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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
20 August 2009 (20.08.2009)

PCT

(10) International Publication Number
WO 2009/102956 A2

(51) International Patent Classification:
A63F 9/24 (2006.01) *G06F 19/00* (2006.01)

(21) International Application Number:
PCT/US2009/034054

(22) International Filing Date:
13 February 2009 (13.02.2009)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
61/065,975 15 February 2008 (15.02.2008) US
12/194,332 19 August 2008 (19.08.2008) US

(71) Applicant *(for all designated States except US)*: **GTECH RHODE ISLAND CORPORATION** [US/US]; 10 Memorial Blvd., Providence, RI 02903 (US).

(72) Inventor; and

(75) Inventor/Applicant *(for US only)*: **RYBAK, Ihor, Bohdan** [CA/US]; 93 Fieldstone Lane, Saunderstown, RI 02874 (US).

(74) Agent: **REIBMAN, Andrew, L.**; K & L Gates LLP, 599 Lexington Avenue, New York, NY 10022-6030 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, 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, ST, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

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

Published:

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

WO 2009/102956 A2

(54) Title: METHODS AND SYSTEMS FOR LICENSE SHARING AMONG GAMING TERMINALS

(57) Abstract: Systems and methods for sharing wagering game licenses among multi- state terminals are provided. Each example terminal may include a token-enabled state and a non-token- enabled state. Each token may be associated with a license for the operation of a wagering game. A token pool may be administered by a host. A terminal at which a customer has indicated a desire to play a wagering game may request a token from the token pool administrating host. The request may be queued if a token is not available, or the requesting terminal may be issued a token. Thus, an unlimited number of multi-state terminals may be operated, while no more than a predetermined quantity of the terminals may be providing a wagering game at any one time.

METHODS AND SYSTEMS FOR LICENSE SHARING AMONG GAMING TERMINALS

CROSS-REFERENCES TO RELATED APPLICATIONS

This application claims the benefit of United States Patent Application No. 61/065,975, filed February 15, 2008 and United States Patent Application No. 12/194,332, filed August 19, 2008. The disclosure of each of said Applications is incorporated by reference herein.

FIELD OF THE INVENTION

The present application is generally related to gaming apparatus that require a license, such as a game terminal, video slot machine, or video poker machine, although it may also be applicable to other computer-based games where licenses are required.

BACKGROUND INFORMATION

Electronic wagering machines such as conventional slot machines, video slot machines, or video poker machines have long been a staple for the gaming industry. However, in many jurisdictions the number of gaming machines allowed in a particular location or that may be operated by a particular operator may be set by law, regulation, or license. The level is often set below the level at which operator profit is maximized. In other words, the number of machines that would be operated based on consumer demand and market conditions typically exceeds the legally allowed gaming machine supply in some jurisdictions.

SUMMARY

One example embodiment of the present invention is a system, including a plurality of wagering game terminals, each having a token-enabled state and a non-enabled state. The system has a plurality of tokens associated with a capability, and the number of tokens is less than the number of terminals. Also, each of the plurality of tokens may be configured, when held by one of the terminals, to allow the terminals to operate in the token-enabled state. The plurality of terminals are each configured to enter the token-enabled state only when a token is received by the terminal and to return to the non-enabled state when the token is surrendered by the terminal. When a terminal is in the token-enabled state, the terminal provides at least one wagering game to a player. When the terminal in the non-enabled state, the terminal is prevented from offering wagering games.

Optionally, several other features may be provided, alone or in combination with each other, in the example system. The token-enabled state may include a plurality of game formats, wherein the terminal may be configured to provide a selection of one or more of the game formats, and the selection may be based, at least in part, on instructions associated with the token. Optionally, the instructions associated with the token may be based, at least in part, on the time of day. Optionally, the return to the non-enabled state may be caused by a user exiting the token-enabled state. Optionally, return to the non-enabled state is caused by the terminal being in the token-enabled state for a period of time.

