



(86) Date de dépôt PCT/PCT Filing Date: 2004/09/17  
(87) Date publication PCT/PCT Publication Date: 2005/03/31  
(85) Entrée phase nationale/National Entry: 2006/03/17  
(86) N° demande PCT/PCT Application No.: US 2004/030441  
(87) N° publication PCT/PCT Publication No.: 2005/028054  
(30) Priorité/Priority: 2003/09/18 (US60/504,566)

(51) Cl.Int./Int.Cl. *A63F 13/00* (2006.01),  
*A63F 13/12* (2006.01)  
(71) Demandeur/Applicant:  
TURBINE ENTERTAINMENT SOFTWARE CORP., US  
(72) Inventeurs/Inventors:  
HOGAN, MIKE, US;  
AUSTREW, ERIC, US;  
DUPRAS, DEREK, US;  
CUNNINGHAM, SEAN, US  
(74) Agent: SMART & BIGGAR

(54) Titre : LE SYSTEME EST PROCÉDE DE COMMANDES D'ACCES A UN JEU DE ROLES EN LIGNE  
MASSIVEMENT MULTIJOUEUR  
(54) Title: SYSTEM AND METHOD FOR CONTROLLING ACCESS TO A MASSIVELY MULTIPLAYER ON-LINE ROLE-  
PLAYING GAME

(57) **Abrégé/Abstract:**

A method for controlling access to a massively multi-player, online role-playing game begins by presenting a web page to a first user. The web page receives a parameter for controlling access of a second user to a massively multi-player, online role-playing game. The received parameter is transmitted to a database server, which stores the received parameter in a database record associated with the second user. A game server receives a connection request from the second user and controls access of the second user to the massively multi-player, online role-playing game using the stored parameter.



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

(19) World Intellectual Property  
Organization  
International Bureau



(43) International Publication Date  
31 March 2005 (31.03.2005)

PCT

(10) International Publication Number  
**WO 2005/028054 A1**

- (51) International Patent Classification<sup>7</sup>: A63F 13/00, 13/12
- (21) International Application Number: PCT/US2004/030441
- (22) International Filing Date: 17 September 2004 (17.09.2004)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data: 60/504,566 18 September 2003 (18.09.2003) US
- (71) Applicant (for all designated States except US): **TURBINE ENTERTAINMENT SOFTWARE CORP.** [US/US]; 60 Glacier Drive, Suite 4000, Westwood, MA 02090 (US).
- (72) Inventors; and
- (75) Inventors/Applicants (for US only): **HOGAN, Mike** [US/US]; 67 Cemetery Road, Charlton, MA 01507 (US). **AUSTREW, Eric** [US/US]; 155 Strathmore Rd., Apt 14, Brighton, MA 02135 (US). **DUPRAS, Derek** [US/US]; 130 Mason Road, Whitinsville, MA 01588 (US). **CUNNINGHAM, Sean** [US/US]; 811 Centre Street, Apt. 5, Jamaica Plain, MA 02130 (US).
- (74) Agents: **LANZA, John, D.** et al.; Lahive & Cockfield, LLP, 28 State Street, Boston, MA 02109 (US).
- (81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, 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, HU, IE, IT, LU, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).
- Published:**  
— with international search report
- For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: SYSTEM AND METHOD FOR CONTROLLING ACCESS TO A MASSIVELY MULTIPLAYER ON-LINE ROLE-PLAYING GAME

(57) Abstract: A method for controlling access to a massively multi-player, online role-playing game begins by presenting a web page to a first user. The web page receives a parameter for controlling access of a second user to a massively multi-player, online role-playing game. The received parameter is transmitted to a database server, which stores the received parameter in a database record associated with the second user. A game server receives a connection request from the second user and controls access of the second user to the massively multi-player, online role-playing game using the stored parameter.

WO 2005/028054 A1



## SYSTEM AND METHOD FOR CONTROLLING ACCESS TO A MASSIVELY MULTIPLAYER ON-LINE ROLE-PLAYING GAME

### FIELD OF INVENTION

[001] This invention relates generally to multiplayer computer games, and in particular, to methods for limiting access to a massively multiplayer on-line role-playing game.

### BACKGROUND

[002] A massively multiplayer game (“MMP”) is a computer game played by a large number of users through a communications network, which can be a local area network (e.g., Ethernet), a medium-area network (e.g., an intranet), or a wide-area network (e.g., the Internet). In addition, the communications network can be a wireless network, a cellular network or any other system which facilitates the transmission of data. In MMPs, humans and their avatars within the game (“players”) are free to interact with other players as well as autonomous “non-player characters” which inhabit and are part of the game. Early examples of MMPs include games such as Ultima Online, EverQuest, and CrossGate.

[003] Most MMPs are fantasy role-playing games (“RPGs”) which take place within a mythical or mystical world. Most MMPs appear timeless, in that from the player’s perspective they have no beginning and no end. New players can join a game in progress at any time, and do not need to wait for the start of a new game. In addition, many MMPs do not even define an absolute game endpoint, making the end of a game a logical impossibility. Thus, once a game has started, it can continue indefinitely. The players of that MMP are thereby involved in a continuing storyline akin to life within the real world. MMPs allow players to develop their avatars, form personal relationships with other players, and to enjoy social interaction through the reality of the game.

[004] Unfortunately, the ongoing, immersive nature of massively multiplayer on-line role playing games (MMORPGs) causes many players to spend large amounts of time engaged in active game play. The reasons for this are manifold. Some players may fear missing an important development in the game. Others may play

constantly in order to further develop their character and develop their characters more rapidly than other players of the game.

[005] A second problem associated with MMORPGs is that the “world” in which play take place is a virtual one that is not easily monitored by supervising individuals. For example, in the case of minors, a parent or guardian has no way of knowing when or for how long the player has played the game, nor can a parent or guardian ensure that the minor does not engage in objectionable behavior while playing the game.

[006] Accordingly, new methods are needed to control access of players to MMORPGs and to monitor their usage of the game.

#### SUMMARY OF THE INVENTION

[007] The present invention provides systems and methods for controlling access to a MMORPG that can be configured to control access at certain times or for certain periods of time. It also provides a convenient way for a supervising user to review the behavior of a player without requiring the supervising user to continually monitor the action of the player. In some embodiments, the systems and methods disclosed may be used by a player of the MMORPG to self-regulate involvement in the game.

[008] In one aspect, the present invention relates to a method for controlling access to a MMORPG. A web page or other suitable interface, such as an in-game or mobile device interface, may be presented to a supervising user. The supervising user provides one or more parameters for controlling the access of a player to the game. The parameters may be transmitted via an intermediary system or server. The received parameter is transmitted to a database where it is stored in a database record associated with the player. When a connection request is received at a game server, the parameters stored in the database control the player's access to the game.

[009] In another aspect, the present invention relates to an article of manufacture embodying computer-readable program means for controlling access to a MMORPG. The computer-readable program means presents a web page or other suitable interface, such as an in-game or mobile device interface, to a supervising user. The computer-readable program means receives one or more parameters for controlling the access of a player to the game and transmits the one or more parameters to a database for storage in a database record associated with the player. The parameters



may be transmitted via an intermediary system or server. When a connection request is received at a game server, the computer-readable program means retrieves parameters stored in the database and controls the player's access to the game responsive to the retrieved parameters. The connection requests may be intercepted by an intermediary system or server before they reach the game server.

