

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
21 February 2008 (21.02.2008)

PCT

(10) International Publication Number
WO 2008/019956 A1

- (51) International Patent Classification:
A63F 13/12 (2006.01) H04L 29/08 (2006.01)
H04L 29/06 (2006.01)
- (21) International Application Number:
PCT/EP2007/058026
- (22) International Filing Date: 2 August 2007 (02.08.2007)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
11/465,385 17 August 2006 (17.08.2006) US
- (71) Applicant (for all designated States except US): INTERNATIONAL BUSINESS MACHINES CORPORATION [US/US]; New Orchard Road, Armonk, NY 10504 (US).
- (71) Applicant (for MG only): IBM UNITED KINGDOM LIMITED [GB/GB]; PO Box 41, North Harbour, Portsmouth Hampshire PO6 3AU (GB).
- (72) Inventors; and
- (75) Inventors/Applicants (for US only): GARBOW,

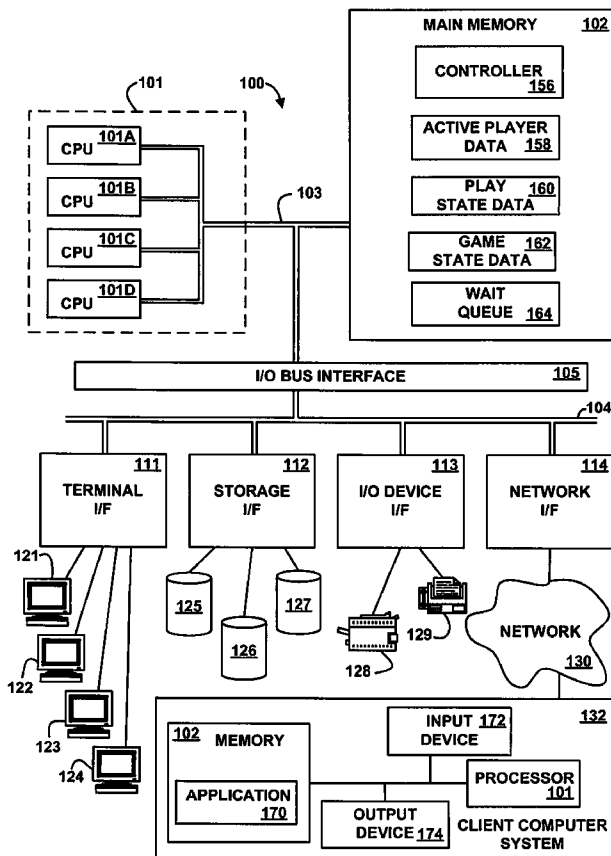
Zachary, Adam [US/US]; 5855 Camellia Court NW, Rochester, MN 55901 (US). LOGAN, Bryan, Mark [US/US]; 4746 Harvest Court NW, Rochester, MN 55901 (US).

- (74) Agent: GASCOYNE, Belinda; IBM United Kingdom Limited, Intellectual Property Law, Hursley Park, Winchester Hampshire SO21 2JN (GB).
- (81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, 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, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, 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,

[Continued on next page]

(54) Title: REPLACING A DEPARTING PLAYER IN A MULTI-PLAYER NETWORK GAME WITH A WAITING PLAYER

(57) Abstract: In an embodiment, if a departing player requests to leave a game before the game is complete, a waiting player is selected and the waiting player replaces the departing player in the game. Replacing the departing player with the waiting player includes assigning a play state of the departing player to the waiting player. The play state is then updated in response to game control input events received from the waiting player. In various embodiments, the waiting player is selected based on skill levels, play styles, time preferences, or game stage preferences of the players, or based on an input criteria specified by the departing or remaining players. In various embodiments, the play state may include a location within the game, an inventory, or a score.



WO 2008/019956 A1



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, IS, IT, LT, LU, LV, MC, MT, 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*
- *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

REPLACING A DEPARTING PLAYER IN A MULTI-PLAYER NETWORK GAME WITH A WAITING
PLAYER**FIELD**

5 The invention relates to the field of computer gaming, and in particular to the replacement of
a player in a network gaming.

BACKGROUND

10 A common use of computer technology is for the playing of games. Computer game
systems come in a variety of forms, such as stand-alone arcade games, personal computer
games, or dedicated game systems (e.g., the Sony® Playstation® console or the Microsoft®
Xbox® console). Some games are single-player games (e.g., the solitaire card game) and
other games are multi-player games where multiple players compete against one another
(e.g., a football simulation game). When playing a multi-player computer game, a player
15 may either play against another human player, who may be in the same room accessing the
same computer or may be an anonymous player who is far away and connected via a
network, or may play against a computer-generated opponent, which is often called an AI
(Artificial Intelligence) opponent.

20 Although AI technology has advanced in recent years, and AI opponents can be quite
sophisticated, many players prefer to play against other human opponents, even if
anonymous, because of the satisfaction of competing with a live person. But, while
human opponents have advantages, they also have disadvantages, such as having a variety of
schedules, skills, interest levels, and commitment levels to continue playing the game. As
such, when a departing player leaves a network game prematurely prior to completion of the
25 game, the remaining player may feel slighted because of the time spent in the preparation
phase of the game, without being able to complete the game. Some game systems end the
game immediately in response to a player departing early, while others replace the departing
player with an AI (Artificial Intelligence) opponent. Neither response to a departing player
is a good solution for the remaining player or players.

Hence, what is needed is a better technique for replacing a departing player with another player, in order to increase the satisfaction of players.

SUMMARY

5 The present invention is defined in the appended claims to which reference should now be made.

In an embodiment, a network game is played over a computer network by a remaining player(s) and a departing player, and controlled by a controlling computer system. A
10 determination is made that the departing player requests to leave a game before the game is complete. A waiting player is selected and the waiting player replaces the departing player in the game. Replacing the departing player with the waiting player includes assigning a play state of the departing player to the waiting player. The play state is then updated in response to a game control input event received from the waiting player.

15 The selection of the waiting player may be made by a variety of input criteria, such as determining whether the waiting player has a skill level that is within a skill threshold of the skill level of the departing player, determining whether the waiting player has a play style that matches the play style of the departing player, determining whether the waiting player
20 has a time preference that is within a time threshold of a time period that remains before the game is complete, determining whether the waiting player has a game stage preference that matches a stage of the game, or determining whether the waiting player meets an input criteria specified by the departing player or the remaining player.

25 Assigning the play state may include assigning a location within the game of the departing player to the waiting player, assigning an inventory of the departing player to the waiting player, or assigning a score of the departing player to the waiting player.

30

BRIEF DESCRIPTION OF THE DRAWINGS

Various embodiments of the present invention are hereinafter described in conjunction with the appended drawings:

5

Fig. 1 depicts a high-level block diagram of an example system for implementing an embodiment of the invention.

10

Fig. 2 depicts a block diagram of an example data structure for active player data, according to an embodiment of the invention.

Fig. 3 depicts a block diagram of an example data structure for play state data, according to an embodiment of the invention.

15

Fig. 4 depicts a block diagram of another example data structure for play state data, according to an embodiment of the invention.

Fig. 5 depicts a block diagram of an example data structure for game state data, according to an embodiment of the invention.

20

Fig. 6 depicts a block diagram of an example data structure for a wait queue, according to an embodiment of the invention.

25

Fig. 7 depicts a flowchart of example processing for events at a controller, according to an embodiment of the invention.

Fig. 8 depicts a flowchart of example processing for handling a new player event, according to an embodiment of the invention.

30

Fig. 9 depicts a flowchart of example processing for handling a departing player leaving a game before the game is complete, according to an embodiment of the invention.

Fig. 10 depicts a flowchart of example processing for replacing a departing player, according to an embodiment of the invention.

Fig. 11 depicts a flowchart of example processing for calculating a play style of a player,

5

It is to be noted, however, that the appended drawings illustrate only example embodiments of the invention, and are therefore not considered limiting of its scope, for the invention may admit to other equally effective embodiments.

10

DETAILED DESCRIPTION

Referring to the Drawings, wherein like numbers denote like parts throughout the several views, Fig. 1 depicts a high-level block diagram representation of a server computer system 100 connected to a client computer system 132 via a network 130, such as the Intranet,

15

according to an embodiment of the present invention. The terms “client” and “server” are used herein for convenience only, and in various embodiments a computer that operates as a client in one environment may operate as a server in another environment, and vice versa. In an embodiment, the hardware components of the computer system 100 may be implemented by a System i5 computer system available from International Business Machines of Armonk,

20

New York. However, those skilled in the art will appreciate that the mechanisms and apparatus of embodiments of the present invention apply equally to any appropriate computing system.

Although only one client computer system 132 is shown in Figure 1, in preferred

25

embodiments a plurality of such client computer system are connected to the server via network 130, which may be any kind of network, such as a Local Area Network, a telecommunications network, or the Internet.

The major components of the computer system 100 include one or more processors 101, a main memory 102, a terminal interface 111, a storage interface 112, an I/O (Input/Output) device interface 113, and communications/network interfaces 114, all of which are coupled

30

for inter-component communication via a memory bus 103, an I/O bus 104, and an I/O bus interface unit 105.

5 The computer system 100 contains one or more general-purpose programmable central processing units (CPUs) 101A, 101B, 101C, and 101D, herein generically referred to as the processor 101. In an embodiment, the computer system 100 contains multiple processors typical of a relatively large system; however, in another embodiment the computer system 100 may alternatively be a single CPU system. Each processor 101 executes instructions stored in the main memory 102 and may include one or more levels of on-board cache.

10 The main memory 102 is a random-access semiconductor memory for storing or encoding data and programs. In another embodiment, the main memory 102 represents the entire virtual memory of the computer system 100, and may also include the virtual memory of other computer systems coupled to the computer system 100 or connected via the network
15 130. The main memory 102 is conceptually a single monolithic entity, but in other embodiments the main memory 102 is a more complex arrangement, such as a hierarchy of caches and other memory devices. For example, memory may exist in multiple levels of caches, and these caches may be further divided by function, so that one cache holds instructions while another holds non-instruction data, which is used by the processor or
20 processors. Memory may be further distributed and associated with different CPUs or sets of CPUs, as is known in any of various so-called non-uniform memory access (NUMA) computer architectures.