Optionally, the terminals may be configured to log data including at least one of: revenue generated by a terminal in the token-enabled state, number of terminals in the token-enabled state, and number of token requests not immediately fulfilled. The plurality of terminals may be configured to communicate with each other, and the surrendering of a token may include the terminals being configured to communicate the availability of a token to each other. The terminals may be configured to queue when token requests exceed token availability. The order of terminals in the queue may be prioritized based, at least in part, on at least one of: the amount of time the terminal has been in the queue, the location of the terminal, the amount of time the terminal has already been in the token-enabled state during a time period, the operator of the terminal, or the profitability of the terminal.

The wagering game terminal may include at least one of: video slots, video poker, keno, bingo, lottery, conventional mechanical slots, blackjack, craps or roulette.

Another example embodiment of the present invention is a method. The example method includes being prevented from providing a wagering game when in a non-enabled state, requesting a token associated with a capability, entering a token-enabled state (responsive to the receipt of a token), providing at least one wagering game while in the enabled state, returning to the non-enabled state, and finally, surrendering the token.

Optionally, several other features may be provided, alone or in combination with each other, in the example method. Entering a token-enabled state may include providing at least one wagering game. The entering a token-enabled state may include providing a plurality of game formats. Additionally, the example method may include receiving instructions associated with a token, and selecting which one or more game formats to provide, from the plurality of game formats, based, at least in part, on the instructions.

Optionally, in the example method, the instructions associated with the tokens may be based, at least in part, on the time of day. The return to the non-enabled state may be caused by a user exiting the token-enabled state. Alternatively, the return to the non-enabled state may be caused by the terminal being in the token-enabled state for a period of time. The example method may also include logging data, which may include at least one of: revenue generated while in the token-enabled state, time spent waiting for a token, or amount of time in the token-enabled state. The example method may also include communicating the availability of a token, responsive to the surrendering of the token.

Another example embodiment of the present invention is a system that includes a plurality of tokens associated with a capability. Each token may be configured to be in one of one or more states, and the states include at least an issued state and an available state. The tokens are configured to enter the issued state responsive to an entity requesting a token and that token being in the available state. Also, tokens are configured to enter the available state, responsive to being surrendered by the entity the token has been issued to. Optionally, in the example system, a token may be associated with a license to operate a wagering terminal.

Another example embodiment of the present invention is a system, including a server configured to control the allocation of a plurality of tokens associated with a capability. The server is configured to receive a request from a terminal for a token, where the number of terminals configured to issue a request for a token to the server is greater than the number of tokens the server is configured to control. Also, the server is configured to issue a token, responsive to a request for a token, if a token is available. The server is configured to queue requests for tokens, if no tokens are currently available.

Optionally, in the example system, the number of tokens the server is configured to control may correspond to a number of licenses for the operation of a wagering terminal. Optionally, in the example system, the server may be configured to control token allocation for at least one jurisdiction, and the maximum number of tokens the server may issue to terminals located within any particular jurisdiction, may depend on the number of licenses issued for that jurisdiction. Optionally, in the example system, the server may be configured to send instructions to the terminal, associated with issuing a token to that terminal.

Optionally, in the example system, the instructions may include at least one of the following: which wagering game formats will be available at the terminal, the minimum wager allowed at the terminal, the maximum wager allowed at the terminal, or the maximum amount of time the terminal will be issued the token. The server may be configured to recall an issued token. The queue of terminals, which have requested a token, may be prioritized according to one of more of the following criteria: the amount of time the terminal has been in the queue, the location of the terminal, the amount of time the terminal has already been in the token-enabled state during a time period, the operator of the terminal, or the profitability of the terminal. The server may also be configured to log data, including any one or more of the following: revenue generated by each terminal that requests a token, number of issued tokens per time period, and number of token requests not immediately fulfilled.

Another example embodiment of the present invention may include a method for controlling the allocation of a plurality of tokens associated with a capability. The example method will receive a request from a terminal for a token. The number of terminals configured to issue a request for a token will be greater than the number of tokens in the plurality of tokens. The example method will issue a token, responsive to a request for a token, if a token is available. The example method will queue requests for tokens, if no tokens are currently available.

Optionally, in the example method the number of tokens in the plurality of tokens corresponds to a number of licenses for the operation of a wagering terminal. The example method may allocate tokens for at least one jurisdiction, and the maximum number of tokens the example method will issue to terminals located within any particular jurisdiction depends on the number of licenses issued for that jurisdiction. The example method may send instructions to the terminal associated with the issuing a token.