**[0010]** In still another aspect, the present invention relates to a system for controlling access to a massively multi-player, online role-playing game. The system includes a web server, a database server in communication with the web server, and a game server in communication with both the web server and the database server. In some embodiments other intermediary systems may be provided. The web server presents a user interface to a supervising user and receives from the supervising user a parameter for controlling access to the game of a player. In some embodiments, other suitable interfaces, such as an in-game or mobile device interface, are used to help control access. The database server receives from the web server the parameter for controlling access of the player to the game and stores the parameter in a database record associated with the player. In other embodiments, the web server communicates with the database server through an intermediary server. The game server receives a connection request from the player and prohibits access to the game responsive to data received from the database server.

#### BRIEF DESCRIPTION OF THE DRAWINGS

**[0011]** These and other aspects of this invention will be readily apparent from the detailed description below and the appended drawings, which are meant to illustrate and not to limit the invention, and in which:

**[0012]** FIG. 1 is a block diagram showing a distributed computer system in accordance with one embodiment of the present invention;

**[0013]** FIG. 2 is a block diagram of an embodiment of a computer useful in connection with the present invention;

**[0014]** FIG. 3 is a block diagram showing one embodiment of a distributed system in which a computer is used to set parameters controlling access to the game;

**[0015]** FIGs. 4-5D are screenshots showing an embodiment of a screen used to collect parameter data for controlling access to the game; and

[0016] FIG. 6 is a flowchart depicting one embodiment of the steps taken by a game server utilizing gameplay parameters supplied by a database server.

#### DETAILED DESCRIPTION

[0017] While the content and nature of MMPs is the key to their widespread success, it is important to understand the technological underpinnings of a typical MMP. Although it is possible to host and play a computer game, and even an MMP, on a single computer, it is not preferred for MMPs. FIG. 1 illustrates a distributed computer system 100 in accordance with one embodiment of the present invention. The system 100 includes a server platform 102 and one or more client systems 112, 114, 116, 118 communicating via a communications network 110. In the embodiment depicted in FIG. 1, the server platform 102 includes a plurality of individual servers 104, 106, 108. In other embodiments the server platform 102 may comprise a single server. The number of clients is virtually limitless, constrained only by the physical characteristics of the server platform 102, and the communications network 110 connecting the two. The topology of the network 110 over which the client nodes 112, 114, 116, 118 communicate with the server platform 102 may be a bus, star, or ring topology. The network 110 can be a local area network (LAN) such as Ethernet, a metropolitan area network (MAN) such as an intranet, or a wide area network (WAN) such as the Internet. In other embodiments, the communications network 110 could include, without limitation, a wireless network, a cellular network or any other system which facilitates the transmission of data. Each client 112, 114, 116, 118 has an associated communications link (or session) with one or more of the servers 104, 106, 108. As shown in Figure 1, client 1 112 could communicate with server A 104 via communications link 122. Similarly, client 2 114 could communicate with server B 106 via communications link 124. As shown in FIG. 1, the communications network 110 may be a dedicated network or a shared network such as the Internet. As will be appreciated, the system 100 is a distributed virtual environment tailored to facilitate MMPs.

[0018] The client systems 112, 114, 116, 118 and servers 104, 106, 108 can connect to the network 110 through a variety of connections including standard telephone lines, LAN or WAN links (e.g., T1, T3, 56 kb, X.25), broadband



connections (ISDN, Frame Relay, ATM), and wireless connections. Connections can be established using a variety of communication protocols (e.g., TCP/IP, IPX, SPX, NetBIOS, Ethernet, ARCNET, Fiber Distributed Data Interface (FDDI), RS232, IEEE 802.11, IEEE 802.11a, and direct asynchronous connections). Other client nodes and server nodes (not shown) may also be connected to the network 110. In other embodiments, client systems may connect to servers using separate, distinct networks.

**[0019]** Each client 112, 114, 116, 118 is generally responsible for displaying interacting objects (other players, terrain, non-player characters, etc.), displaying the game's interface, processing a player's inputs, playing music and sounds, and performing other CPU or bandwidth intensive operations. The client nodes 112, 114, 116, 118 can be any device capable of displaying video, audio or tactile game content and receiving user input for game interaction. Client nodes 112, 114, 116, 118 may be personal computers, windows-based terminals, network computers, information appliances, palmtop computers (such as the Series 7 by Psion PLC), X-devices, workstations, mini computers, personal digital assistants (such as the Tungsten C by Palm Inc.), portable digital handheld game systems (such as the Gameboy Advance by Nintendo of America Inc.), game consoles (such as the Play Station 2 by Sony Corporation of America), or cellular telephones (such as the Motorola Inc. A388c).

**[0020]** For embodiments in which the client 112, 114, 116, 118 is a personal computer, the client may be constructed around the Pentium family of processors manufactured by Intel Corp. of Mountain View, California, which includes the Pentium, Pentium II XEON, Celeron, and Pentium III microprocessors, the Power PC line of processors manufactured by Motorola Corp. of Schaumburg, Illinois, or the Crusoe line of processors manufactured by TransMeta Corp. of Santa Clara, California. In these embodiments, the client nodes 112, 114, 116, 118 can operate under the control of a variety of operating systems including, but not limited to, WINDOWS 3.x, WINDOWS 95, WINDOWS 98, WINDOWS 2000, WINDOWS NT 3.51, WINDOWS NT 4.0, WINDOWS CE, and WINDOWS XP, all of which are manufactured by Microsoft Corporation of Redmond, Washington, MacOS, manufactured by Apple Computer of Cupertino, California, Java, UNIX, or Linux.



[0021] In other embodiments, the client 112, 114, 116, 118 may have different processors, operating systems, and input devices consistent with the device. For example, in one embodiment the client node is a Zire 71 personal digital assistant manufactured by Palm, Inc. of Santa Clara, California. In this embodiment, the Zire 71 operates under the control of the PalmOS operating system and includes a stylus input device as well as a five-way navigator device.

[0022] Each of the servers 104, 106, 108 generally coordinates communication, database storage, and overall control and administration of the game. The servers 104, 106, 108 generally maintain state information and coordinate client interaction with various objects in a virtual environment, including but not limited to other clients, vehicles, artificial intelligence, terrain, music and sound. Each server 104, 106, 108 provides additional functions, such as security, recording game goals and scoring and tracking each player's advancement towards those goals. The server nodes 104, 106, 108 can be any computing device that stores files representing game elements and capable of receiving and processing game input. The servers 104, 106, 108 can be provided as a group of server systems logically acting as a single server system, referred to herein as a server farm. In one embodiment, the servers 104, 106, 108 are a multi-user server system supporting multiple concurrently active client connections. Servers 104, 106, 108 may be computers based on the Pentium family of processors manufactured by Intel Corp. of Mountain View, California, which includes the Pentium, Pentium II XEON, Celeron, and Pentium III microprocessors, the Power PC line of processors manufactured by Motorola Corp. of Schaumburg, Illinois, or the Crusoe line of processors manufactured by TransMeta Corp. of Santa Clara, California. The servers 104, 106, 108 may operate under the control of a variety of operating systems including, but not limited to, WINDOWS 3.x, WINDOWS 95, WINDOWS 98, WINDOWS 2000, WINDOWS NT 3.51, WINDOWS NT 4.0, WINDOWS CE, and WINDOWS XP, all of which are manufactured by Microsoft Corporation of Redmond, Washington, MacOS, manufactured by Apple Computer of Cupertino, California, Java, UNIX, or Linux.