25 The memory 102 includes a controller 156, active player data 158, play state data 160, game state data 162, and a wait queue 164. Although the controller 156, the active player data 158, the play state data 160, the game state data 162, and the wait queue 164 are illustrated as being contained within the memory 102 in the computer system 100, in other
30 embodiments some or all of them may be on different computer systems and may be accessed remotely, e.g., via the network 130. The computer system 100 may use virtual addressing mechanisms that allow the programs of the computer system 100 to behave as if they only have access to a large, single storage entity instead of access to multiple, smaller storage entities. Thus, while the controller 156, the active player data 158, the play state

data 160, the game state data 162, and the wait queue 164 are illustrated as being contained within the main memory 102, they are not necessarily all completely contained in the same storage device at the same time. Further, although the controller 156, the active player data 158, the play state data 160, the game state data 162, and the wait queue 164 are illustrated as being separate entities, in other embodiments some or all of them may be packaged together.

The active player data 158 represents the players who are actively playing a game. The active player data 158 is further described below with reference to Fig. 2. The play state data 160 represents the state of active players within a game. Examples of the play state data 160 are further described below with reference to Figs. 3 and 4. The game state data 162 represents the state of a game. The game state data 162 is further described below with reference to Fig. 5. The wait queue 164 represents players who are not actively playing a game, but that desire to play a game and are thus waiting to play a game. The wait queue 164 is further described below with reference to Fig. 6.

The controller 156 receives events from players at a plurality of client computer systems 132, and in response updates the active player data 158, the play state data 160, the game state data 162, and the wait queue 164 and sends the play state data 160 and the game state data 162 to the client computer system 132. In an embodiment, the controller 156 includes instructions capable of executing on the processor 101 or statements capable of being interpreted by instructions executing on the processor 101 to perform the functions as further described below with reference to Figs. 7, 8, 9, 10, and 11. In another embodiment, the controller 156 may be implemented in microcode. In another embodiment, the controller 156 may be implemented in hardware via logic gates and/or other appropriate hardware techniques.

The memory bus 103 provides a data communication path for transferring data among the processor 101, the main memory 102, and the I/O bus interface unit 105. The I/O bus interface unit 105 is further coupled to the system I/O bus 104 for transferring data to and from the various I/O units. The I/O bus interface unit 105 communicates with multiple I/O interface units 111, 112, 113, and 114, which are also known as I/O processors (IOPs) or I/O

adapters (IOAs), through the system I/O bus 104. The system I/O bus 104 may be, e.g., an industry standard PCI (Peripheral Component Interface) bus, or any other appropriate bus technology.

5 The I/O interface units support communication with a variety of storage and I/O devices. For example, the terminal interface unit 111 supports the attachment of one or more user terminals 121, 122, 123, and 124. The storage interface unit 112 supports the attachment of one or more direct access storage devices (DASD) 125, 126, and 127 (which are typically rotating magnetic disk drive storage devices, although they could alternatively be other
10 devices, including arrays of disk drives configured to appear as a single large storage device to a host). The contents of the main memory 102 may be stored to and retrieved from the direct access storage devices 125, 126, and 127, as needed.

The I/O device interface 113 provides an interface to any of various other input/output
15 devices or devices of other types. Two such devices, the printer 128 and the fax machine 129, are shown in the exemplary embodiment of Fig. 1, but in other embodiment many other such devices may exist, which may be of differing types. The network interface 114 provides one or more communications paths from the computer system 100 to other digital devices and computer systems; such paths may include, e.g., one or more networks 130.

20 Although the memory bus 103 is shown in Fig. 1 as a relatively simple, single bus structure providing a direct communication path among the processors 101, the main memory 102, and the I/O bus interface 105, in fact the memory bus 103 may comprise multiple different buses or communication paths, which may be arranged in any of various forms, such as
25 point-to-point links in hierarchical, star or web configurations, multiple hierarchical buses, parallel and redundant paths, or any other appropriate type of configuration. Furthermore, while the I/O bus interface 105 and the I/O bus 104 are shown as single respective units, the computer system 100 may in fact contain multiple I/O bus interface units 105 and/or multiple I/O buses 104. While multiple I/O interface units are shown, which separate the
30 system I/O bus 104 from various communications paths running to the various I/O devices, in other embodiments some or all of the I/O devices are connected directly to one or more system I/O buses.

The computer system 100 depicted in Fig. 1 has multiple attached terminals 121, 122, 123, and 124, such as might be typical of a multi-user “mainframe” computer system. Typically, in such a case the actual number of attached devices is greater than those shown in Fig. 1, although the present invention is not limited to systems of any particular size. The computer system 100 may alternatively be a single-user system, typically containing only a single user display and keyboard input, or might be a server or similar device which has little or no direct user interface, but receives requests from other computer systems (clients). In other embodiments, the computer system 100 may be implemented as a personal computer, portable computer, laptop or notebook computer, PDA (Personal Digital Assistant), tablet computer, pocket computer, telephone, pager, automobile, teleconferencing system, appliance, or any other appropriate type of electronic device.

The network 130 may be any suitable network or combination of networks and may support any appropriate protocol suitable for communication of data and/or code to/from the computer system 100. In various embodiments, the network 130 may represent a storage device or a combination of storage devices, either connected directly or indirectly to the computer system 100. In an embodiment, the network 130 may support the Infiniband architecture. In another embodiment, the network 130 may support wireless communications. In another embodiment, the network 130 may support hard-wired communications, such as a telephone line or cable. In another embodiment, the network 130 may support the Ethernet IEEE (Institute of Electrical and Electronics Engineers) 802.3x specification. In another embodiment, the network 130 may be the Internet and may support IP (Internet Protocol).

In another embodiment, the network 130 may be a local area network (LAN) or a wide area network (WAN). In another embodiment, the network 130 may be a hotspot service provider network. In another embodiment, the network 130 may be an intranet. In another embodiment, the network 130 may be a GPRS (General Packet Radio Service) network. In another embodiment, the network 130 may be a FRS (Family Radio Service) network. In another embodiment, the network 130 may be any appropriate cellular data network or cell-based radio network technology. In another embodiment, the network 130 may be an IEEE 802.11B wireless network. In still another embodiment, the network 130 may be any

suitable network or combination of networks. Although one network 130 is shown, in other embodiments any number (including zero) of networks (of the same or different types) may be present.

5 The client computer system 132 may include some or all of the hardware and/or software elements previously described above for the computer system 100. The client computer system 132 may be implemented as a computer system, game console, arcade game, or any other appropriate type of electronic device. The client computer system 132 includes a processor 101 connected to memory 102, an input device 172, and an output device 174. In
10 various embodiments, the input device 172 may be a keyboard, mouse, pointing device, microphone, speech recognition device, touch screen, joystick, game control device, throttle, button, lever, wheel, slider, pedal, stick, dial, user interface widget, virtual reality device, any other appropriate input device, or any combination, multiple, or portion thereof. In various embodiments, the output device 174 may be video screen, terminal, virtual reality
15 helmet, printer, speaker, any other appropriate output device, or any combination, multiple, or portion thereof. A player may input information via the input device 172 and receive information via the output device 174.

The memory 102 in the client computer system 132 includes an application 170. The
20 application 170 executes on the processor 101 to send game control input events, new player requests, and requests to depart a game from the input device 172 to the server computer system 100. The application 170 further receives the play state data 160 and the game state data 162 from the server computer system 100 and presents the play state data 160 and the game state data 162, or portions thereof, via the output device 174. Although the client
25 computer system 132 is illustrated as being separate from and connected to the computer system 100 via the network 130, in another embodiment, the client computer system 132 may be part of the computer system 100, e.g., the client 132 may be implemented as a software program and data stored in the memory 102.

30 It should be understood that Fig. 1 is intended to depict the representative major components of the computer system 100, the network 130, and the client computer system 132 at a high level, that individual components may have greater complexity than represented in Fig. 1,

that components other than or in addition to those shown in Fig. 1 may be present, and that the number, type, and configuration of such components may vary. Several particular examples of such additional complexity or additional variations are disclosed herein; it being understood that these are by way of example only and are not necessarily the only such variations.

The various software components illustrated in Fig. 1 and implementing various embodiments of the invention may be implemented in a number of manners, including using various computer software applications, routines, components, programs, objects, modules, data structures, etc., referred to hereinafter as "computer programs," or simply "programs." The computer programs typically comprise one or more instructions that are resident at various times in various memory and storage devices in the computer system 100, and that, when read and executed by one or more processors 101 in the computer system 100, cause the computer system 100 to perform the steps necessary to execute steps or elements comprising the various aspects of an embodiment of the invention.

Moreover, while embodiments of the invention have and hereinafter will be described in the context of fully-functioning computer systems, the various embodiments of the invention are capable of being distributed as a program product in a variety of forms, and the invention applies equally regardless of the particular type of signal-bearing medium used to actually carry out the distribution. The programs defining the functions of this embodiment may be delivered to the computer system 100 via a variety of tangible signal-bearing media that may be operatively or communicatively connected (directly or indirectly) to the processor 101. The signal-bearing media may include, but are not limited to:

(1) information permanently stored on a non-rewriteable storage medium, e.g., a read-only memory device attached to or within a computer system, such as a CD-ROM readable by a CD-ROM drive;

(2) alterable information stored on a rewriteable storage medium, e.g., a hard disk drive (e.g., DASD 125, 126, or 127), CD-RW, or diskette; or

(3) information conveyed to the computer system 100 by a communications medium, such as through a computer or a telephone network, e.g., the network 130.

5 Such tangible signal-bearing media, when encoded with or carrying computer-readable and executable instructions that direct the functions of the present invention, represent embodiments of the present invention.

10 Embodiments of the present invention may also be delivered as part of a service engagement with a client corporation, nonprofit organization, government entity, internal organizational structure, or the like. Aspects of these embodiments may include configuring a computer system to perform, and deploying software systems and web services that implement, some or all of the methods described herein. Aspects of these embodiments may also include analyzing the client company, creating recommendations responsive to the analysis, generating software to implement portions of the recommendations, integrating the software
15 into existing processes and infrastructure, metering use of the methods and systems described herein, allocating expenses to users, and billing users for their use of these methods and systems.