Optionally, in the example method, the instructions may include at least one of the following: which wagering game formats will be available at the terminal, the minimum wager allowed at the terminal, the maximum wager allowed at the terminal, or the maximum amount of time the terminal will be issued the token. Optionally, the example method may recall an issued token. Optionally, the example method may prioritize the queue of terminals, which have requested a token, according to one or more of the following criteria: the amount of time the terminal has been in the queue, the location of the terminal, the amount of time the terminal has already been in the token-enabled state during a time period, the operator of the terminal, or the profitability of the terminal.

Optionally, the example method may log data, including any one or more of the following: revenue generated by each terminal that requests a token, number of issued tokens per time period, and number of token requests not immediately fulfilled.

Another example embodiment of the present invention is a method of logging the allocation of a plurality of tokens to a plurality of wagering game terminals. The method will provide an account of data associated with the allocation of terminals, including how many tokens were allocated at any particular time.

Optionally, the example method may provide a report which satisfies a reporting or compliance requirement of an entity associated with the licenses.

One example embodiment of the present invention is a system for sharing authorization tokens, which includes a plurality of terminals. Each terminal is configured in a first state. The example system has at least one server, which is configured to control the allocation of at least one token. The token may provide an authorization grant, and the number of tokens may correspond to a number of licenses for the operation of a wagering game. The server may be configured to control token allocation at least one location, and the number of tokens the server will issue to terminals located in a particular location is based on the number of licenses available for that particular location. The example system has a first terminal, which is configured to request a token, responsive to a request by a user to enter a second state. The second state includes providing a wagering game, and the first state does not include providing a wagering game. The first terminal is configured, responsive to receipt of a token, to enter a second state. The second state is configured to have at least two sub-states. The server determines which sub-states are available for the first terminal to enter. The first terminal is configured, responsive to returning to the first state, to return the token to the server. The server is configured to queue a request for a token from a second terminal, when no tokens are currently available. The server is configured to ensure that the number of terminals in the second state does not exceed the number of terminals legally allowed to operate a wagering game within the jurisdiction the terminals are located in.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 illustrates another example system, according to an example embodiment of the present invention.

Figure 2A illustrates an example procedure, according to an example embodiment of the present invention.

Figure 2B illustrates another example procedure, according to an example embodiment of the present invention.

Figure 3 illustrates another example procedure, according to an example embodiment of the present invention.

Figure 4 illustrates another example procedure, according to an example embodiment of the present invention.

Figure 5 illustrates another example procedure, according to an example embodiment of the present invention.

Figure 6 illustrates an example system, according to an example embodiment of the present invention.

Figure 7 illustrates another example system, according to an example embodiment of the present invention.

Figure 8 illustrates an example timeline of example systems, according to an example embodiment of the present invention.

Figure 9 illustrates another example procedure, according to an example embodiment of the present invention.

DETAILED DESCRIPTION OF EXAMPLE EMBODIMENTS

An underutilization of gaming licenses has been observed. By decoupling the one-to-one association of the physical operation of a gaming machine with the legal operation of a gaming machine for wagering, a higher number of gaming machines may be utilized legally, potentially increasing utilization of available licenses, and thereby increasing operator revenues and profits. For a simple illustrative example, it may be the case that only two gaming machines are allowed to operate at a location. At a location, a first machine might see a 100% utilization, a second