[0023] FIG. 2 depicts a block diagram of a typical computer 200 that may be used as a client 112, 114, 116, 118 or server 104, 106, 108. As shown in FIG. 2, the computer 200 includes a central processor 210, a main memory unit 220 for temporarily storing programs and/or data, an input/output (I/O) controller 230, a



display device 240, and a data bus 250 coupling these components to allow them to communication with each other. The data bus 250 may be a VESA bus, a VESA VL bus, an ISA bus, an EISA bus, a PCI bus, a NuBus, or a MicroChannel Architecture bus. In some embodiments, the display device 240 communicates with the data bus 250 via a video card (not shown), such as the Radeon 7000 Mac Edition PCI video card, manufactured by ATI Technologies of Santa Clara, California. The main memory unit 220 may include both random access memory (RAM) and read only memory (ROM) chips. The computer 200 typically also has one or more input/output devices, such as a keyboard 260, a mouse 265, or a game controller 270 connected to the I/O controller 230. In some embodiments, the computer 200 includes dedicated circuitry for producing sound, such as a "sound card" (not shown). In some of these embodiments the sound card is controlled by the I/O controller 230. In other of these embodiments, the sound card interfaces directly to system bus 250.

**[0024]** As shown in FIG. 2, the computer 200 typically also has a hard disk drive 280. The computer may optionally provide other storage devices (not shown in FIG. 2) such as a floppy disk drive for receiving floppy disks such as 3.5-inch or 5.25-inch disks, a CD-ROM drive, a CD-R drive, a CD-RW drive, a DVD-ROM drive, or tape drives of various formats. In still other embodiments, the computer 200 may provide USB connections to receive handheld USB storage devices such as the USB Flash Drive line of devices manufactured by Twintech Industry, Inc. of Los Alamitos, California.

**[0025]** Referring now to FIG. 3, an administration client 300 is used to connect to an administration server 310 via a second communications network 110'. In some embodiments, the administration client 300 may be one of the clients 112, 114, 116, 118 used for enjoyment of the game. In still other embodiments, the administration client 300 may be a personal computer, a windows-based terminal, network computer, information appliance, palmtop computer, X-device, workstation, mini computer, personal digital assistant, portable digital handheld game, or game console not typically used for enjoying the game, or it may be a cellular telephone or line-based telephone. The administration server 310 may be one of the servers included in the server platform 102. In other embodiments, such as the one shown in FIG. 3, the administration server 310 is separate from, but in communication with, the server platform 102 via a third communications network 110". The administration server



310 includes a database, shown as 320 in FIG. 3. The database 320 may be a relational database, such as MicroStrategy 7i, manufactured by MicroStrategy, Inc. of McLean, Virginia or Transformation Server, manufactured by DataMirror Corporation of Toronto, Canada. In other embodiments, the database 320 may be provided as a flat-file database, such as JET, manufactured by Microsoft Corporation of Redmond, Washington.

[0026] The second communications network 110' may be a different portion of the same public network that the clients 112, 114, 116, 118 use to connect to the server platform 102. In other embodiments, the second communications network 110' is physically separate from communications network 110. For embodiments in which the administration client 300 is a personal computer, a windows-based terminal, network computer, information appliance, palmtop computer, X-device, workstation, mini computer, personal digital assistant, portable digital handheld game, or game console, the communications network 110' may be provided as a packet-switched network such as the Internet. In some of these embodiments the administration client 300 may communicate with the server platform 102 the Point-to-Point (PPP) communications protocol or the Serial Line Internet Protocol (SLIP).

[0027] For embodiments in which the administration client 300 is a cellular telephone it may communicate with the server platform via the Wireless Access Protocol (WAP) or i-Mode. For embodiments in which the administration client 300 is a telephone, the communication network 100' may use plain old telephone service (POTS).

[0028] The third communications network 110'' may be a different portion of the same public network that the clients 112, 114, 116, 118 use to connect to the server platform 102. In other embodiments, the third communications network 110'' is physically separate from communications network 110. In other embodiments, the third communications network 110'' is part of the second communications network 110', both of which are separate from communications network 100. In still other embodiments, the third communications network 110'', the second communications network 110', and the communications network 110 are all part of the same network.

[0029] FIG. 4 depicts one embodiment of a screen shot presented to a user of the administration client 300 (referred to as a "supervising user") upon connection and proper authentication to the administration server 310. Although shown in FIG. 4 as



a separate client, the administration client can be any client system 112, 114, 116, 118 used by a properly authenticated user. The supervising user may authenticate herself to the administration server 310 using any one of a number of well-known techniques including, but not limited to, passwords, passphrases, certificates, token-based techniques, public key techniques, or biometric techniques.

[0030] FIG. 4 depicts a screen shot embodiment in which the administration client 300 communicates with the administration server 310 with the aid of a browser program, such as INTERNET EXPLORER, manufactured by Microsoft Corporation of Redmond, Washington or SAFARI, manufactured by Apple Computer, of Cupertino, California. In other embodiments, the administration client 300 may not require a browser program to communicate with the administration server 310. For example, the administration server 310 may execute a traditional client-server program in which the application executes on the administration server 310 and application output is transmitted to the administration client 300 using a presentation-level protocol such as X-Windows. In other embodiments, the administration client 300 is an application program that is specifically written to execute on the client system 112, 114, 116, 118. In still other embodiments, as noted above, the administration client 300 may be a cellular telephone or standard telephone. In these embodiments, the administration server 310 may communicate with the administration client 300 using a voice menu instead of a screen of visual data.

[0031] Once the supervising user has authenticated herself, selected the game-playing user for which she wishes to enter account parameters, and the administration server 310 has verified that the supervising user has the right to change the account parameters for that particular game-playing user, the administration server 310 provides a number of dialog boxes for setting parameters associated with the game-playing user's account. In the embodiment shown in FIG. 4, five dialog boxes are present: a Game Time-Out dialog 410, a Parental Notifications dialog 420, a History dialog 430, a Chat Logging dialog 440, and a Restrictions dialog 450. The dialog boxes 410, 420, 430, 440, 450 can be implemented as HTML code within an HTML page describing the screen 400. Alternatively, each dialog box may be implemented as an ActiveX control or JAVA



applet. In still other embodiments the dialog boxes 410, 420, 430, 440, 450 may be created by application code specifically written to execute on the target client device.

[0032] The Game Time-Out dialog 410 allows a user to specify the maximum length of time for a single game session. A user clicks the Time-Out button 412 to specify parameters for this control. FIG. 5A depicts an embodiment of a page 510 delivered to the administration client 300 when the Time-Out button 412 is clicked. In the embodiment shown in FIG. 5A, the page includes a pull-down menu 512 allowing one of multiple time-out lengths to be selected. In other embodiments, the time-out length may be selected using radio buttons. Alternatively, the time-out length may be provided by the supervising user in a blank field. Upon selection or entry of a time-out length, the time-out length is transmitted to the administration server 310, which stores it in a database record corresponding to the player's account.

[0033] In other embodiments, the page 510 may include a control for immediately logging a game-playing user out of the game. In these embodiments, activation of that control causes a message to be sent to the server platform 102 to terminate the session associated with the identified player. In still other embodiments, the page 510 may include a control for locking the player's account. In these embodiments, a locked account prevents a player from accessing the game until the account is unlocked.

[0034] Referring again to FIG. 4, the Parental Notifications dialog 420 displays information to a supervising user relating to the behavior of the identified player. In the embodiment shown in FIG. 4, the Parental Notifications dialog 420 includes summary information 422 regarding the most recent behavior events that have occurred. Although the embodiment shown in FIG. 4 relates to negative behavioral events, positive behavior events, such as achievement of goals in the game, may also be displayed. The depicted embodiment of the Parental Notifications dialog 420 also includes a "View All" button 424. Clicking the "View All" button 424 causes a message to be sent to the administration server 310. In response to the message, the administration server 310 queries the database 320 for a complete list of behavior events for the specified player. In HTML embodiments, the administration server 310 responds with an HTML page of the sort depicted in FIG. 5B. As shown in FIG. 5B, the Parental Notifications page 520 includes additional information regarding the



behavior event 522, the start time of any discipline imposed 524, the end time of any discipline imposed 526, whether the player appealed the discipline 528 and the status of the appeal 529. As shown in FIG. 5B, multiple Parental Notification pages 520 may be required.

