyboard 162 and pointing device 161, commonly referred to as a mouse, trackball or touch pad. Other input devices (not shown) may include a microphone, joystick, game pad, satellite dish, scanner, or the like. These and other input devices are often connected to the processing unit 120 through a user input interface 160 that is coupled to the system bus, but may be connected by other interface and bus structures, such as a parallel port, game port or a universal serial bus (USB). A monitor 191 or other type of display device is also connected to the system bus 121 via an interface, such as a video interface 190. In addition to the monitor, computers may also include other peripheral output devices such as speakers 197 and printer 196, which may be connected through an output peripheral interface 195.

[0047] The computer 110 may operate in a networked environment using logical connections to one or more remote computers, such as a remote computer 180. The remote computer 180 may be a personal computer, a server, a router, a network PC, a peer device or other common network node, and typically includes many or all of the elements described above relative to the computer 110, although only a memory storage device 181 has been illustrated in FIG. 5. The logical connections depicted in FIG. 5 include a local area network (LAN) 171 and a wide area network (WAN) 173, but may also include other networks. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets and the Internet.

[0048] When used in a LAN networking environment, the computer 110 is connected to the LAN 171 through a network interface or adapter 170. When used in a WAN networking environment, the computer 110 typically includes a modem 172 or other means for establishing communications over the WAN 173, such as the Internet. The modem 172, which may be internal or external, may be connected to the system bus 121 via the user input interface 160, or other appropriate mechanism. In a networked environment, program modules depicted relative to the computer 110, or portions thereof, may be stored in the remote memory storage device. By way of example, and not limitation, FIG. 5 illustrates remote application programs 185 as residing on memory device 181. It will be appreciated that the network connections shown are exemplary and other means of establishing a communications link between the computers may be used.

Example Distributed Computing Frameworks or Architectures

[0049] Various distributed computing frameworks have been and are being developed in light of the convergence of personal computing and the Internet. Individuals and business users alike are provided with a seamlessly interoperable and web-enabled interface for applications and computing devices, making computing activities increasingly web browser or network-oriented.

[0050] For example, MICROSOFT®'s .NET platform includes servers, building-block services, such as web-based data storage, and downloadable device software.

Generally speaking, the .NET platform provides (1) the ability to make the entire range of computing devices work together and to have user information automatically updated and synchronized on all of them, (2) increased interactive capability for web sites, enabled by greater use of XML rather than HTML, (3) online services that feature customized access and delivery of products and services to the user from a central starting point for the management of various applications, such as e-mail, for example, or software, such as Office .NET, (4) centralized data storage, which will increase efficiency and ease of access to information, as well as synchronization of information among users and devices, (5) the ability to integrate various communications media, such as e-mail, faxes, and telephones, (6) for developers, the ability to create reusable modules, thereby increasing productivity and reducing the number of programming errors, and (7) many other cross-platform integration features as well.

[0051] While example embodiments herein are described in connection with software residing on a computing device, one or more portions of the invention may also be implemented via an operating system, API, or middleware software between a coprocessor and requesting object, such that services may be performed by, supported in, or accessed via all of .NET's languages and services, and in other distributed computing frameworks as well.

Network Environment

[0052] FIG. 6 illustrates an example network environment in which the present invention may be employed. Of course, actual network and database environments may be arranged in a variety of configurations; however, the example environment shown here provides a framework for understanding the type of environment in which an embodiment may operate.

[0053] The example network may include one or more client computers 200a, a server computer 200b, data source computers 200c, and/or databases 270, 272a, and 272b. The client computers 200a and the data source computers 200c may be in electronic communication with the server computer 200b by way of the communications network 280 (e.g., an intranet, the Internet or the like). The client computers 200a and data source computers 200c may be connected to the communications network by way of communications interfaces 282. The communications interfaces 282 can be any type of communications interfaces such as Ethernet connections, modem connections, wireless connections and so on.

[0054] The server computer 200b may provide management of the database 270 by way of database server system software such as MICROSOFT®'s SQL SERVER or the like. As such, server 200b may act as a storehouse of data from a variety of data sources and provides that data to a variety of data consumers.