machine might see a 60% utilization, while a third machine, which is not permitted, might see a 40% utilization. A game machine operator would then be forced to operate only the first and second machine, as the operator only has two licenses for the operation of gaming machines. However, without a license for the third machine, the operator will experience an underutilization of their licenses, which corresponds to an unfulfilled consumer demand. Some example embodiments of the present invention seek to eliminate this economic underutilization, by disassociating the physical capacity to operate a game, and the actual operation of the game. That is, some example embodiments of the present invention would allow the game operator to operate three multi-state devices, such as wager game terminals, where the devices may be configured to operate in a wagering game mode and a non-gaming "entertainment only" or "sleep" mode. The devices may be further configured to ensure that only the legally allowed number of the devices are in the gaming mode at any one time. In this way, a certain number of licenses for the operation of a gaming machine may be used for more than that number of devices so long as only the allowed number or fewer devices are operating in gaming mode at any one time. In the example given, if the demand at the third machine exactly matched the lack of demand at the second machine and vice versa, the two licenses would be fully utilized. To the extent the demand times overlap, there may still be some underutilization. Some of that overlapping demand may be captured by queueing requests for customers willing to wait. Regardless, more customers may be served and higher license utilizations achieved as compared to the prior situation where no third machine was provided. Further example embodiments with reference to illustrative figures follow below.

Figure 1 illustrates an example system, according to an example embodiment of the present invention. A Host 100 may control the allocation of tokens from a Token Pool 112. The Host 100 may include a Server 110, which may include one or more general purpose computers in one or more locations. Host 100 may have a network I/O device to connect the Host 100 to Network 150, which may be the Internet or an intranet network. Server 110 may be connected to a Database 111, which may store a variety of information, including the Event Log 116. The Event Log 116 may log different types of relevant events, and may be used to create compliance audits for business or regulatory purposes. The Server 110 may be connected to a Request Queue 114, which may be connected to a Queue Prioritizer 115. Together the Request Queue 114 and Queue Prioritizer 115 may be responsible for maintaining and organizing any queued requests from requesting terminals. The Host 100 may be connected via Network 150 to a number of terminals. These may include a number of "Self-serve terminals" 120, which may include terminals similar to the example embodiment described in Figure 6. There may be a

number of terminals with a special designation such as, for example, "Casino terminals" 130. These terminals may have special prioritization rights in the Request Queue 114, or may have permanently assigned license tokens, which are never or rarely sent back to Token Pool 112. There may be a number of User Devices 140, for example a home computer, cell phone, or PDA. These devices may also be enabled to request license tokens from the Host 100 and perform "enabled state" actions when issued a license token.

Figure 2A illustrates an example procedure, according to an example embodiment of the present invention. The example procedure here might be used by an example terminal. The example procedure may initialize in a non-enabled state at 201. Alternatively, the terminal may be set to initialize in an enabled state, so long as doing so would not cause the total number of enabled terminals to exceed some limiting value. Next, the example procedure may wait at 210 and 215 until there is a need to enter an enabled state, e.g., a player requests to wager at the terminal. At this point, the example procedure may request a license token 220. At 222, the example procedure may check to see if license tokens are available. If license tokens are not available, the example procedure may wait at 224, and then check to see if a license token is still needed at 226. If a license token is still needed, another request may be made at 220. If a license token is no longer needed, e.g., the customer gave up on waiting or "balked", the example procedure may return to 215 to wait for the next request. If, however, a license token is available at 122, then the example procedure may receive a license token 230, and enter the enabled state 140. The example procedure may then perform an enabled state task at 250, or alternatively, perform any quantity of enabled state tasks before the example procedure returns to the non-enabled state at 260. The example procedure may then surrender the license token at 270. At 280, the example procedure may perform a non-enabled state task, e.g., entertainment only games or other non-wagering tasks. The example procedure may continue to perform the non-enabled tasks until there is another need to enter the enabled state.

Figure 2B illustrates an example procedure for controlling access to tasks using a license token, according to an example embodiment of the present invention. The example procedure here may be used by an example wager game terminal or other system that provides wagering game functionality. The example procedure may initialize in a non-enabled state at 202. Next, the example procedure may wait at 211 and 216 until an enabled state needs to be entered, e.g., when a user indicates a desire to wager at the terminal. At this point, the example procedure may request a license token 221. If there is no license token available, the example procedure may wait 225 and, at 227, check to see if the license token is still needed. If the license token is

no longer needed, e.g., the customer does not want to continue to wait, then the example procedure may go back to waiting for another license token to be needed at 216. At 227, if the license token is still needed, then another request may be made at 221. If, however, a license token is available at 223, then at 231, a license token may be required. At 236, instructions may be received along with the license token, and enter the enabled state at 241. At 251, an enabled state task may be performed, e.g., providing a wagering game to a player, such as video poker, keno, blackjack, slots, a video table game such as craps or roulette, or other wagering games. At 256, the enabled state task may continue (e.g., providing wagering games) until a license token return condition is met. Example license token return conditions include, e.g., an outside device may recall or request the license token, a time limit may have expired, a maximum amount wagered may have been met, etc. Once the return condition is met, at 261 the non-enabled state may be entered, and at 271, the license token may be surrendered. At 281, non-enabled state tasks may be performed, until there is another need to enter the enabled state.