**[0035]** In other embodiments, the Parental Notifications dialog box 420 may include a control allowing the supervising user to request a copy of each parental control notification by email or SMS message as it occurs. In still other embodiments, the Parental Notifications dialog box 420 may include a field for entering a limit on the number of negative parental notifications that may be received. In these embodiments, when the number of negative parental notifications exceeds the entered limit, the player's account is locked. In still others of these embodiments, the supervising user may specify a time period during which, if the number of negative parental controls received exceeds the limit, the player's account is locked.

**[0036]** The Gameplay History dialog box 430 provides information to the supervising user concerning gameplay statistics for the player's account. In the depicted embodiment, the history for a single day is listed in detail box 432. As shown in the depicted embodiment, the detail box includes the date, the total amount of time spent playing the game by the player, and the earliest and latest time during the day that the player was active. The embodiment depicted in FIG. 4 includes a check box for requesting weekly summaries of the player's activity. The requested summaries may be sent to the supervising user by email or as an SMS message. The depicted embodiment also includes a "View More" button 436. Clicking the "View More" button 436 causes a message to be sent to the administration server 310. In response to the transmitted message, the administration server 310 queries the database 320 to compile statistics concerning the gameplay history of the specified user. In HTML embodiment, an HTML file of the sort depicted in FIG. 5C is displayed to the supervising user.

**[0037]** The embodiment of a Gameplay History screen 530 depicted in FIG. 5C includes a pull-down menu 532 allowing the supervising user to specify the number of days for which gameplay statistics are requested. In other embodiments, the history value may be selected using radio buttons. Alternatively, history value may be provided by the supervising user in a blank field. Upon selection or entry of a



history value, it is transmitted to the administration server 310, which queries the database 320 to retrieve the appropriate number of values from concerning the specified user's gameplay history. Also included in the embodiment of a Gameplay History page 530 as shown in FIG. 5C is a table 534 that includes the total number of hours spent playing the game and the last login time for each day.

**[0038]** A well-known feature of online games is the ability to "chat" with other players of the game. A player types text into a text field that is transmitted to another player. The transmitted text appears on the other player's screen. The Chat Logging dialog 440 enables or disables storage of "chats." When a supervising user clicks the "On" button 442 in the Chat Logging dialog box 440, that selection is transmitted to the administration server 310, which stores the parameter in a database record corresponding to the game-playing user's account.

**[0039]** The Restrictions dialog 450 allows a supervising user to set parameters controlling specific times and durations for which a player will be allowed to play the game. In the embodiment shown in FIG. 4, a supervising user may set the amount of time per week a player may play the game or may control the specific times during the week that a player may play. Clicking the "Edit" button 452 causes a message to be sent to the administration server 310. In HTML embodiments, the administration server 310 responds with an HTML page 550 of the sort depicted in FIG. 5D. In the embodiment shown in FIG. 5D, a "Game Time Calendar" page 550 is displayed to the supervising user. The page 550 includes a pull-down menu 552 allowing a supervising user to select the amount of time per week a player may play the game. The page 550 also includes a calendar that allows a supervising user to specify blocks of time during which a player may play the game and blocks of time during which a player may not play the game. In some embodiments, selection of a block of time is accomplished by clicking on the rectangle representing the time period. For example, in the embodiment depicted in FIG. 5D, a supervising user may enable playing Sunday nights from 9:00 PM to 10:00 PM by clicking rectangle 558. Similarly, in the depicted embodiment, a supervising user may disable playing Saturdays from 11:00 PM to midnight by clicking rectangle 560.

**[0040]** The embodiment depicted in FIG. 5D also includes a "Weekdays" column 562 and a "Weekends" column 564. These columns allow a playing to be enabled or disabled uniformly across Monday through Friday or Saturday and



Sunday, respectively. Once a supervising user has specified a total amount of time game play is allowed and specific time periods during which game play is allowed, the supervising user clicks the "Submit" button 556 to transfer the parameters to the administration server 510. The administration server 310 stores the transmitted parameters in a database record associated with the player.

[0041] Referring now to FIG. 6, a flowchart showing one embodiment of the steps taken by the server platform 102 when a user logs into the game is shown. The user authenticates herself to the server platform 102 (step 602). Once authenticated, the server platform 102 requests gameplay parameters from the administration server 320 (step 604). Upon receipt of the player's gameplay parameters, the server platform 102 determines whether the user is prohibited from logging into the game by comparing the current date and time to information entered via the Gameplay Calendar 562 (step 606). In other embodiments, the server platform 102 checks whether the user has been explicitly banned by game administrators or by supervising users. If play is not allowed at the current date and time the player's session is terminated (step 608). Otherwise, the server platform 102 checks the total time played by the player during this period and compares it to the limit entered in the Restrictions pull-down menu 552 (step 610). If the player has exceeded her time allotment for the current period, the server platform 102 terminates her session (step 608). Otherwise, the server platform 102 checks the player's parameters to determine if chat logging has been enabled (step 612). If so, the server platform 102 transmits a message to the player's client to log all chat activity (step 614) and play begins. Otherwise, play begins without chat logging. The server platform 102 activates a timer during gameplay to track the amount of time played by the user. During gameplay the user may voluntarily logout (step 618). During active gameplay, the server platform 102 periodically checks the amount of time spent by the user to determine if the user has exceeded her allotted time or to determine if the user has entered a time period for which play is prohibited (step 620). If this occurs, the server platform 102 terminates the player's game session.

[0042] Still referring to FIG. 6 and in more detail, the player authenticates herself to the server platform 102 (step 602). The player may authenticate herself to the server platform 102 using any one of a number of well-known techniques



including, but not limited to, passwords, passphrases, certificates, public key techniques, or biometric techniques.

[0043] Once the player is properly authenticated, the server platform 102 requests gameplay parameters for the authenticated player from the administration server 310 (step 604). The server platform 102 may issue a Common Gateway Interface (CGI) request using the HyperText Transfer Protocol (HTTP) to the administration server 310. In other embodiments, the server platform 102 transmits an SQL-compliant database query to the administration server 310. In still other embodiments, the server platform 102 may issue an XML-RPC request using the HyperText Transfer Protocol (HTTP) or other suitable interface mechanism to the administration or other intermediary server. Other embodiments may include SOAP, RPC, or DCOM. The administration server 310 retrieves gameplay parameters for the authenticated player and transmits the parameters to the server platform 102.

[0044] In the embodiment of steps shown in FIG. 6, the server platform 102 first checks the gameplay calendar to determine if the player is permitted to play during this time period (step 606). In some embodiments, this determination is effected by comparing the current time to a table representing the gameplay calendar. The table includes each time period with a flag indicating whether gameplay is permitted or not during the associated period. If the gameplay calendar indicates that gameplay is prohibited, the player's session is terminated (step 608). In some embodiments, the server platform 102 may terminate a player's session if the player attempts to login to the game with insufficient time to play, e.g., five minutes before a prohibited period of time. Time period information may be stored relative to the game player's local time or relative to a referenced time zone, such as Greenwich Mean Time (GMT).