20 In addition, various programs described hereinafter may be identified based upon the application for which they are implemented in a specific embodiment of the invention. But, any particular program nomenclature that follows is used merely for convenience, and thus embodiments of the invention should not be limited to use solely in any specific application identified and/or implied by such nomenclature.

25 The exemplary environments illustrated in Fig. 1 are not intended to limit the present invention. Indeed, other alternative hardware and/or software environments may be used without departing from the scope of the invention.

30 Fig. 2 depicts a block diagram of an example data structure for the active player data 158, according to an embodiment of the invention. The active player data 158 includes example records 205, 210, 215, and 220, each of which includes a player identifier field 225, a skill level field 230, a play style field 235, a time preference field 240, an input criteria field 245,

and an input information field 250. The player identifier field 225 identifies users or players at one or more client computer system(s) 132 who are playing a game.

5 The skill level field 230 indicates a relative or absolute rating of the skill, talent, or aptitude of the associated player 225 playing the game. In an embodiment, the skill 230 may indicate one of multiple skill levels, such as beginner, intermediate, and advanced. In various
embodiments, the skill 230 may be self-assessed or self-reported by the player or may be
assessed, calculated, computed, or selected by the controller 156. In various embodiments,
10 the controller 156 may calculate the skill level 230 for the associated player 225 based on the number of games that the player has played, based on the number of games that the player has won, based on scores that the player has achieved in the games, based on the level of difficulty of games that the player has played, based on the skill of the opponents that the player has faced, based on the scores that the player's opponents have achieved, based on the amount of time that the player has taken to achieve a given score, complete a given level or
15 task within a game, or win the game, or any multiple, combination, or portion thereof.

The play style field 235 characterizes the style of play of the game by the player 225. In various embodiments, the play style 235 may be self-assessed or self-reported by the player or may be assessed, calculated, computed, or selected by the controller 156. In various
20 embodiments, the controller 156 may calculate the play style 235 for the associated player 225 by categorizing the number and type of actions or game control input events that the player has provided, by categorizing the number and type of inventory that the player possesses, by categorizing the number and type of locations within the game that the player or the player's inventory have occupied, or any multiple, combination, or portion thereof, as
25 further described below with reference to Fig. 11.

The time preference field 240 indicates the amount of time that the player 225 prefers to play the game. The input criteria field 245 indicates the criteria that the player 225 prefers to use
30 when the controller 156 selects a waiting player to replace the player 225 or when the controller 156 selects a waiting player to replace an opponent of the player 225. For example, the input criteria 245 may identify a specific player, may identify a minimum or maximum skill level of the waiting player, or may identify a play style of the waiting player.

The input information field 250 includes any information provided by the player 225 that the player 225 desires the controller 156 to communicate to a replacement player, such as a waiting player that the controller 156 replaces the player 225 with in the game.

5 Figs. 3 and 4 depict block diagrams of example data structures for the play state data 160-1 and 160-2, which are examples of the play state data 160 (Fig. 1), according to an embodiment of the invention. The play state data 160-1 includes example records 305 and 310 and the play state data 160-2 includes example records 405 and 410. Each of the records includes a player identifier field 315 and a play state field 320. The player identifier
10 field 315 identifies a player at the client computer system 132. The play state field 320 may include a location or position 325 of the player within the game or of any object in the inventory of the player 315, an inventory 330-1, 330-2, 330-3, or 330-4 for the player 315, and a score 335 for the player 315.

15 The location 325 is relative to the environment (see Fig. 5) of the game and may be expressed as coordinates (in any dimension) or relative to objects that exist in the environment of the game (e.g. in the barn or on top of the house). The location 325 may include the location of any object that represents the player in the game and/or of any object that the player possesses, uses, or controls, such as any object with the inventory 330-1, 330-
20 2, 330-3, or 330-4 of the player 315.

In various embodiments, the inventory 330-1, 330-2, 330-3, and 330-4 includes a specification of objects in the game that the player may possess, use, control or direct. The inventory 330-1, 330-2, 330-3, and 330-4 may include status (e.g., health, strength, power,
25 or intensity) of the player or of any of the objects in the inventory. The inventory 330-1, 330-2, 330-3, or 330-4 may further include any attributes or properties of the objects, such as the speed, acceleration, field of view, direction, orientation, or intensity. The inventory 330-1, 330-2, 330-3, or 330-4 may further include knowledge, information, or talents that the player has about the game or with respect to the objects.

30

The score 335 indicates the progress of the player 315 toward winning or finishing the game and/or sub-games or levels within the game. The score 335 maybe expressed in absolute

terms or relative to other players. The score 335 may have any units, such as points, goals, levels, or achievements accomplished, objects of value, or money.

Fig. 5 depicts a block diagram of an example data structure for the game state data 162, according to an embodiment of the invention. The game state data 162 includes example records 505 and 510, each of which includes a game identifier field 515, a players field 520, and a game state field 525. The game identifier field 515 identifies the game that the players 520 are playing. In an embodiment, the players 520 are opponents in the game 515 and are competing against each other. In another embodiment, some or all of the players 520 may act in cooperation or be on the same team (some or all of the time of the game) within the game 515. The game state 525 indicates the stage and the environment of the game. The stage indicates the elapsed time since the beginning of the game, the time remaining until the end of the game, or the level, hand, turn, tier, or any other appropriate indication of progress within the game. The environment indicates the location, attributes, type, and characteristics (e.g., speed, color, acceleration, force, impact) of any objects within the game, whether or not known to the players, the scenery of the game, and the climate of the game.

Fig. 6 depicts a block diagram of an example data structure for the wait queue 164, according to an embodiment of the invention. The wait queue 164 includes example records 605, 610, 615, and 620, each of which includes a waiting player identifier field 625, a game identifier field 630, a skill level field 635, a play style field 640, a time preference field 645, and a game stage preference field 650. The player identifier field 625 identifies a waiting player who is waiting to play the game identified by the game identifier field 630. The waiting player 625 is not currently playing the game 630. The skill level field 635 indicates the skill of the waiting player 625 and is analogous the skill level 230. The play style field 640 indicates the style of play of the waiting player 625 and is analogous to the play style 235. The time preference field 645 indicates the preferred amount of time that the waiting player 625 desires to the play the game 630. The game stage preference field 650 indicates the relative stage within the game 630 that the player desires to play.

Fig. 7 depicts a flowchart of example processing for events by the controller 156, according to an embodiment of the invention. Control begins at block 700. Control then continues to

block 705 where the controller 156 receives an event from the application 170 at a client computer system 132. The event may be implemented as a message, a request, a packet, or any other appropriate technique. Control then continues to block 710 where the controller 156 determines whether the received event indicates that a new player desires to play a game.

If the determination at block 710 is true, then the received event indicates that a new player desires to play a game, so control continues to block 715 where the controller 156 processes the new player event, as further described below with reference to Fig. 8. Control then returns to block 705 where the controller 156 receives the next event, as previously described above.

If the determination at block 710 is false, then the received event does not indicate that a new player desires to play a game, so control continues to block 720 where the controller 156 determines whether the received event indicates that a departing player requests to leave, exit, quit, or stop playing a game before the game has completed. In various embodiments, a game is complete when one of the players wins the game, when one of the players in the game achieves the ultimate goal, objective, or level of the game, or when further manipulation or control of the game by the players is no longer possible via the game input controls.

If the determination at block 720 is true, then the received event indicates that a departing player requests to leave a game before the game has completed, so control continues to block 725 where the controller 156 processes the event of the departing player requesting to leave the game before the game is completed, as further described below with reference to Fig. 9. Control then returns to block 705 where the controller 156 receives the next event, as previously described above.

If the determination at block 720 is false, then the received event does not indicate that a departing player requests to leave a game before the game has completed, so control continues to block 730 where the controller 156 determines whether the received event is a game control input event received from a player of a game via the input device 172 and the

application 170. Game control input events are the mechanism via which the players play, control, manipulate, and give input to the game. Examples of game control input events include requests to move an object or character or any portion, change the field of view, run a play, display inventory, request information, converse with characters, or any other
5 appropriate action. The game control input event may be from any player in the game, such as a player that was a remaining player after a departing player left the game prior to completion or a waiting player that replaced the departing player and is now playing the game in place of the departing player.

10 If the determination at block 730 is true, then the received event is a game control input event received from a player of a game, so control continues to block 735 where the controller 156 updates the active player data 158, the game state data 162, and the play state data 160 based on and in response to the game control input event. For example, if the game control input event requested that the location 325 of the player (or an object or character
15 controlled by the player) be moved, then the controller 156 modifies the location 325 in the play state data 160 to reflect the new location 325. If the game control event requested an action that affected the inventory 330-1, 330-2, 330-3, or 330-4 (of any player), then the controller 156 modifies the inventory 330-1, 330-2, 330-3, or 330-4 (of any player). If the game control event requested an action that affected the score 335 of any player, then the
20 controller 156 modifies the score 335 of any player.

The controller 156 further sends the game state data 162 and the play state data 160 to each of the players at each of the client computer systems 132 via the respective applications 170, and the applications 170 interpret the game state data 162 and the play state data 160, and in
25 response, display or present appropriate portions of the game state data 162 and the play state data 160 via the respective output device 174. Control then returns to block 705 where the controller 156 receives the next event, as previously described above.

If the determination at block 730 is false, then the event is not a game control input event, so control continues to block 740 where the controller 156 processes other events. Control then
30 returns to block 705 where the controller 156 receives the next event, as previously described above.

Fig. 8 depicts a flowchart of example processing for handling a new player event, according to an embodiment of the invention. Control begins at block 800. Control then continues to block 805 where the controller 156 receives the skill level, play style, time preference, game stage preference, and/or input criteria from the new player or optionally from any saved
5 profile for the new player that may exist. The new player is the player identified by the new player event as desiring to play a game, which is the stimulus that caused the logic of Fig. 8 to be invoked, as previously described above with reference to blocks 710 and 715 of Fig. 7.