Figure 3 illustrates an example procedure for a state changing token, according to an example embodiment of the present invention. The license token may be in one of the two states, e.g., the "issued" state 320 or the "available" state 340. At 330, the license token may stay in the "issued" state, until the license token is disassociated from the terminal the license token was issued to. At this point, the license token may enter the "available" state 340, and wait until the license token is again associated with a terminal, at 310. It will be appreciated that additional states including more features may also be taken on by a token, e.g., enablement of only particular games.

Figure 4 illustrates an example procedure, according to an example embodiment of the present invention. The illustrated example procedure may be used, e.g., at a central server providing the allocation control system described elsewhere in the present application. The example procedure may wait at 410 and 415 for a license token request. Upon a license token request, at 420, the example procedure may check to see if a license token is available. If a license token is not available, the example procedure may queue the request at 440 and continue to check for a license token at 420. The example procedure may also maintain the queue at 440, which may include prioritizing queued requests. After a request has been queued, the example procedure may wait at 445 and then check if a license token is still needed at 420. If a license token is no longer needed, the example procedure may return to checking for more requests at 410 and 415. If the license token is still needed at 420, then the example procedure may again check to see if one is available. If a license token is available, or when a license token becomes available, the

example procedure may issue the license token at 430, and return to waiting for requests at 410. It will be appreciated that alternative token allocation control procedures may be used. For example, the procedure may be able to receive multiple requests, and may receive additional requests while fulfilling prior requests. The queue may have several queued requests, and may prioritize those requests, while waiting for available license tokens and receiving additional requests. It will be appreciated that various queueing and prioritization approaches may be employed, e.g., FIFO, LIFO, preferential treatment of high yielding machines or preferred customers, etc.

Figure 5 illustrates another example procedure for token handling, according to an example embodiment of the present invention. At 501, the example procedure may create or receive a quantity of license tokens. The example procedure may then check to see if there in a license token request at 510. If not, at 520 the example procedure may then see if any recall conditions have been met. If not, at 530 the example procedure may check to see if the queue holds any requests in it. If the queue is empty, at 510 the example procedure may return to check for license token requests. At 510, if a request was received, the example procedure may check to see if a license token is available at 540. If no token is available at 540, the example procedure may queue the request at 560, prioritize the queue at 565, and return to checking for a request at 510. If a license token was available at 540, the example procedure may issue the license token at 550. The example procedure may send additional instructions to the requesting terminal at 555, and then return to checking for license token requests at 510. If the recall condition of 520 was met, the example procedure may recall a license token at 525, and then check to see if the queue contains any requests at 530. If the queue does contain requests, at 535 the example procedure may check to see if a license token is available. If not, the example procedure may return to checking for new requests at 510. However, if a license token is available, the example procedure may issue the license token at 550, and may send instructions to the terminal at 555. The example procedure may then return to checking for license token requests at 510. The example described above is sequential. However, it will be appreciated that alternative procedures may execute concurrently, so that a system executing the example procedure may always be ready to receive requests, check recall conditions, and maintain a queue, even while executing other parts of the example procedure.