[0045] If the gameplay parameters indicate that play is allowed during the current time period, in the embodiment shown in FIG. 6, the server platform 610 next checks to determine if the player has exceeded total allowed game time during the period (step 612). In some embodiments, this is done by comparing a value representing total time played to the time limit entered in the Restrictions pull-down menu 552. If the player's totals for game play exceed the entered limit, the player's session is terminated (step 608). In some embodiments, the server platform 102 may



terminate the player's session if insufficient time remains to achieve meaningful interaction with the game.

**[0046]** If the player is authorized to play during the current time period and has not exceeded her gameplay time limits, the server platform 102 checks the player's gameplay parameters to determine if chat logging should be enabled (step 612). If not, active gameplay starts at once. If, however, chat logging should be enabled, the server platform 102 transmits a message to the player's client instructing it to log all chat activity (step 614). In some embodiments, the client effects chat logging by storing a copy of all outgoing and incoming chat messages in a secure file on the client platform that may only be accessed by the supervising user. In other embodiments, the client platform transmits a copy of all outgoing and incoming chat messages to a separate computer or to the administration client 310 for storing and later review by the supervising user. In still other embodiments, the client may store all chat messages in a file which is emailed to the supervising user. In some of these embodiments, the client may send batches of chat messages without regard to whether the current chat session has ended.

**[0047]** In the embodiment depicted in FIG. 6, the server platform 102 activates a timer during active gameplay (step 616). If the user voluntarily logs out of the game (step 618), the player's session is terminated (step 608). Otherwise, the timer value is used to ensure that the player does not exceed gameplay limits during the current session (step 620). In some embodiments, this is accomplished by periodically adding together the timer value with the gameplay total from the player's gameplay parameters. The total is checked at any interval determined to provide sufficient resolution. This total may be checked every hour, every half hour, every fifteen minutes, every five minutes or every minute. If the total exceeds the player's gameplay limit, the user's session is terminated (step 608).

**[0048]** The server platform 102 also periodically checks the gameplay calendar to determine if the user is authorized to play during the current time period (step 620). The total is checked at any interval determined to provide sufficient resolution. This may be done by the server platform 102 once an hour. In other embodiments, the administration server may send a message to the server platform 102 at the start of each hour with a list of players prohibited from playing during the next hour. If any of those users are logged in, their game sessions are terminated. In other

embodiments, the server platform 102 may send a message to a centralized access system or other intermediary system or server with a request that is checked against a database for access.

[0049] Although it has been described with respect to a supervising user and a game player, it is contemplated that certain players may provide parameters for their own account in a form a self-regulation. Players may do this because they recognize that they spend excessive amounts of time playing the game. Alternatively, players may use these parameters to stay within certain financial boundaries. For example, a player that pays a set price for twenty hours of game access with additional fees required for each hour over twenty can instruct the game to prohibit her access to the game in excess of twenty hours.

[0050] The present invention can be provided as one or more computer-readable programs embodied on or in one or more articles of manufacture. The article of manufacture can be a floppy disk, a hard disk, a CD ROM, CD-R, CD-RW, DVD, DVD-RAM a flash memory card, a USB storage device, a PROM, a RAM, a ROM, or a magnetic tape. In general, the computer-readable programs can be implemented in any programming language. Some examples of languages that can be used include C, C++, or JAVA. The software programs can be stored on or in one or more articles of manufacture as object code.

[0051] Methods and systems to limit player access to a MMP have been described with respect to preferred embodiments; however, the methods and systems of the present invention are not limited to the preferred embodiments. In particular, various arrangements of data to be stored in the database and used to control access are readily apparent to those of ordinary skill in the art. The skilled artisan will readily appreciate that various omissions, additions and modifications can be made to the methods and systems described above without departing from the scope of the invention, and all such modifications and changes are intended to fall within the scope of the invention, as defined by the appended claims.



CLAIMS

1. A method for controlling access to a massively multi-player, online role-playing game, the method comprising the steps of:
  - (a) presenting a user interface to a first user;
  - (b) receiving, via the user interface, a parameter for controlling access of a second user to a massively multi-player, online role-playing game;
  - (c) transmitting to a database server the received parameter;
  - (d) storing the received parameter in a database record associated with the second user;
  - (e) receiving, at a game server, a connection request from the second user; and
  - (f) controlling access of the second user to the massively multi-player, online role playing game using the stored parameter.
2. The method of claim 1 wherein step (a) comprises presenting a web page to a user, the web page displaying to the first user information regarding gameplay behavior of a second user.
3. The method of claim 1 wherein step (a) comprises presenting a web page to a first user, the web page displaying information regarding gameplay statistics for a second user.
4. The method of claim 3 wherein the web page displays information regarding the most recent login time of a second user.
5. The method of claim 3 wherein the web page displays information regarding total playtime of a second user during the most recent game session of the second user.
6. The method of claim 1 wherein step (b) comprises receiving, via the user interface, a parameter for prohibiting access of a second user to the game for a predetermined amount of time.
7. The method of claim 1 wherein step (b) comprises receiving, via the user interface, a parameter for terminating a game session of a second user.

8. The method of claim 1 wherein step (b) comprises receiving, via the user interface, a parameter for prohibiting access of a second user to the game for a predetermined period of time on a predetermined day of the week.
9. The method of claim 1 wherein step (f) comprises:
  - (f-1) transmitting, by the game server in response to the connection request, an authentication request to the database server, the transmitted request identifying the second user and the time of the connection request;
  - (f-2) comparing, by the database server, the second user and the time of the connection request to the database record associated with the second user;
  - (f-3) sending to the game server, by the database server, a authentication denial message; and
  - (f-4) refusing, by the game server, access of the second user to the game.
10. The method of claim 1 wherein step (f) comprises:
  - (f-1) transmitting, by the game server in response to the connection request, an authentication request to the database server, the transmitted request identifying the second user;
  - (f-2) sending, by the database server, the database record associated with the second user;
  - (f-3) comparing, by the game server, the second user and the time of the connection request to the transmitted database record; and
  - (f-4) refusing, by the game server, access of the second user to the game responsive to the comparison.
11. An article of manufacture embodying computer-readable programs means for controlling access to a massively multi-player, online role-playing game, the article of manufacture comprising:
  - (a) computer-readable program means for presenting a user interface to a first user;
  - (b) computer-readable program means for receiving, via the user interface, a parameter for controlling access of a second user to a massively multi-player, online role-playing game;
  - (c) computer-readable program means for transmitting to a database server the received parameter;



- (d) computer-readable program means for storing the received parameter in a database record associated with the second user;
  - (e) computer-readable program means for receiving, at a game server, a connection request from the second user; and
  - (f) computer-readable program means for controlling access of the second user to the massively multi-player, online role playing game using the stored parameter.
12. The article of manufacture of claim 11 further comprising computer-readable program means for presenting a web page to a user, the web page displaying to the first user information regarding gameplay behavior of a second user.
13. The article of manufacture of claim 11 further comprising computer-readable program means for presenting a web page to a first user, the web page displaying information regarding gameplay statistics for a second user.
13. The article of manufacture of claim 11 further comprising computer-readable program means for receiving, via the user interface, a parameter for prohibiting access of a second user to the game for a predetermined amount of time.
14. The article of manufacture of claim 11 further comprising computer-readable program means for receiving, via the user interface, a parameter for terminating a game session of a second user.
15. The article of manufacture of claim 11 further comprising computer-readable program means for receiving, via the user interface, a parameter for prohibiting access of a second user to the game for a predetermined period of time on a predetermined day of the week.
16. The article of manufacture of claim 11 further comprising:
- (f-1) computer-readable program means for transmitting, by the game server in response to the connection request, an authentication request to the database server, the transmitted request identifying the second user and the time of the connection request;