Control then continues to block 810 where the controller 156 determines whether the wait
10 queue 164 has a record for any waiting player with a skill level that matches, is the same as, or is within a skill threshold of the skill level of the new player. If the determination at block 810 is true, then a waiting player (not currently playing the game) exists who has a skill level that matches, is the same as, or is within a skill threshold of the skill level of the new player, so control continues to block 815 where the controller 156 initializes the game state data 162
15 for the new player and the waiting player, removes the waiting player from the wait queue 164, adds the new player and the waiting player to the active player data 158, initializes respective play state data 160 for the new player and the waiting player, and sends the respective play state data 160 and the game state data 162 to the respective applications 170 at the respective client computer systems 132 for the new player and the waiting player.
20 Control then continues to block 899 where the logic of Fig. 8 returns.

If the determination at block 810 is false, then a waiting player does not exist who has a skill level that matches, is the same as, or is within a skill threshold of the skill level of the new player, so control continues to block 820 where the controller 156 determines whether the
25 active player data 158 has a record for an active player with a skill level 230 that matches, is the same as, or is within a skill threshold of the skill level of the new player and the active player is playing an AI opponent. If the determination at block 820 is true, then the active player data 158 has a record for an active player with a skill level 230 that matches, is the same as, or is within a skill threshold of the skill level of the new player and the active
30 player is playing an AI opponent, so control continues to block 825 where the controller 156 replaces the AI opponent player with the new player in the active player data 158, the play state data 160, and the game state data 162. The controller 156 then sends the play state data

160 and the game state data 162 to the applications 170 of the new player and the active player at the respective client computer systems 132. Control then continues to block 899 where the logic of Fig. 8 returns.

5 If the determination of block 820 is false, then the active player data 158 does not have a record for an active player with a skill level 230 that matches (is the same as or is within a threshold) the skill level of the new player or the active player is not playing an AI opponent, so control continues to block 830 where the controller 156 adds the new player to a new record in the wait queue 164, including setting the player identifier 625, the game identifier
10 630, the skill level 635 of the new player, the play style 640 of the new player, the time preference 645 of the new player, and the game stage preference 650 of the new player. The new player is now a waiting player. Control then continues to block 899 where the logic of Fig. 8 returns.

15 Fig. 9 depicts a flowchart of example processing for handling a departing player that leaves a game before the game is complete, according to an embodiment of the invention. Control begins at block 900. Control then continues to block 905 where the controller 156 determines whether the input criteria 245 of the departing player (or optionally the input criteria 245 of the remaining player) chooses a waiting player (or a set of possible waiting
20 players) as a candidate or candidates to replace the departing player, such as by specifying a minimum or maximum skill level, designating a specific waiting player, or designating a particular play style.

If the determination at block 905 is true, then the input criteria 245 of the departing player
25 (or optionally the input criteria 245 of the remaining player) chooses a waiting player (or set of possible waiting players), so control continues to block 907 where the controller 156 selects a waiting player from the wait queue 164 whose skill level, player identifier, or play style meets the input criteria 245. Control then continues to block 910 where the controller 156 sends the game state data 162, the play state data 160, and the input information 250 (if
30 any) of the departing player to the selected waiting player via network 130. The controller 156 may optionally allow the waiting player to view the departing player and the remaining player play the game for a time period. The controller 156 may further optionally send

information (e.g., chat data) between the departing player and the waiting player for a time period.

5 Control then continues to block 915 where the controller 156 determines whether the waiting player consents to join the game as a replacement for the departing player. If the determination at block 915 is true, then the waiting player consents to join the game as a replacement for the departing player, so control continues to block 920 where the controller 156 processes the replacement of the departing player with the waiting player, as further described below with reference to Fig. 10. Control then continues to block 999 where the
10 logic of Fig. 9 returns.

If the determination at block 915 is false, then the waiting player does not consent to join the game as a replacement player for the departing player, so control continues to block 925 where the controller 156 removes the waiting player from consideration as a replacement
15 player for the departing player in the game. Control then continues to block 930 where the controller 156 determines whether any player in the wait queue 164 still exists that has not been removed from consideration (by the action of block 925, as previously described above). If the determination at block 930 is true, then a player in the wait queue 164 still exists that has not been removed from consideration, so control returns to block 905, as
20 previously described above.

If the determination at block 930 is false, then all waiting players in the wait queue 164 have been considered and no suitable waiting players who consent to replace the departing player in the game have been found, so control continues to block 940 where the controller 156
25 offers an AI opponent as a replacement player to the remaining player, or optionally saves the game state data 162, the play state data 160 and moves the remaining player from the active player data 158 to the wait queue 164, based on the preference of the remaining player. Control then continues to block 999 where the logic of Fig. 9 returns.

30 If the determination at block 905 is false, then the input criteria 245 of the departing player (or optionally the input criteria 245 of the remaining player) does not choose a waiting player (or set of possible waiting players), so control continues to block 935 where the

controller 156 selects the waiting player from the wait queue 164 whose skill level 635 and play style 640 best matches (within a threshold) the skill level 230 and play style 235 of the departing player (or whose skill level 635 and play style 640 best matches the input criteria 245 of the remaining player), whose time preference 645 best matches (within a time
5 threshold) the time remaining in the game state 525 of the game state data 162 before the game is complete, and/or whose game stage preference 650 best matches the stage in the game state 525 of the game state data 162. Control then continues to block 910, as previously described above.

10 Fig. 10 depicts a flowchart of example processing for replacing a departing player, according to an embodiment of the invention. Control begins at block 1000.

Control then continues to block 1005 where the controller 156 removes the departing player from the active player data 158, from the play state data 160, and from the game state data
15 162. Control then continues to block 1010 where the controller 156 adds the waiting player to the play state data 160 and the game state data 162. Control then continues to block 1015 where the controller 156 assigns the play state 320 of the departing player to the play state 320 of the waiting player, including transferring player state information such as the inventory 330-1, 330-2, 330-3, or 330-4, the score 335, and the location 325 of the departing
20 player to client computer system of the waiting player. Control then continues to block 1020 where the controller 156 moves the waiting player from the wait queue 164 to the active player data 158 (deletes the waiting player from the wait queue 164 and adds the waiting player to the active player data 158). The waiting player is now an active player in the game and is a replacement player for the departing player. Control then continues to block 1025
25 where the controller 156 sends the play state data 160 and the game state data 162 to the players (the remaining player and the waiting player who is now the replacement player) via the respective applications 170 of the players at their respective client computer systems 132. The waiting player (now the replacement player) may now send game control input events, as previously described above with reference to Fig. 7. Control then continues to
30 block 1099 where the logic of Fig. 10 returns.

Fig. 11 depicts a flowchart of example processing for optionally calculating a play style of a player, according to an embodiment of the invention. The logic of Fig. 11 may be invoked periodically, invoked at the end of a game for the players in the game, invoked in response to a player departing a game, invoked in response to a player joining a game, invoked if a
5 player does not self report a play style, invoked after a player has reached a threshold number of games, game control input events, scores 335, or levels within a game.

Control begins at block 1100. Control then continues to block 1105 where the controller 156
10 optionally categorizes the game control input events during a period of time for a player based on the type of the game control input events and the number of each type of the game control input events. For example, for a football game, the controller 156 may categorize the game control input events for a player and by counting (the number) of passing plays (the type of the game control input event) by counting (the number) of running plays (the type) that the player has requested via a game control input event.

15 Control then continues to block 1110 where the controller 156 optionally calculates the play style of the player based on the type of the game control input events or the set of the types of the game control input events that are the most highest in number, indicating that they are the most frequently entered game control input events.

20 For example, if the type of game control input events with the highest number is more than a threshold greater than the type of game control input events with the next-highest number, the controller 156 may calculate the play style as the type of the game control input event that has highest number. For example, if the player has input two running plays and ten pass
25 plays, the controller 156 may calculate pass as the play style. If the type of game control input events with the highest numbers are within a threshold of each other, then the controller 156 may calculate the play style as a mixture of the two types of game control input events. For example, if the player uses 52% pass plays and 48% running plays, the controller 156 may calculate the play style as a mixed style of run and pass.

30 Control then continues to block 1115 where the controller 156 optionally categorizes the inventory 330-1, 330-2, 330-3, or 330-4 of the player by type and number of objects in the

player's inventory. For example, the controller 156 may count the number of attack points, defend points, wisdom points, or widgets (types of objects) in the inventory 330-1, 330-2, 330-3, or 330-4. Control then continues to block 1120 where the controller 156 optionally calculates the play style of the player based on the type or types of the objects in the
5 inventory 330-1, 330-2, 330-3, or 330-4 that have the highest number. For example, if the type of object in the player's inventory 330-1, 330-2, 330-3, or 330-4 with the highest number is attack points, the controller 156 may calculate the play style of the player as an aggressive player.

10 Control then continues to block 1125 where the controller 156 optionally categorizes the location 325 of the player or the location 325 of the inventory 330-1, 330-2, 330-3, or 330-4 of the player by time (categorizing the locations 325 by most recent to least recent time of occupancy) and number (categorizing the locations 325 by frequency of occupation). For example, the controller 156 may count the number of times that a player or the inventory
15 330-1, 330-2, 330-3, or 330-4 of a player (or the player's inventory) has occupied at each location 325 within a time period and may record the times that the player (or the player's inventory) has occupied each location 325. Control then continues to block 1130 where the controller 156 optionally calculates the play style of the player based on the categorization of locations 325 that are the most recent in time or the most frequent (highest) in number. For
20 example, if the inventory 330-1, 330-2, 330-3, or 330-4 of a player most frequently occupies the player's home location 325, the controller 156 may calculate the player's play style as a settler or a farmer. But, if the inventory of a player most frequently occupies a variety of locations 325, the controller 156 may calculate the player's play style as an explorer.

25 Control then continues to block 1135 where the controller 156 optionally calculates the play style of the player based on a combination of the categorization of game control input events, inventory 330-1, 330-2, 330-3, or 330-4, and the location 325. For example, if the controller 156 determines that the player has number of wisdom points (inventory) greater than a threshold, has occupied a number of locations 325 greater than a threshold, and
30 requests a number of game control input events greater than a threshold that have a type of a request to sell objects in the player's inventory, the controller 156 may calculate the play style of the player as a trader.

Control then continues to block 1199 where the logic of Fig. 11 returns.