Examples of recall conditions were previously illustrated and may include, e.g., the license token being with a particular terminal for a predetermined period of time, the license token being with a particular player for a predetermined period of time, the time of day, the amount wagered

at the terminal since the license token was issued, the amount wagered by a single player at the terminal, the profitability of the machine, the playing habits of a player at the machine, the size or contents of the request queue, the age of the oldest request in the queue, the average age of the requests in the queue, the location of the terminal the license token was issued to, the owner or operator of the terminal the license token was issued to, any other relevant condition, or any combination of these conditions. Similarly, the prioritization of the queue may be based on several factors, e.g., a simple first in first out (FIFO) configuration, the amount of time a request has been in the queue, the amount of time the terminal that sent the request has been in the queue during a particular interval, the profitability of a terminal which sent the request, the owner or operator of the terminal which sent the request, the location of the terminal which sent the request, the time of day, the identity of the player at the terminal which made the request, the amount wagered by the player at the terminal which sent the request, the betting habits of the player at the terminal which sent the request, as well as other relevant attributes, or any combination of these attributes. The additional instructions may include, e.g., any one of the recall conditions sent as an instruction to return the license token upon the condition (e.g., a push return instruction instead of a pull recall). The instruction may include an instruction on what games the terminal should make available, and that instruction may depend on any one of the above listed conditions or attributes (e.g., the time of day or location of the terminal). In some systems, it may be advantageous to always send a license token time-out instruction. In this case, if a terminal becomes disconnected from the system, the terminal will still know to discard the license token after the time-out period and the host will know to create a new license token after the time-out period. This "virtual return" may prevent a disconnected terminal with a license token from causing the loss of that license token (or occupying that license token) until the terminal is able to come back online into the system.

Figure 6 illustrates an example game terminal, according to an example embodiment of the present invention. The example system illustrated in Figure 6 may be a multi-state terminal system. The system may have a Processor 610, which may be connected to a number of devices. For example, the processor may be one or more microprocessors. The system may have a Network I/O Device 650 used to communicate with other systems via a network (e.g., a LAN, a private network, or a secure Internet connection). The system may have a Display 630, e.g., an LCD display, which may be used for a number of tasks including displaying a wagering game, advertisements, non-wagering games, a web browser, television signals, or any number of other visual media. The system may have a Memory 620. As is common with such systems the memory 620 may include several layers of memory, including hard disks, RAM, ROMS,

removable disks, external hard drives, or any number of other data storage accessible to the processor 610. The system may have an Input Device or Input Devices 640. These may include a keyboard, mouse, joystick, microphone, touch screen, buttons, levers, switches, or any other input device. The system may have other output devices (not shown) such as a printer, speaker(s), or other types of output devices.

The terminal may have a State Controller 660, with a Non-Enabled State Module 663, and an Enabled State Module 667. The state controller may be provided as software on the processor, or alternatively as a separate device in communication with the processor. The State Controller 660 may be responsible for controlling when the terminal is in which state. It may do this by receiving an enabling "license token" from another device or "license token pool" via the Network I/O Connection 650. While in the enabled state, the Enabled State Module 667 may be responsible for providing certain features such as, for example, a wagering game. The example terminal may also have Game Operation Software 670, which may be connected to Wagering Game Software 680 and optional Bonus Game Software 685. The software components 670, 680, and 685 may provide the wagering game as described with reference to the Enabled State Module 667. The Non-Enabled State Module 663 may be responsible for providing content other than the enabled state content. The terminal may include other Game Operation Software 670 components, e.g., Non-Wagering Software 673 or Attract Mode Software 677. These may include a web browser for Internet surfing or email checking, it may be advertisements, non-wagering games, television signals/shows, a movie, movie previews/trailers, an indication of the wait time (if any) before the terminal may enter the enabled state, or any number of other activities. The example terminal may know when tokens are available, and the Attract Mode Software 677 may be focused more on encouraging customers to request enabled state games. When no license tokens are available, Non-Wagering Software 673 may provide content with less focus on encouraging customers to request enabled state games. In some embodiments, the content of the two states will be mutually exclusive, and in other embodiments, the two states will have some shared content.