- (f-2) computer-readable program means for comparing, by the database server, the second user and the time of the connection request to the database record associated with the second user;
  - (f-3) computer-readable program means for sending to the game server, by the database server, a authentication denial message; and
  - (f-4) computer-readable program means for refusing, by the game server, access of the second user to the game.
17. The article of manufacture of claim 11 further comprising:
- (f-1) computer-readable program means for transmitting, by the game server in response to the connection request, an authentication request to the database server, the transmitted request identifying the second user;
  - (f-2) computer-readable program means for sending, by the database server, the database record associated with the second user;
  - (f-3) computer-readable program means for comparing, by the game server, the second user and the time of the connection request to the transmitted database record; and
  - (f-4) computer-readable program means for refusing, by the game server, access of the second user to the game responsive to the comparison.
18. A system for controlling access to a massively multi-player, online role-playing game, the system comprising:
- a user interface server presenting a user interface to a first user and receiving from the first user a parameter for controlling access of a second user to the game;
  - a database server in communication with the user interface server, the database server receiving from the user interface server the parameter for controlling access of the second user to the game and storing the parameter in a database record associated with the second user; and
  - a game server in communication with the database server, the game server receiving a connection request from the second user and prohibiting access to the game responsive to data received from the database server.



19. The system of claim 18 wherein the database server further comprises an Open Database Connectivity-compliant database.
20. The system of claim 18 wherein the user interface server communicates with the database server over a first network and the database server communicates with the game server over a second network.

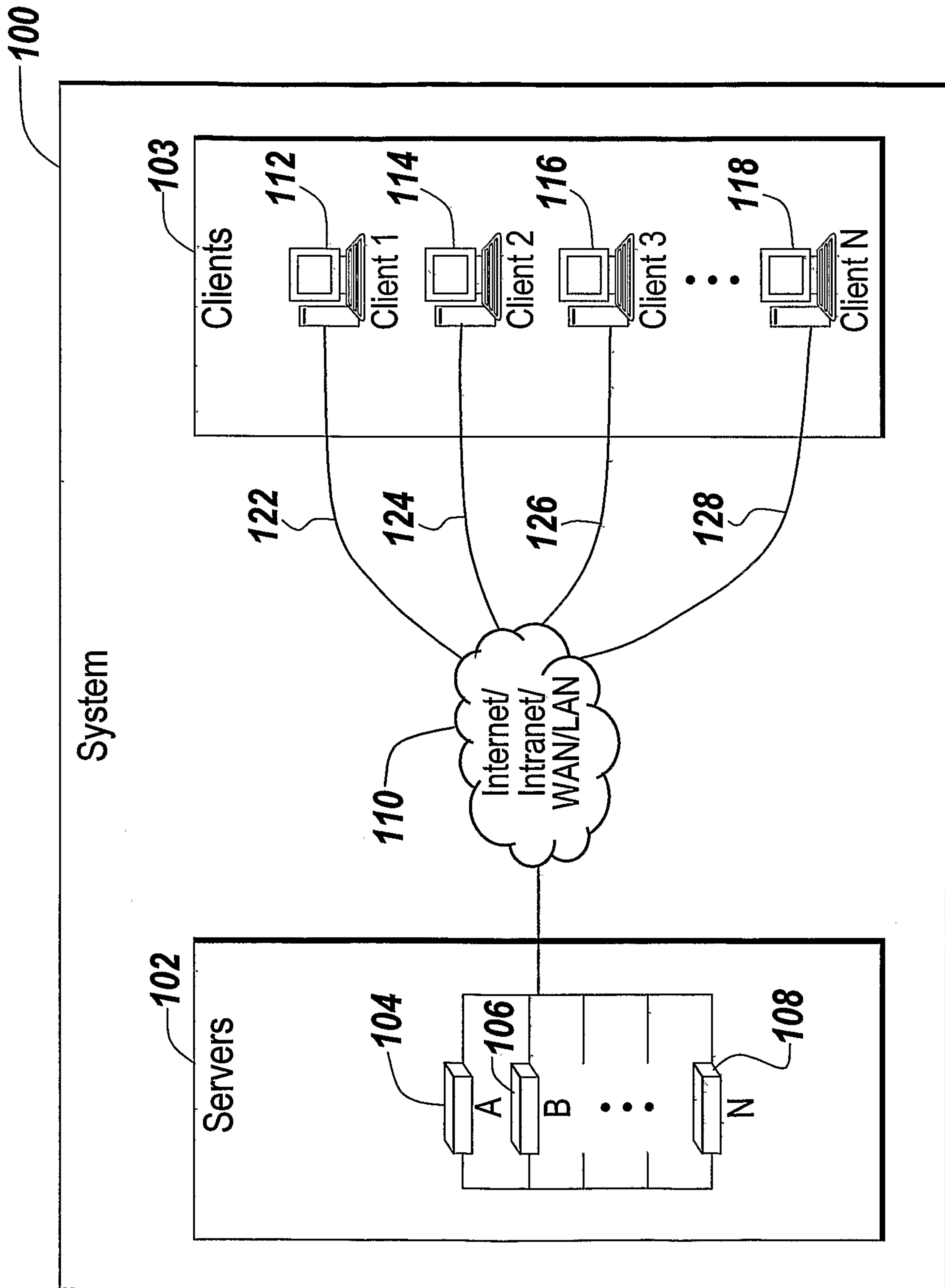
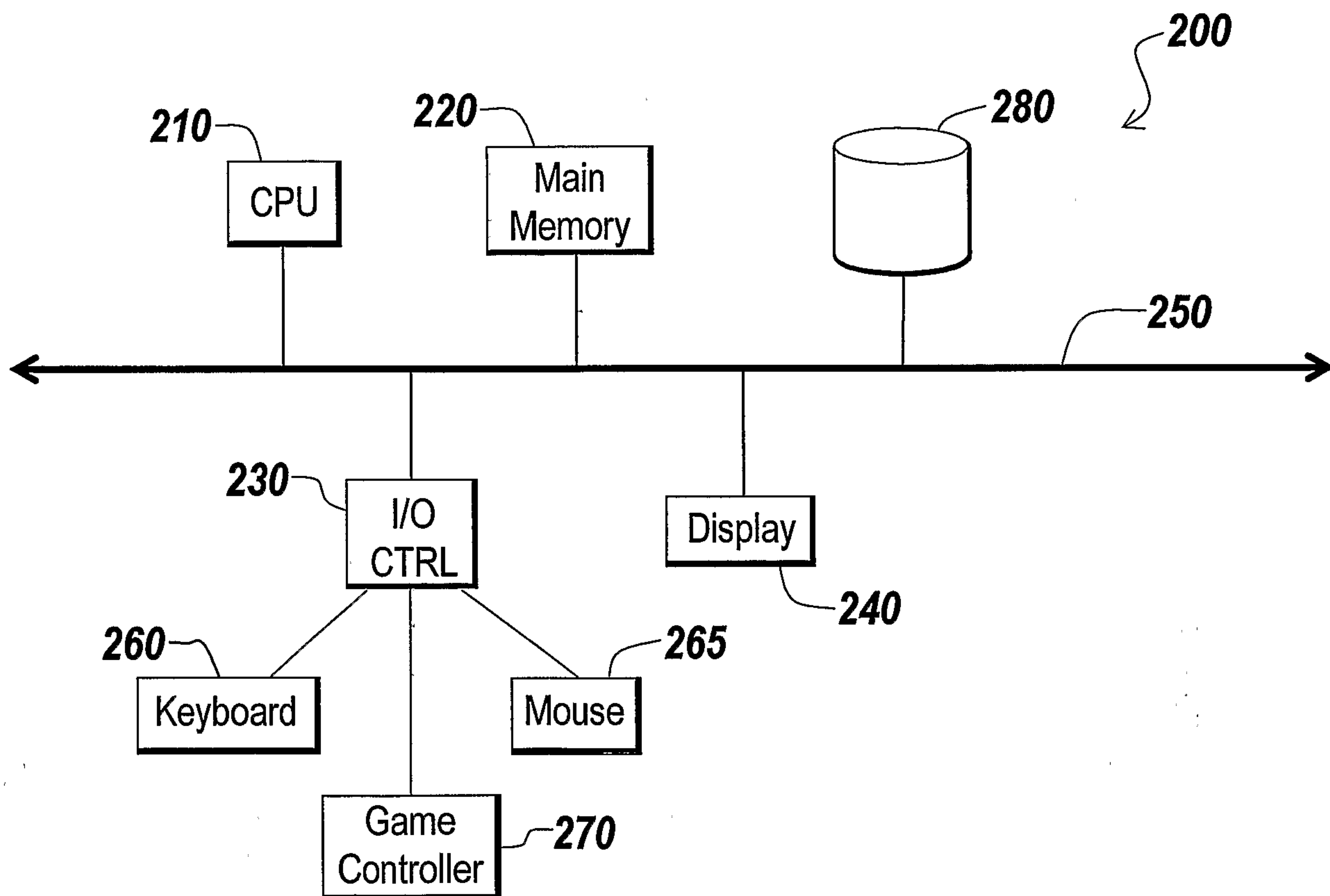