In the previous detailed description of exemplary embodiments of the invention, reference was made to the accompanying drawings (where like numbers represent like elements),
5 which form a part hereof, and in which is shown by way of illustration specific exemplary embodiments in which the invention may be practiced. These embodiments were described in sufficient detail to enable those skilled in the art to practice the invention, but other embodiments may be utilized and logical, mechanical, electrical, and other changes may be made without departing from the scope of the present invention. For example, whilst some
10 described embodiments refer two players participating in a game, there may be any number of players taking part from a plurality of client computer systems. In the previous description, numerous specific details were set forth to provide a thorough understanding of embodiments of the invention. But, the invention may be practiced without these specific details. In other instances, well-known circuits, structures, and techniques have not been
15 shown in detail in order not to obscure the invention.

Different instances of the word “embodiment” as used within this specification do not necessarily refer to the same embodiment, but they may. Any data and data structures illustrated or described herein are examples only, and in other embodiments, different
20 amounts of data, types of data, fields, numbers and types of fields, field names, numbers and types of rows, records, entries, or organizations of data may be used. In addition, any data may be combined with logic, so that a separate data structure is not necessary. The previous detailed description is, therefore, not to be taken in a limiting sense, and the scope of the present invention is defined only by the appended claims.

The scope of the present disclosure includes any novel feature or combination of features disclosed herein. The applicant hereby gives notice that new claims may be formulated to such features or combination of features during prosecution of this application or of any such
25 further applications derived therefrom. In particular, with reference to the appended claims, features from dependent claims may be combined with those of the independent claims and
30 features from respective independent claims may be combined in any appropriate manner and not merely in the specific combinations enumerated in the claims.

For the avoidance of doubt, the term “comprising”, as used herein throughout the description and claims is not to be construed as meaning “consisting only of”.

CLAIMS

1. A method of replacing a player in a multi-player network game, the method comprising:

5 determining that a departing player requests to leave a game before the game is complete;

selecting a waiting player;

replacing the departing player with the waiting player in the game, wherein the replacing further comprises assigning a play state of the departing player to the waiting
10 player and transferring play state data to computer system of the waiting player; and

updating the play state in response to a game control input event received from the waiting player after the replacing.

2. The method of claim 1, wherein the selecting the waiting player further comprises:

15 determining whether the waiting player has a skill level that is within a skill threshold of the skill level of the departing player.

3. The method of claim 1 or 2, wherein the selecting the waiting player further comprises:

20 selecting the waiting player that has a play style that matches the play style of the departing player.

4. The method of claim 1, 2 or 3, wherein the selecting the waiting player further comprises:

25 determining whether the waiting player has a time preference that is within a time threshold of a time period that remains before the game is complete.

5. The method of any preceding claim wherein the selecting the waiting player further comprises:

30 determining whether the waiting player has a game stage preference that matches a stage of the game.

6. The method of any preceding claim, wherein the selecting the waiting player further comprises:

determining whether the waiting player meets an input criteria specified by the departing player.

5

7. The method of any preceding claim, wherein the selecting the waiting player further comprises:

determining whether the waiting player meets an input criteria specified by a remaining player in the game.

10

8. The method of any preceding claim, wherein the assigning the play state further comprises:

assigning a location within the game of the departing player to the waiting player.

15

9. The method of any preceding claim, wherein the assigning the play state further comprises:

assigning an inventory of the departing player to the waiting player.

20

10. The method of any preceding claim, wherein the assigning the play state further comprises:

assigning a score of the departing player to the waiting player.

25

11. A signal-bearing medium encoded with instructions, wherein the instructions when executed comprise a method of replacing a player in a multi-player computer game comprising:

determining that a departing player requests to leave a game before the game is complete;

selecting a waiting player;

replacing the departing player with the waiting player in the game, wherein the replacing further comprises assigning a play state of the departing player to the waiting player, and wherein the assigning the play state further comprises assigning a location, an inventory, and a score of the departing player to the waiting player; and

30

updating the play state in response to a game control input event received from the waiting player.

5 12. The signal-bearing medium of claim 11, further comprising:
sending information supplied by the departing player to the waiting player.

13. The signal-bearing medium of claim 11 or 12, further comprising:
allowing the waiting player to view game play between a remaining player and the departing player for a time period after the selecting and before the replacing.

10

14. The signal-bearing medium of claim 11, 12 or 13, wherein the replacing the departing player with the waiting player further comprises:
replacing the departing player with the waiting player if the waiting player consents.

15

15. The signal-bearing medium of any of claims 11 to 14, wherein the selecting the waiting player further comprises:
selecting the waiting player that has a play style that matches the play style of the departing player.

20

16. The signal-bearing medium of claim 15, further comprising:
calculating the play style, wherein the calculating further comprises categorizing the game control input events by type and number.

25

17. The signal-bearing medium of claim 15 or 16, further comprising:
calculating the play style, wherein the calculating further comprises categorizing the inventory by type and number.

30

18. The signal-bearing medium of claim 15, 16 or 17, further comprising:
calculating the play style, wherein the calculating further comprises categorizing the location by time and number.

19. An apparatus for controlling a computer game played over a network of client computer systems by a plurality of players, the apparatus comprising:

a memory for storing data including play stat data for each active player;

5 a controller for receiving events from players and configured to update the play state in response to a game control input event received from the waiting player determine that a departing player requests to leave a game before the game is complete;

to select a waiting player;

10 replace the departing player with the waiting player in the game, comprising assigning a play state of the departing player to the waiting player, and sending the play state to the client computer system of the waiting player; and

update the play state in response to a game control input event received from the waiting player.

20. An apparatus according to claim 19, wherein the assigning the play state further comprises assigning a location, an inventory, and a score of the departing player to the waiting player.

21. The apparatus of claim 19, or 20, wherein the controller is further configured to determine whether the waiting player meets an input criteria specified by the departing player.

22. The apparatus of claim 19, 20 or 21, wherein the controller is further configured to determine whether the waiting player meets an input criteria specified by a remaining player in the game.

23. The apparatus of any of claims 19 to 22, wherein the controller is further configured to send information supplied by the departing player to the waiting player.

24. An apparatus according to any of claims 19 to 23, wherein the controller is configured to allow the waiting player to view game play between a remaining player and the departing player for a time period after the select and before the replace.

25. An apparatus according to any of claims 19 to 24, wherein the controller is configured to:

receive a new player; and

5 to replace the waiting player with the new player if the waiting player is an artificial intelligence player and the skill level of the new player is within the skill threshold of a remaining player.

1/11

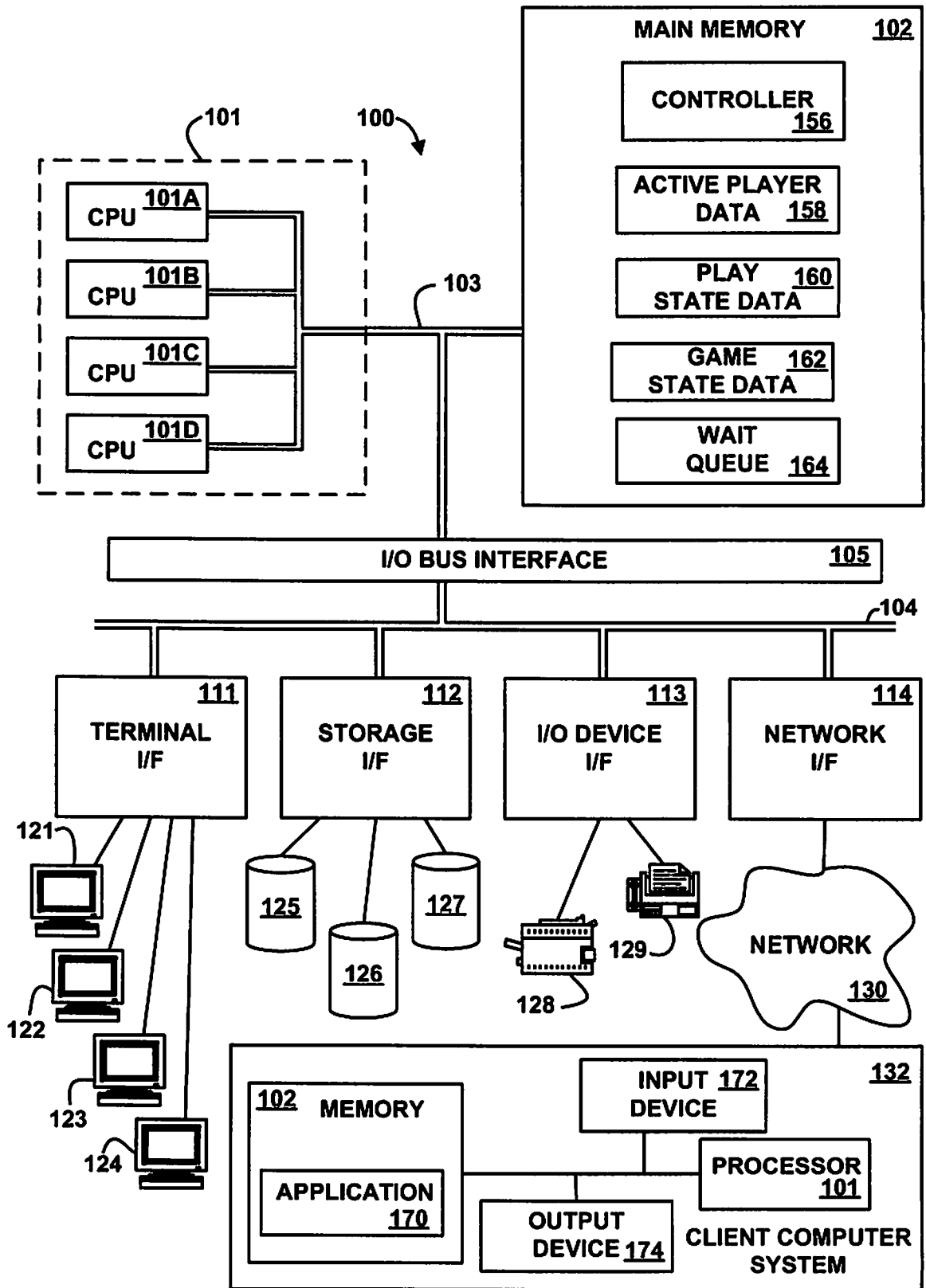


FIG. 1

2/11

ACTIVE PLAYER DATA						158
225 PLAYER ID	230 SKILL LEVEL	235 PLAY STYLE	240 TIME PREF	245 INPUT CRITERIA	250 INPUT INFO	
USER A	84	RUN	2 HOURS	MINIMUM SKILL 75	USE ZONE DEFENSE ON THIRD DOWN	205
USER B	79	PASS	1 HOUR			210
USER C	92	AGGRESSIVE	1.5 HOURS	CHOOSE USER E	WHEN HAVE 100 WIDGETS MAKE MEGA-WEAPON AND GIVE TO MAN IN WOODS	215
USER D	79	DIPLOMATIC	50%	PARTISAN STYLE		220