[0055] In the example network environment of FIG. 6, a data source may be provided by data source computer 200c. Data source computer 200c may communicate data to server computer 200b via communications network 280, which may be a LAN, WAN, Intranet, Internet, or the like. Data source computer 200c may store data locally in database 272a, which may be database server or the like. The data provided by data source 200c can be combined and stored in a large database such as a data warehouse maintained by server 200b.

[0056] Client computers 200a that desire to use the data stored by server computer 200b can access the database 270 via communications network 280. Client computers 200a access the data by way of, for example, a query, a form, etc. It will be appreciated that any configuration of computers is equally compatible with an embodiment of the present invention.

What is claimed is:

1. A method for use in a computing system, the computing system comprising a processing device and a remote input device, the remote input device being operable over a plurality of sensitivity ranges, the method comprising:

determining, at the processing device, a desired sensitivity range for the remote input device; and

signaling by the processing device to the remote input device to operate in the desired sensitivity range.

2. The method of claim 1, wherein the computing system comprises a game playing system, and wherein determining the desired sensitivity range is based at least in part on a current context in an application executing on the game playing system.

3. The method of claim 1, wherein the computing system comprises a game playing system, and wherein determining the desired sensitivity range is based at least in part on a user profile.

4. The method of claim 1, wherein the remote input device comprises a motion sensor that is operable over a plurality of selectable sensitivity ranges, and wherein signaling by the processing device to the remote input device comprises signaling by the processing device to the remote input device to cause the motion sensor to operate in a sensitivity range selected from the plurality of sensitivity ranges.

5. The method of claim 1, wherein the remote input device comprises a plurality of motion sensors, each said motion sensor being operation in at least one sensitivity range, and wherein signaling by the processing device to the remote input device comprises signaling by the processing device to operate a sensor selected from the plurality of sensors.

6. The method of claim 1, wherein the remote input device comprises at least one of a gyro, an accelerometer, or a magnetometer.

7. The method of claim 1, wherein the remote input device is wirelessly coupled to the processing device.

8. A system, comprising:

a context-determining module configured to ascertain a current context in an application executing on a computing device;

a sensitivity-determining module configured to receive information from the context-determining module, and to determine a desired sensitivity range for a remote input device, wherein the remote input device is operationally coupled to provide input to the computing device, and wherein the remote input device is operable over a plurality of sensitivity ranges; and

a communications module configured to communicate information indicative of the desired sensitivity range to the remote input device for use in selecting a sensitivity range for the remote input device.

9. The system of claim 8, wherein the remote input device is wirelessly coupled to the computing device.

10. The system of claim 8, wherein the application executing on the computing device is a game-playing application, and the context-determining module is configured to ascertain a current scripted situation within the game playing application.

11. The system of claim 8, wherein the context-determining module is configured to ascertain a user profile, and the sensitivity-determining module is configured to determine the desired sensitivity range based at least in part on the user profile.

12. The system of claim 8, wherein the remote input device comprises a physical motion sensor having a plurality of sensitivity ranges, and is configured to respond to communicated information indicative of the desired sensitivity range by operating in the desired sensitivity range.

13. The system of claim 8, wherein the remote input device comprises a plurality of physical motion sensors, each said motion sensor having at least one sensitivity range, and wherein the remote input device is configured to respond to communicated information indicative of the desired sensitivity range by activating at least one of the physical motion sensors, the at least one physical motion sensor being operable in a sensitivity range corresponding to the desired sensitivity range.

14. The system of claim 8, wherein the remote input device comprises at least one of a gyro, an accelerometer, or a magnetometer.

15. A computer-implemented game-playing system, comprising:

a remote input device comprising a physical motion sensor that is operable over a plurality of sensitivity ranges; and

a processing device that ascertains a current scripted situation within a game playing application executing on the processing device, determines a desired sensitivity range for the remote input device based at least in part on the current scripted situation, and communicates information indicative of the desired sensitivity range to the remote input device,

wherein the remote input device receives from the processing device the communicated information indicative of the desired sensitivity range, and responds to receiving the communicated information by causing the physical motion sensor to operate in the desired sensitivity range.