Fig. 1





*Fig. 2*

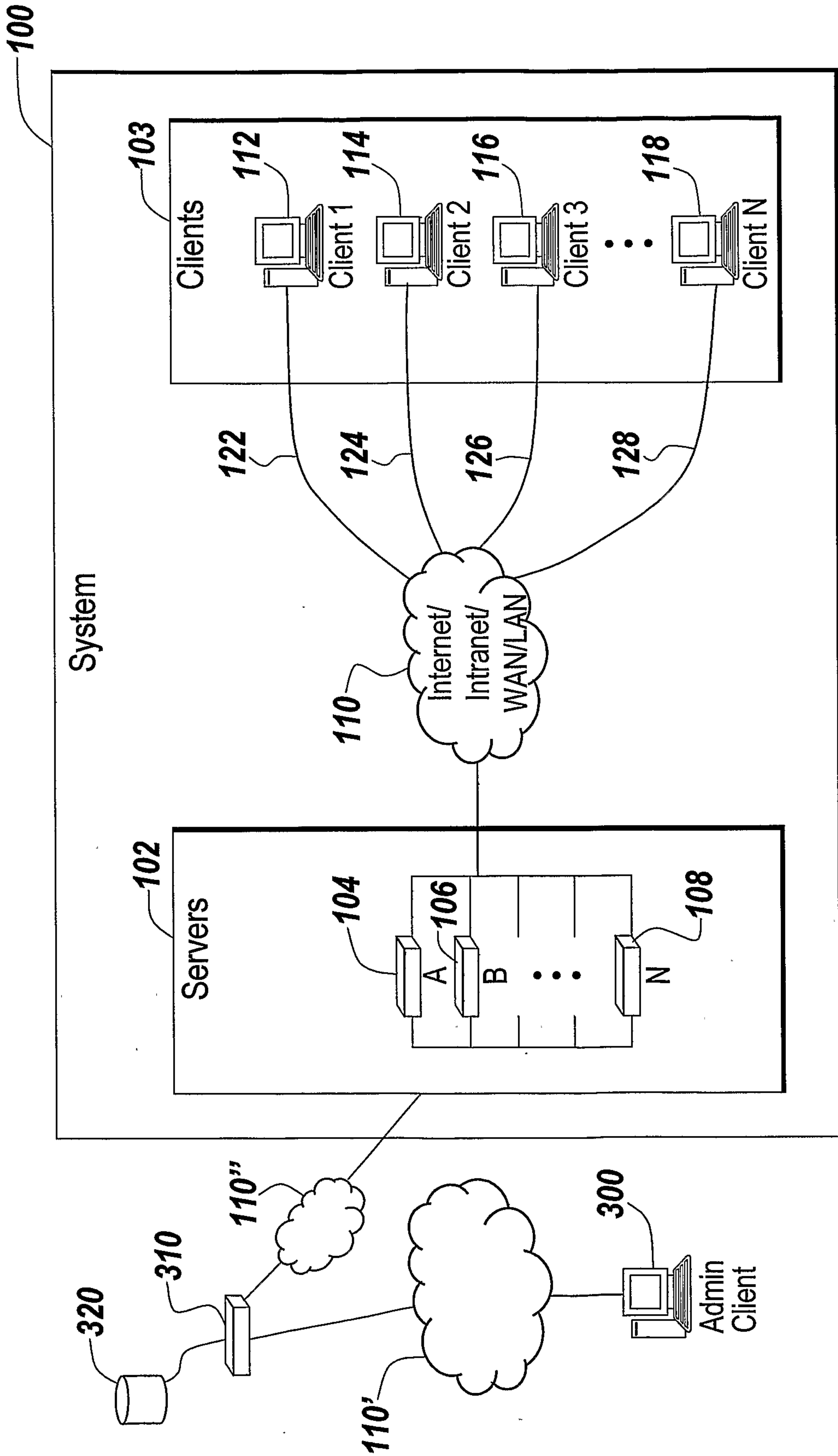


Fig. 3





**Account Management: Parental Controls**

Welcome to the Middle-earth Online Parental Controls options. Middle-earth Online provides these tools to help parents and guardians ensure a safe and enjoyable game play experience for their children.

**Game Time-Out**  
 The Game Time-Out is a tool that logs your child out of the game, terminating the current playsession. **\*Using this feature may cause your child to fail quests they are on in the game!\***  
 Time-Out

**Recent Parental Notifications**  
 1/2/2003 5:00 – "Temporary suspension due to Code of Conduct violation.."  
 12/5/2003 3:15 – "Temporary suspension due to Code of Conduct violation.."  
 View All

**Gameplay History**  
 (gameplay history description)  
 Today's gameplay history for this account:

Today's History	Total Hours	Time Played
2003-7-10	4hrs 2mins	6:00am – 2:00pm

View More

Send me weekly summaries of my child's game history.

**Chat Logging**  
 Use this control to enable or disable chat logging.  
 On Off

**Set Restrictions**  
 No more than 15 hours per week Edit  
 Play only during these times:

Monday	7am – 9am, 5pm – 11pm
Tuesday	5pm – 11pm
Wednesday	5pm – 11pm
Thursday	5pm – 11pm
Friday	5pm – 12am
Saturday	12am – 12am
Sunday	12am – 12am