FIG. 2

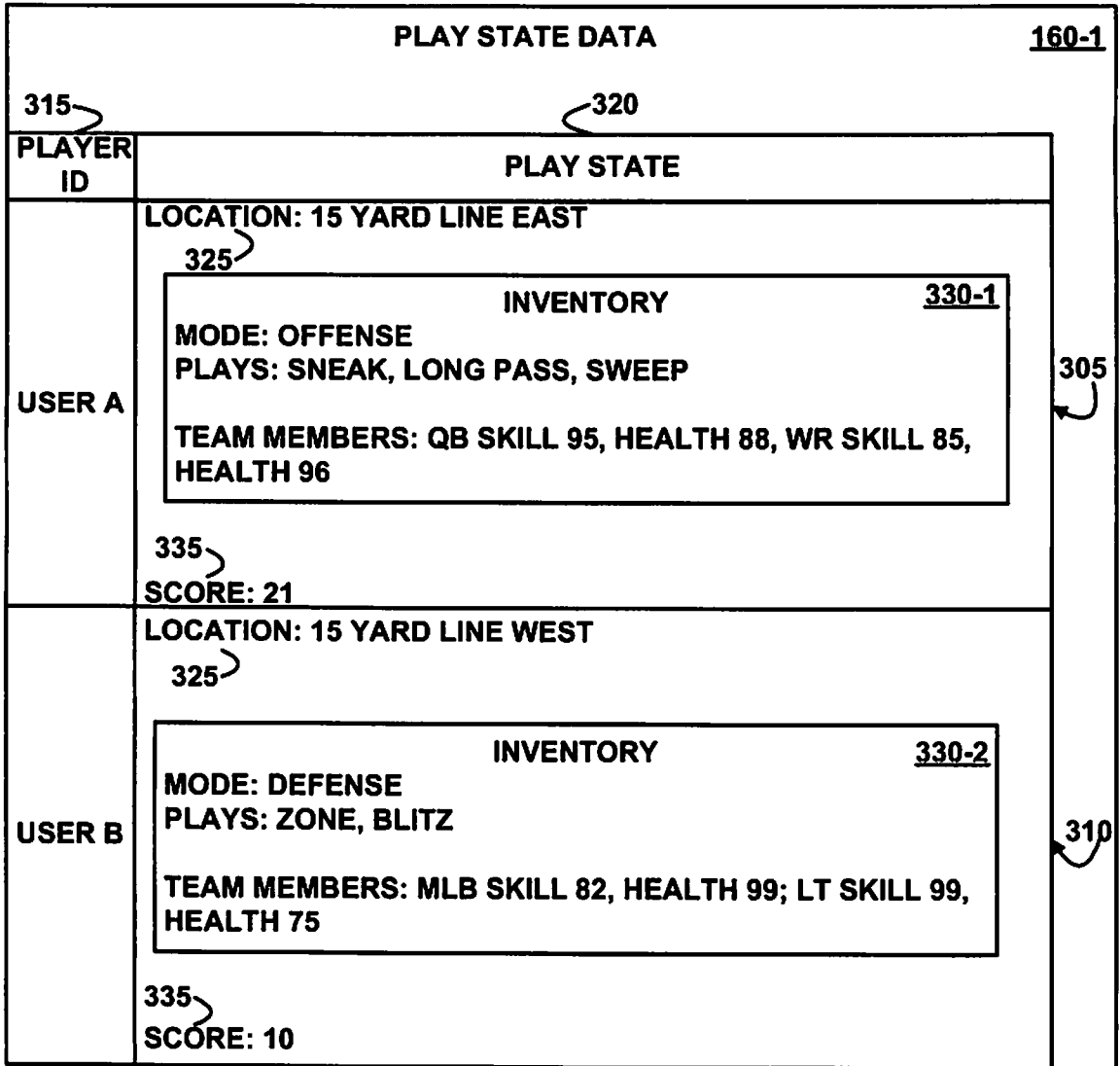


FIG. 3

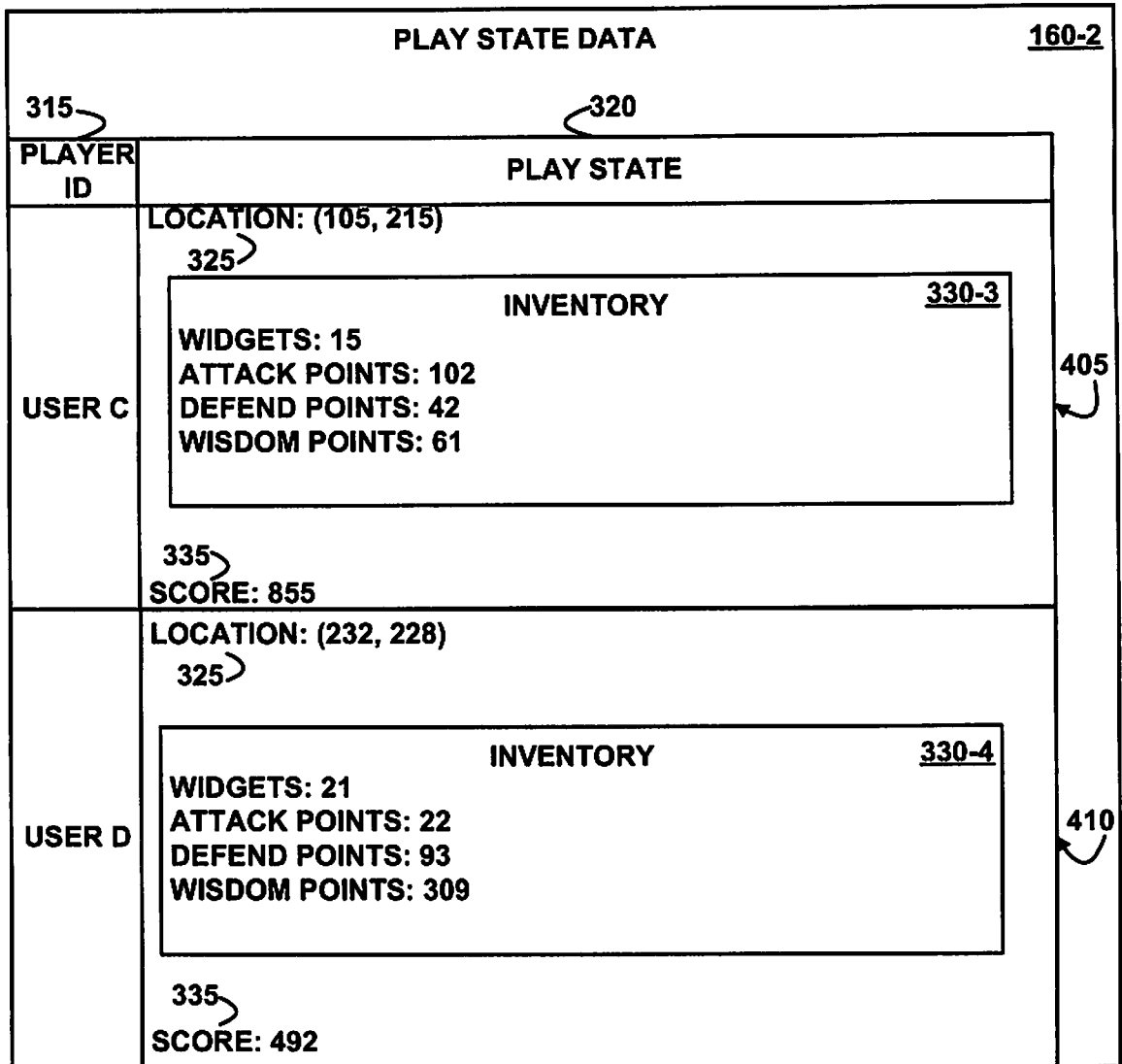


FIG. 4

GAME STATE DATA			162
515 GAME ID	520 PLAYERS	525 GAME STATE	
GAME X	USER A, USER B	STAGE: FIRST DOWN, THIRD QUARTER, 2:15 ENVIRONMENT: FIELD MUDDY, STADIUM FULL, CROWD CHEERING 50%	505
GAME Y	USER C, USER D	STAGE: RESEARCH ENVIRONMENT: WOODS AT (45, 90), WAREHOUSE ON FIRE AT (108, 860), RIVER AT (192, 333), MAN AT (45, 90), ENERGY PACK AT (22, 215)	510

FIG. 5

6/11

WAIT QUEUE						164
625 PLAY ID	630 GAME ID	635 SKILL LEVEL	640 PLAY STYLE	645 TIME PREFERENCE	650 GAME STAGE PREFERENCE	
USER E	GAME X	75	BALANCED	.5 HOUR	THIRD QUARTER	605
USER F	GAME X	40	PASS	2 HOURS	SECOND QUARTER	610
USER G	GAME Y	88	PARTISAN	1.5 HOURS	DEVELOPMENT	615
USER H	GAME Y	67	AGGRESSIVE	2.5 HOURS	BEGINNING ONLY	620