16. The system of claim 15, wherein the processing device determines the desired sensitivity range based at least in part on a user profile.

17. The system of claim 15, wherein the physical motion sensor is operable over a plurality of selectable sensitivity ranges, and wherein the remote input device responds to receiving the communicated information by causing the physical motion sensor to operate in a selected one of the plurality of sensitivity ranges.

18. The system of claim 15, wherein the remote input device comprises a plurality of physical motion sensors, each said physical motion sensor operable in at least one respective sensitivity range, and wherein the remote input device responds to receiving the communicated information by causing a selected one of the plurality of physical motion sensors to operate in the desired sensitivity range.

19. The system of claim 15, wherein the physical motion sensor includes at least one of a gyro, an accelerometer, or a magnetometer.

20. The system of claim 15, wherein the remote input device is wirelessly coupled to the computing device.

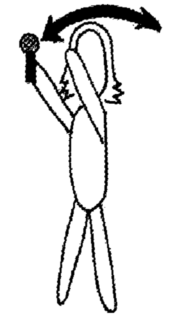


FIG. 1

20 sec tilt mode

FIG. 2A

FIG. 2B

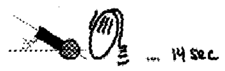


FIG. 2C

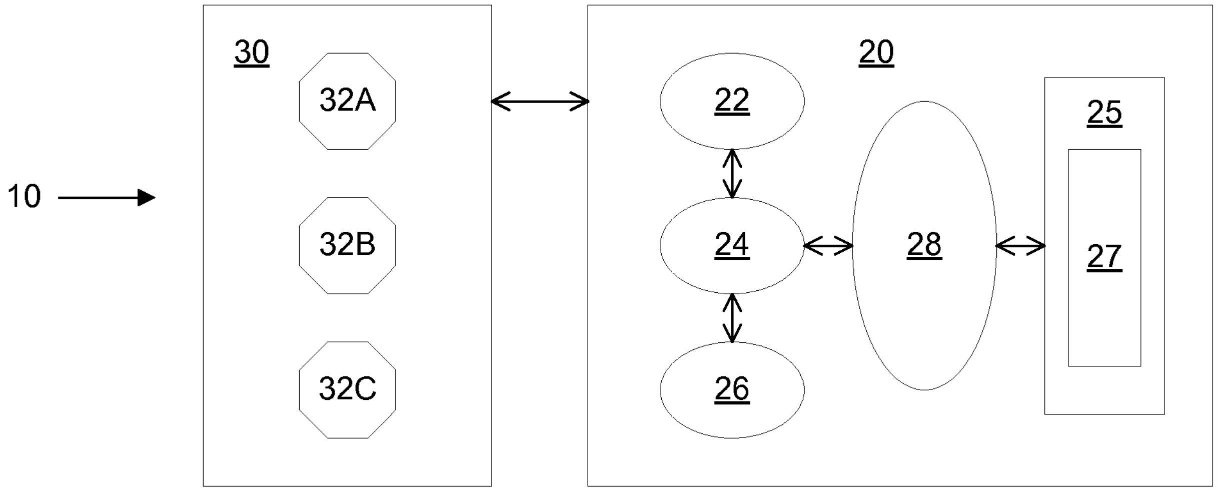


FIG. 3

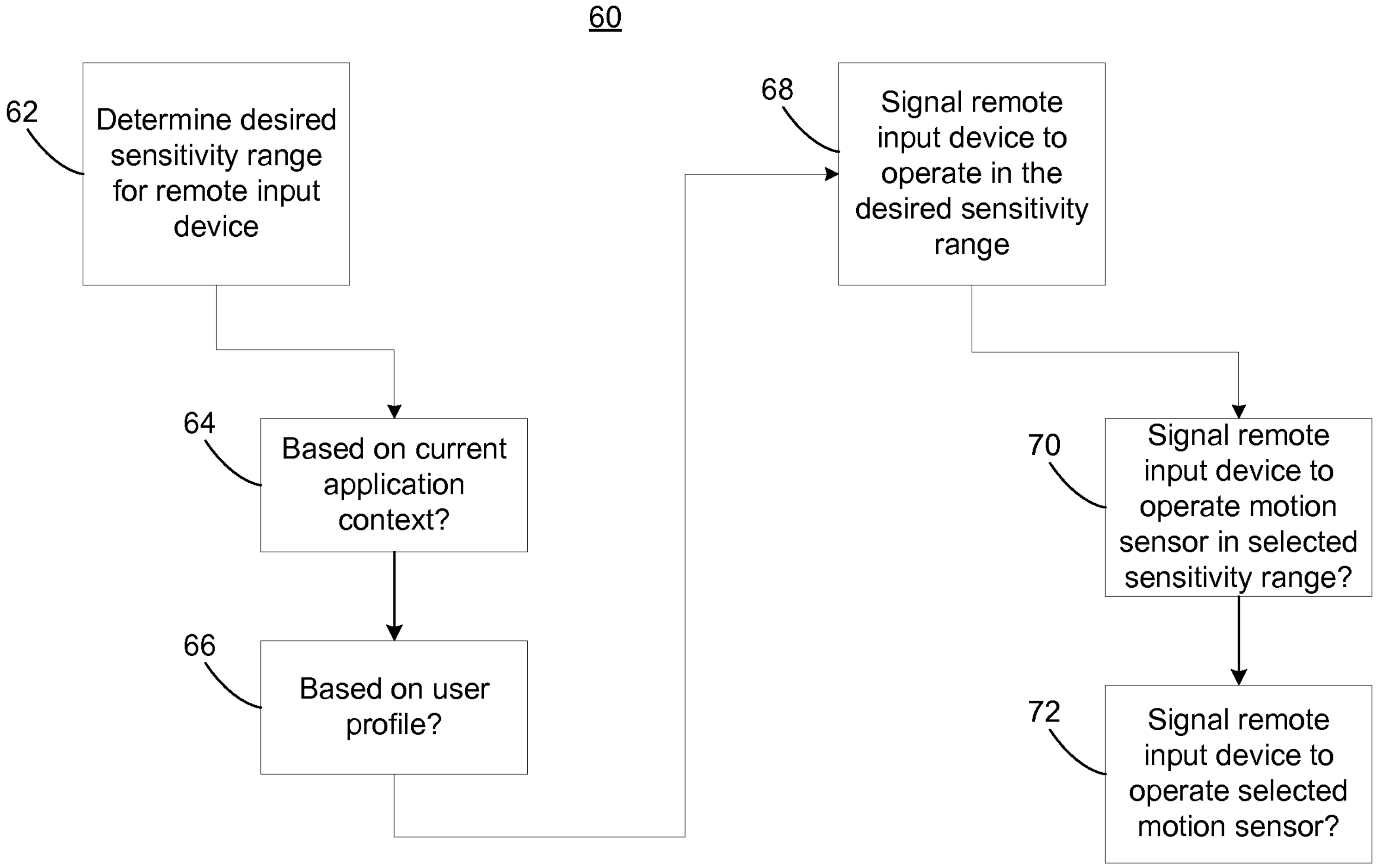
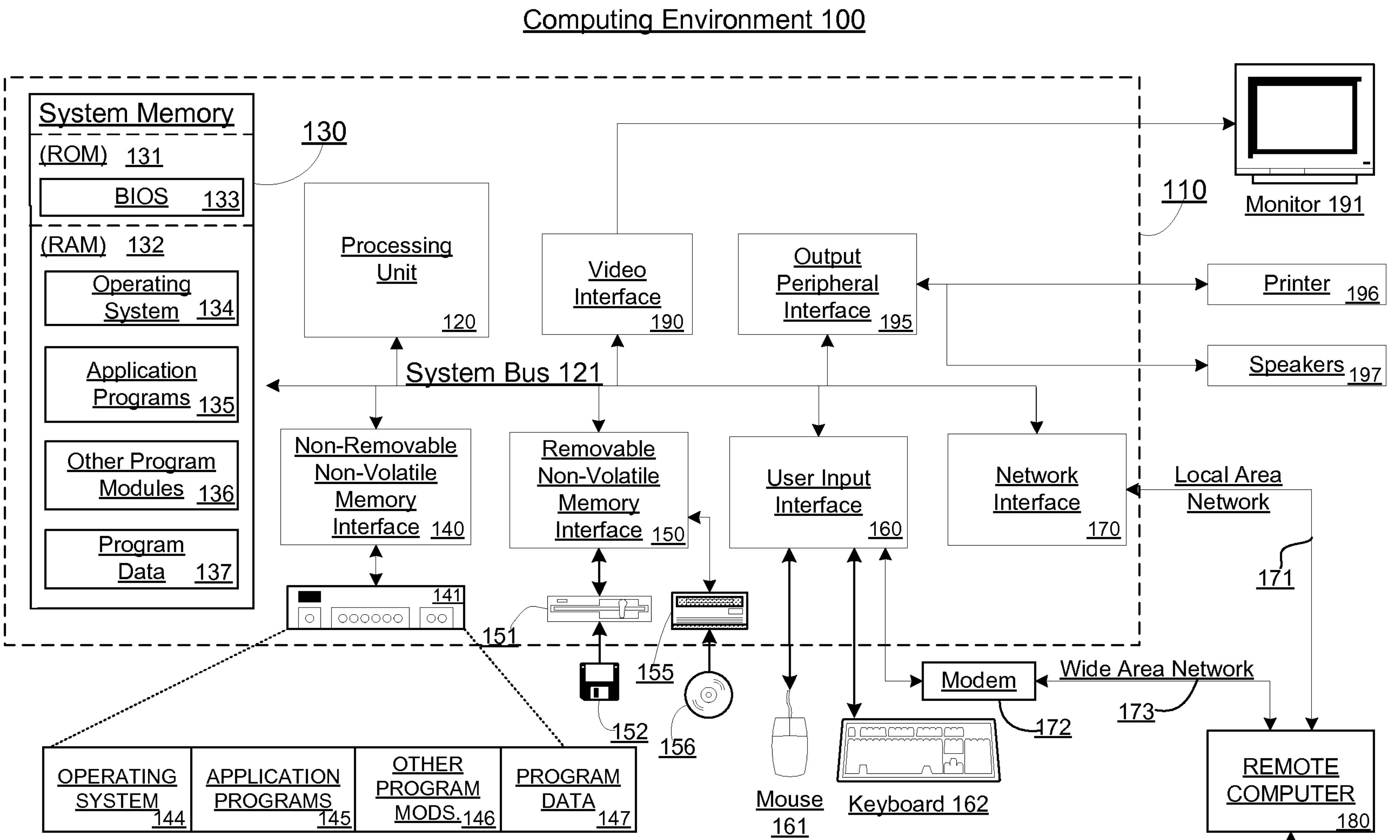