Fig. 4

510

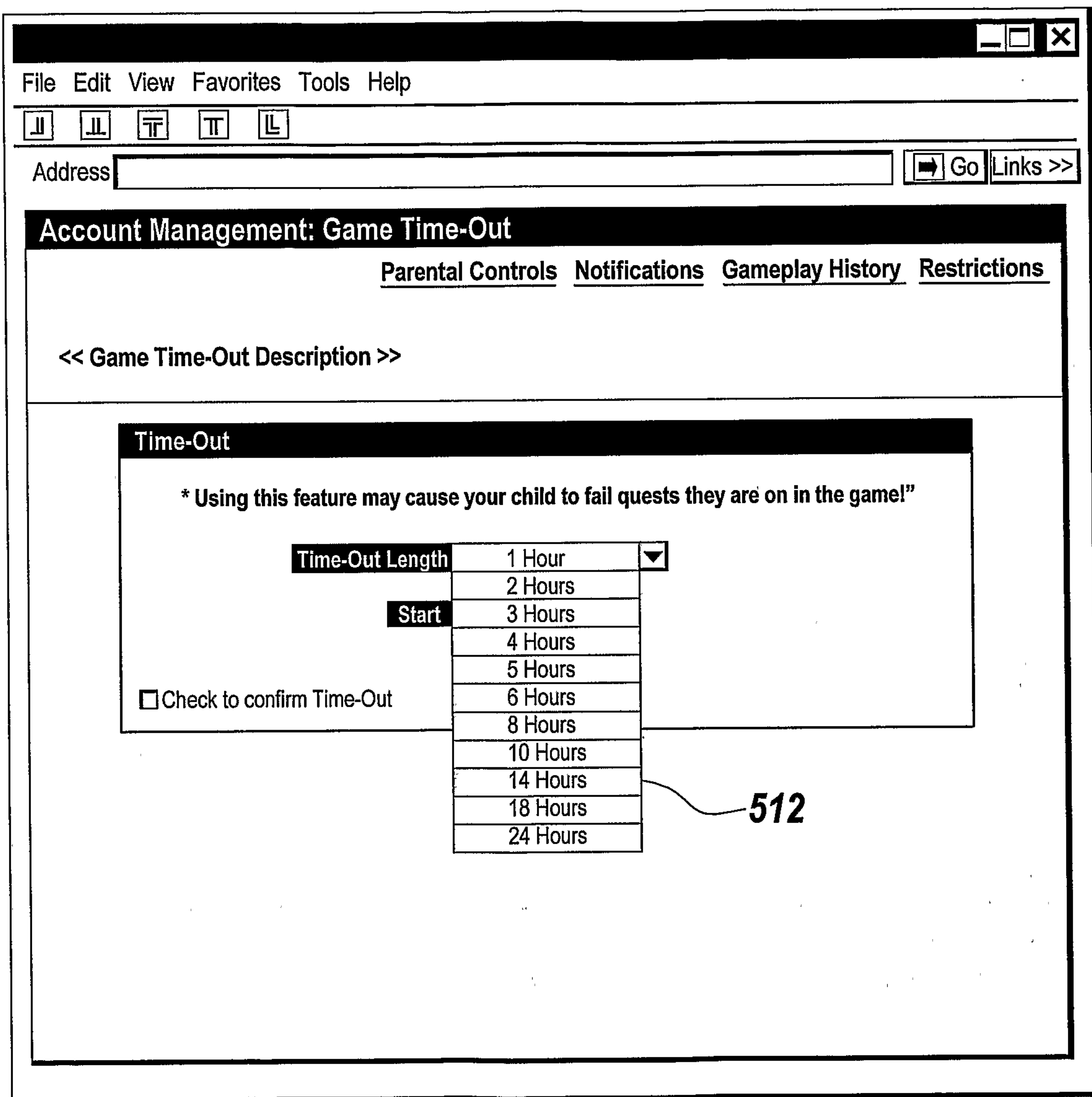


Fig. 5A



520

524  
528

The screenshot shows a web browser window with a menu bar (File, Edit, View, Favorites, Tools, Help) and a toolbar with navigation icons. The address bar is empty. The main content area is titled "Account Management: Parental Notifications" and includes sub-links for "Parental Controls", "Notifications", "Gameplay History", and "Restrictions". A navigation link "<< Parental Notifications Descriptions >>" is present. Below this is a table of "Parental Notifications" with three rows of suspension records. Each row includes start/end dates, appeal status, and appeal outcomes. A note for each record states: "Temporary Suspension for Code of Conduct Violation. Notes from CS - 'User was swearing at other players in starting area. Player was asked to stop but did not. 24 hour suspension.'" At the bottom, there is a pagination control: "Pages (3) : 1 >> 2 >> 3".

Parental Notifications Descriptions >> 526 529 522

Parental Notifications		
Start Date & Time 11/2/2003 12:48pm	End Date & Time 11/3/2003 12:48pm	Temporary Suspension for Code of Conduct Violation Notes from CS - "User was swearing at other players in starting area. Player was asked to stop but did not. 24 hour suspension."
Appeal Yes	Appeal Outcome Pending	
Start Date & Time 9/12/2003 12:48pm	End Date & Time 9/13/2003 12:48pm	Temporary Suspension for Code of Conduct Violation Notes from CS - "User was swearing at other players in starting area. Player was asked to stop but did not. 24 hour suspension."
Appeal Yes	Appeal Outcome Suspension Upheld	
Start Date & Time 8/1/2003 11:48 pm	End Date & Time 8/1/2003 12:48pm	Temporary Suspension for Code of Conduct Violation Notes from CS - "User was swearing at other players in starting area. Player was asked to stop but did not. 24 hour suspension."
Appeal No	Appeal Outcome No	

Pages (3) : 1 >> 2 >> 3

Fig. 5B

530

The screenshot shows a web browser window with a menu bar (File, Edit, View, Favorites, Tools, Help) and a toolbar. The address bar is empty. The main content area is titled 'Account Management: Gameplay History' and includes navigation links for 'Parental Controls', 'Notifications', 'Gameplay History', and 'Restrictions'. Below these is a section for 'Gameplay History' with a description: 'These are the times and length of time that your child has played the game in the last X days.' A dropdown menu labeled 'Show me the last' is open, showing options for 5, 10, 15, 20, and 30 days. A 'Search!' button is next to it. Below the dropdown is a table with three columns: 'Date', 'Total Hours', and 'Last Login'. The table contains five rows of data. At the bottom of the table is a pagination control: 'Pages (3) : 1 >> 2 >> 3'. Reference numbers 532 and 534 point to the dropdown menu and the table respectively.

**Gameplay History**

These are the times and length of time that your child has played the game in the last X days.

Show me the last **5 Days** Search!

5 Days
10 Days
15 Days
20 Days
30 Days

Date	Total Hours	Last Login
2003-7-10	10	1:00am
2003-7-9	2	2:00pm
2003-7-8	5	1:10am
2003-7-7	9	2:10pm
2003-7-6	1	1:20am

Pages (3) : 1 >> 2 >> 3

Fig. 5C



550

File Edit View Favorites Tools Help

Address

**Account Management: Game Time Calendar**

[Parental Controls](#) [Notifications](#) [Gameplay History](#) [Restrictions](#)

<< Game Time Calendar Description >>

**Game Time Calendar**

How many hours per week do you want your child to play MEO?  1 Hour  **552**

What times each day do you want your child to be able to play Middle-earth Online? Select the hours. **556** **554**

Time	Weekends	Weekdays	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
12:00am	■							■	■
1:00am	■							■	■
2:00am	■							■	■
3:00am									
4:00am									
5:00am									
6:00am									
7:00am	■							■	■
8:00am	■							■	■
9:00am	■							■	■
10:00am	■							■	■
11:00am	■							■	■
12:00pm	■							■	■
1:00pm	■							■	■
2:00pm	■							■	■
3:00pm	■							■	■
4:00pm	■							■	■
5:00pm	■	■	■	■	■	■	■	■	■
6:00pm	■	■	■	■	■	■	■	■	■
7:00pm	■	■	■	■	■	■	■	■	■
8:00pm	■	■	■	■	■	■	■	■	■
9:00pm	■	■	■	■	■	■	■	■	■
10:00pm	■						■	■	
11:00pm	■						■	■	

**564** **562**  = Play Time  = Non-Play Time **560** **558**

Fig. 5D

*Fig. 6*