FIG. 6

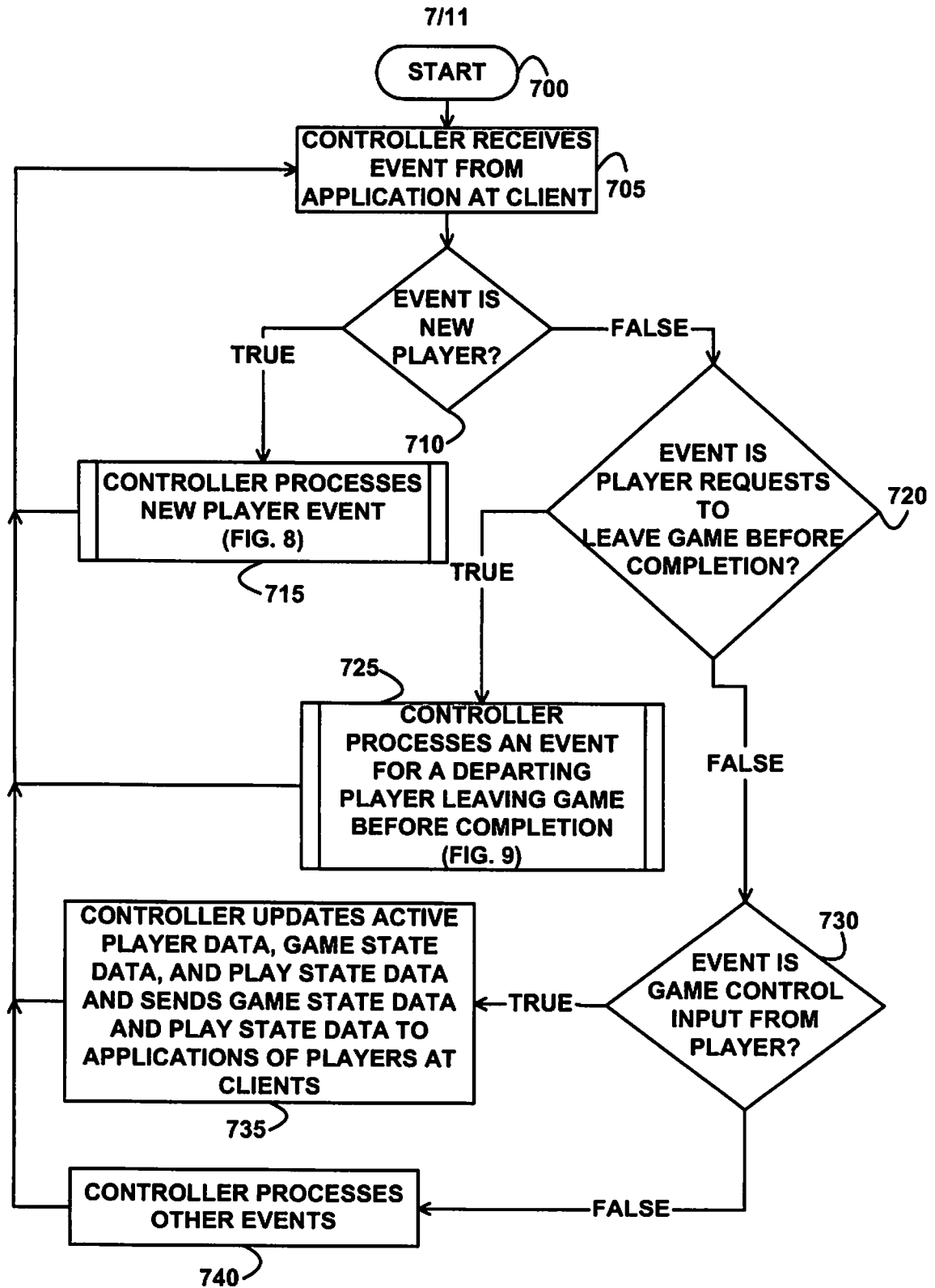


FIG. 7

8/11

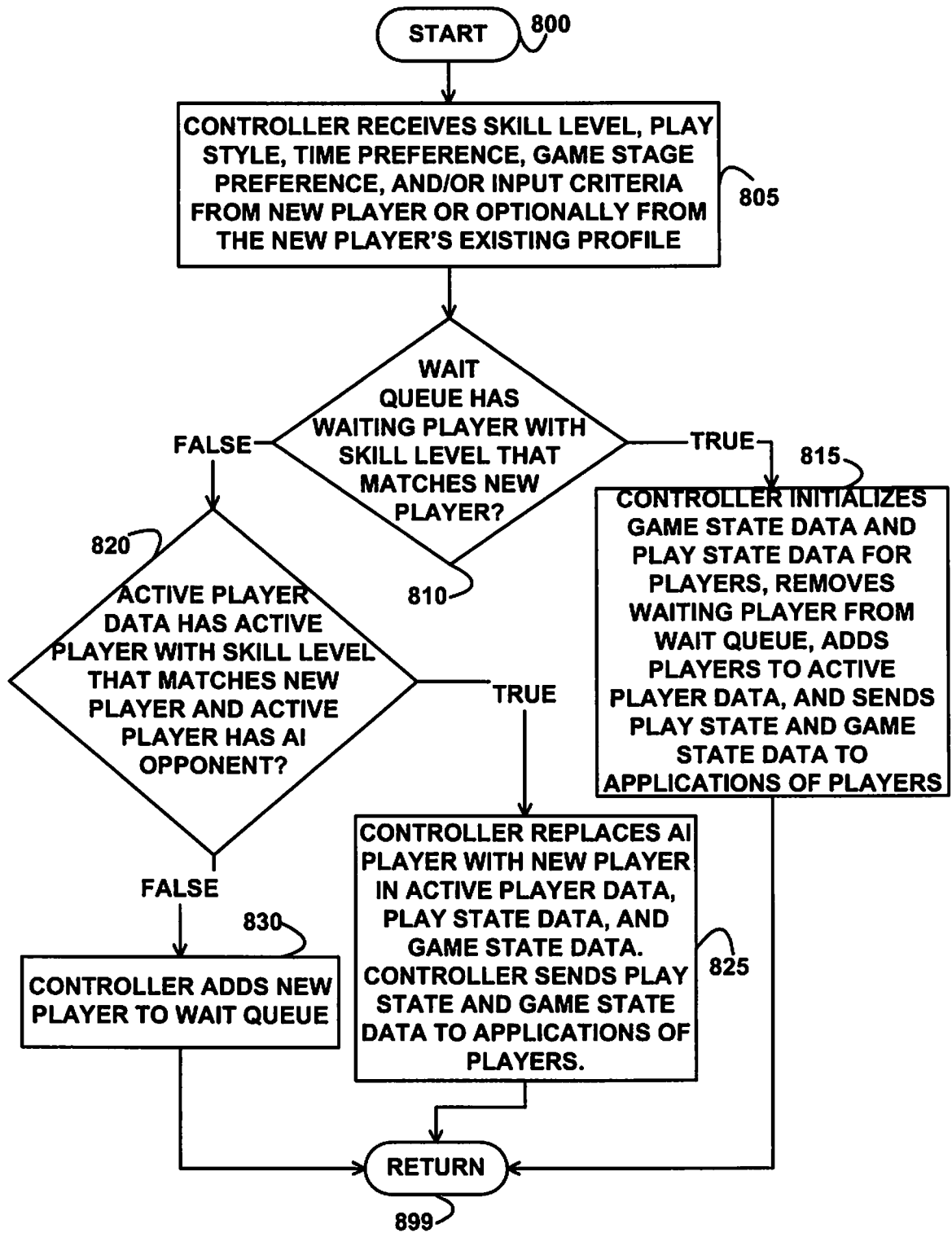


FIG. 8

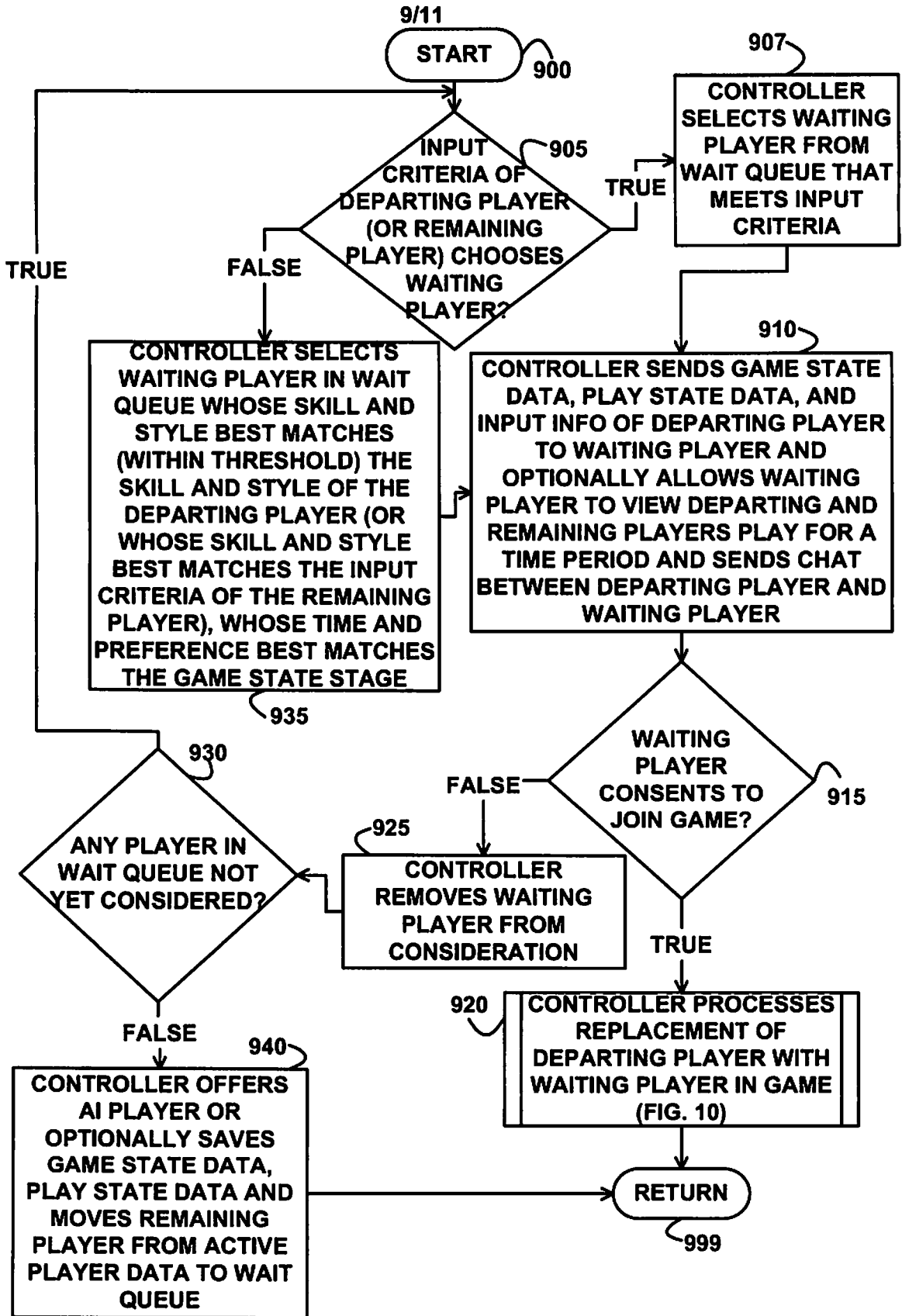


FIG. 9

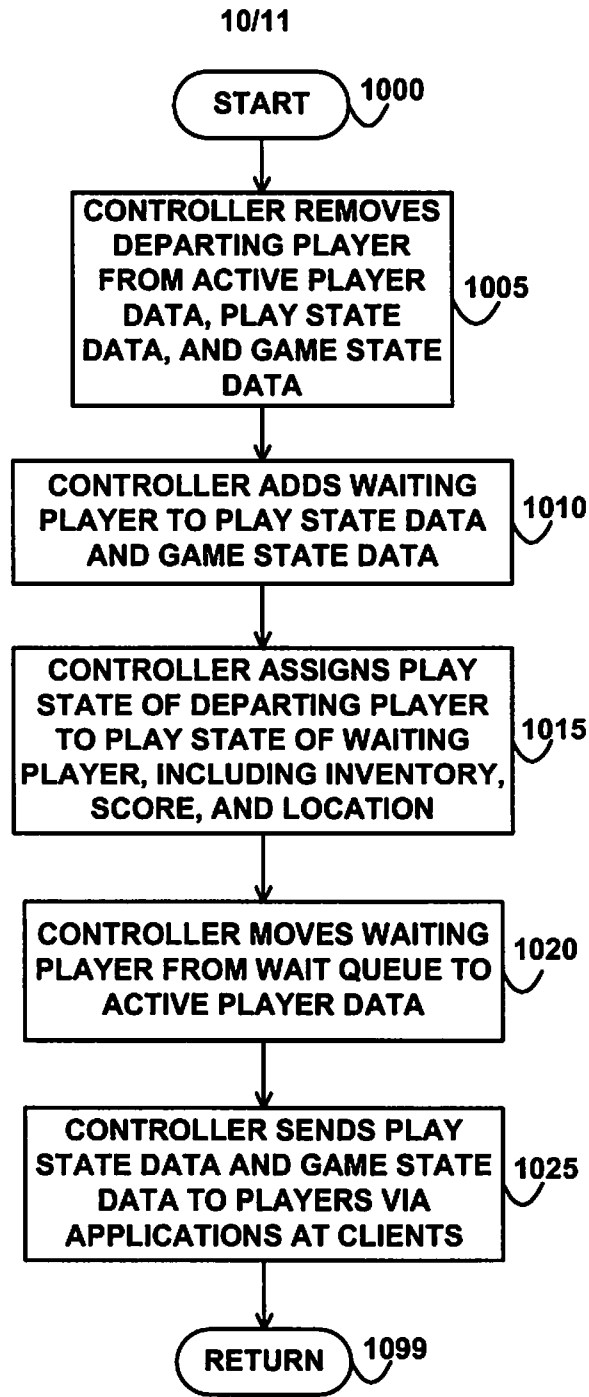


FIG. 10

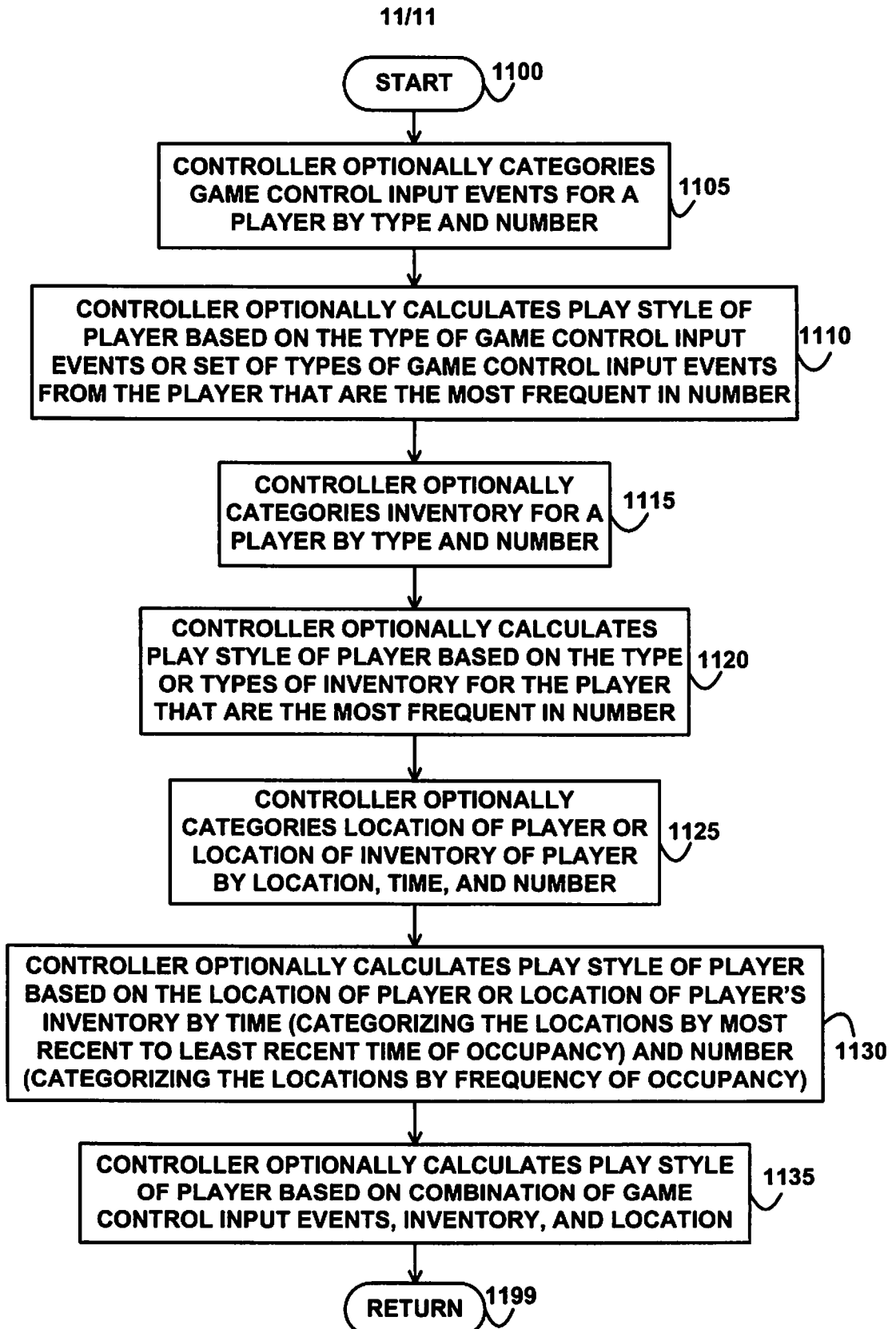


FIG. 11

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2007/058026

A. CLASSIFICATION OF SUBJECT MATTER
INV. A63F13/12 H04L29/06 H04L29/08

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
A63F H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 1 656 976 A (SEGA DOING BUSINESS AS SEGA CO [JP]) 17 May 2006 (2006-05-17) figures 1,2,5 abstract paragraphs [0016], [0017] paragraph [0019] paragraphs [0030], [0031] paragraphs [0035] - [0037] paragraphs [0040] - [0047] ----- -/--	1-25

Further documents are listed in the continuation of Box C.

See patent family annex.

* Special categories of cited documents :

- *A* document defining the general state of the art which is not considered to be of particular relevance
- *E* earlier document but published on or after the international filing date
- *L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- *O* document referring to an oral disclosure, use, exhibition or other means
- *P* document published prior to the international filing date but later than the priority date claimed

- *T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- *X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- *Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
- *Z* document member of the same patent family

Date of the actual completion of the international search

28 November 2007

Date of mailing of the international search report

07/12/2007