60
62
Determine desired sensitivity range for remote input device
64
Based on current application context?
66
Based on user profile?
68
Signal remote input device to operate in the desired sensitivity range
70
Signal remote input device to operate motion sensor in selected sensitivity range?
72
Signal remote input device to operate selected motion sensor?

FIG. 4


Computing Environment 100
System Memory
130
(ROM) 131
BIOS 133
(RAM) 132
Operating System 134
Application Programs 135
Other Program Modules 136
Program Data 137
Processing Unit 120
Video Interface 190
Output Peripheral Interface 195
110
Monitor 191
Printer 196
Speakers 197
System Bus 121
Non-Removable Non-Volatile Memory Interface 140
Removable Non-Volatile Memory Interface 150
User Input Interface 160
Network Interface 170
Local Area Network
171
141
151
155
152
156
Modem
172
Wide Area Network
173
Mouse 161
Keyboard 162
REMOTE COMPUTER 180
OPERATING SYSTEM 144
APPLICATION PROGRAMS 145
OTHER PROGRAM MODS. 146
PROGRAM DATA 147

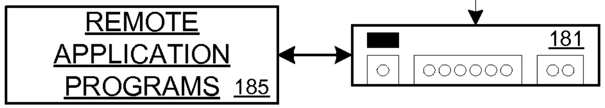
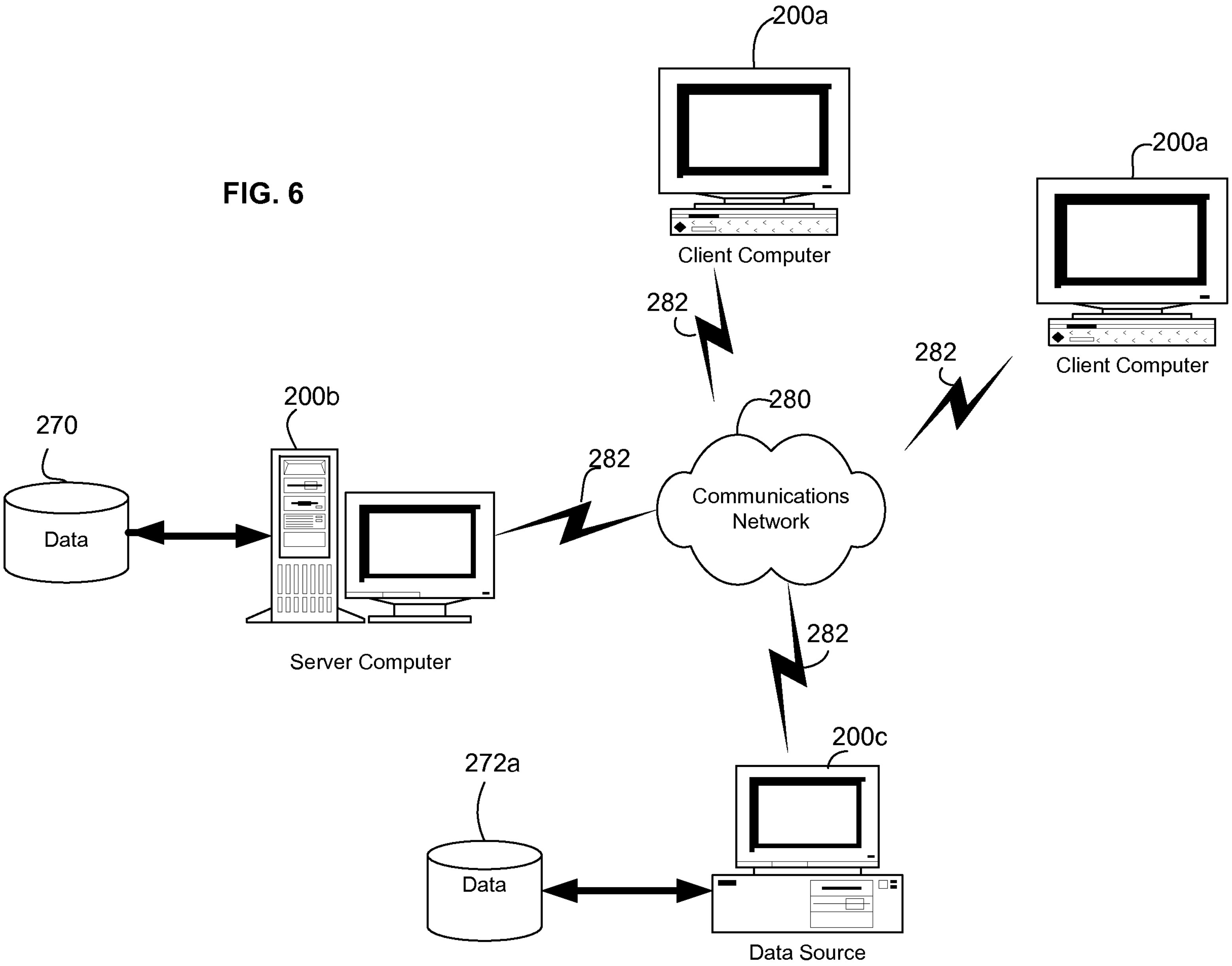

REMOTE APPLICATION PROGRAMS 185
181

FIG. 5

FIG. 6


200a
Client Computer
200a
Client Computer
282
282
280
Communications Network
270
Data
200b
Server Computer
282
282
200c
272a
Data
Data Source