Name and mailing address of the ISA/
 European Patent Office, P.B. 5818 Patentlaan 2
 NL - 2280 HV Rijswijk
 Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
 Fax: (+31-70) 340-3016

Authorized officer

Jeampierre, Gérald

INTERNATIONAL SEARCH REPORT

International application No

PCT/EP2007/058026

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	<p>WO 03/053531 A (MAJOREM LTD [IL]; AMIR GIDEON [IL]; AXELROD RAMON [IL]; MENDELSON YUVA) 3 July 2003 (2003-07-03) figures 9-11 abstract page 2, lines 6-17 page 3, lines 7-11,16,17 page 4, line 24 - page 5, line 2 page 5, lines 4-16 page 19, line 30 - page 20, line 25 page 20, line 31 - page 21, line 6 page 21, lines 9-23</p> <p style="text-align: center;">-----</p>	1-25
X	<p>DATABASE WPI Week 200281 Derwent Publications Ltd., London, GB; AN 2002-746307 XP002460191 -& JP 2002 306851 A (NAMCO LTD) 22 October 2002 (2002-10-22) An automatic computer translation of this Japanese publication can be found on the Web site of the Japanese Patent Office: http://www.ipdl.ncipi.go.jp/homepg_e.ipdl abstract figure 1 paragraphs [0003], [0004] paragraphs [0026] - [0028] paragraphs [0033], [0034] paragraphs [0036] - [0042] paragraph [0046] paragraph [0050] paragraph [0053] paragraph [0056] paragraphs [0060] - [0069]</p> <p style="text-align: center;">-----</p>	1-25

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No PCT/EP2007/058026

Patent document cited in search report		Publication date	Patent family member(s)	Publication date
EP 1656976	A	17-05-2006	CN 1773518 A JP 2006136350 A KR 20060052380 A US 2006100020 A1	17-05-2006 01-06-2006 19-05-2006 11-05-2006
WO 03053531	A	03-07-2003	AU 2002358959 A1	09-07-2003
JP 2002306851	A	22-10-2002	NONE	