Figure 7 illustrates an example token allocation host, according to an example embodiment of the present invention. The example host may have a Processor 710, which may be connected to a number of devices. For example, the processor may be one or more microprocessors. The example host may have a Network I/O Device 750 used to communicate with other devices via a network (e.g., a LAN, a private network, or a secure Internet connection). The example host may have a Display 730, which may be used for a number of tasks including presenting a GUI

used for controlling and administering the license token pool system. The example host may have a Memory 720. As is common with such devices, the Memory 720 may include several layers of memory, including hard disks, RAM, ROMS, removable disks, external hard drives, or any number of other data storage accessible to the processor. The example host may have an Input Device or Input Devices 740. These may include a keyboard, mouse, joystick, microphone, touch screen, buttons, levers, switches, or any other input device. The example host may have other output devices (not shown) such as a printer, speaker(s), or other types of output devices. The example host may have an Event Log 725 where it records various events and information. Examples of these may include, e.g., revenue generated by each terminal that requests a license token, number of issued license tokens per time period, and number of license token requests not immediately fulfilled, or any other relevant data. The example host may have a License Token Log 728, which may track the location and/or status of every license token. The Event Log 725 and License Token Log 728 may have features allowing the devices to be auditable, so that total license usage reports may be produced to check regulatory compliance. These features may allow a regulatory authority or other entity to audit any of the described procedures or devices to verify that no more than the allowed quantity of gaming terminals or other devices operated wagering games (or were in a wagering game enabled mode, or ran a particular wagering game) at a particular time. Approaches to producing an auditable record may include, e.g., writing a record of all token transfers to a write-only or other secure archival media either on a host, at a dedicated remote location, or on the game machines themselves; recording the times that game machines enter or exit particular modes to hard meters or other auditable media on the game machines or in secure records on the host; recording logs of token transfers or machine state changes to a soft auditable record together with cryptographically secure timestamps obtained from an outside source; or random sampling of the states of various machines to produce statistical evidence that the required thresholds were note exceeded. It will be appreciated that the features provided may be altered or varied based on the particular regulatory requirements of a jurisdiction where the devices or procedures are employed.

The example host may have a Token Pool 760, which may include one or more license tokens whose allocation may be controlled by the example host. Here, as with all other example embodiments disclosed in this application, a license token does not have to be a physical thing. The example host may have an optional License Token Assignment Table 765 where the host may keep track of license tokens. This may be in addition to passing physical or virtual license tokens, or may be an alternative to passing physical or virtual license tokens. A license token may be a database maintained by the example host that knows which terminals are in "enabled"

states, and a license token pool may include ensuring that the number of terminals in the "enabled" state are less than or equal to some limit. In this example, the example host may send a message to the terminal informing that it may enter the enabled state, and the terminal may send a message back when leaving the enabled state. These messages and the recording database may be one embodiment of a "license token pool" and "license tokens". Other embodiments are possible to implement a shared resource among multiple resource requesting devices.

Figure 8 illustrates an example embodiment of the present invention, which shows five example terminals during fifteen consecutive points in time. In this example, the token pool (not shown) consists of two tokens. At T1, Machines 1-4 may initialize in non-wager mode. Machine 5 indicates that the machine is out of order 8502. At T2, Machine 2 requests a license token 8204, and Machine 5 is still out of order 8504. Machines 1, 3, and 4 enter Attract Mode. In this example, the machines (e.g., wagering game terminals) are configured to determine if tokens are available, even when no request has been made for a token. The machines may then enter an attract mode when tokens are available and the terminal is not currently in an enabled state. Attract mode may include content designed to encourage customers to request a wagering game. The machines may enter a non-wagering mode when the terminal is not in an enabled state and no tokens are available. In the non-wagering mode, non-wagering content may be provided, which does not encourage players to request a wagering game as much as the content of the attract mode. Attract mode content may include additional prompting, more wagering game advertisements, blocking out content such as television signals or reducing the screen size to allow for more advertisements, or any other types of content designed to encourage customers to request a wagering game.

At T3, since tokens are available, Machine 2 progresses to receiving a license token and entering the enabled state. The entering of the enabled state should never occur before the receiving of a token, but in some embodiments may occur at the same time. Also at T3, Machine 1 has requested a license token 8106. Machines 3 and 4 remain in attract mode, and Machine 5 remains out of order. At T4, Machine 1 receives a license token and enters the enabled state 8108. Machine 2 is providing a wagering game 8208 while in the enabled state. Now that both tokens have been issued, there are no additional tokens available, and Machines 3 and 4 switch from attract mode to non-wagering mode. Machine 5 has been fixed at T4, and is initializing in non-wagering mode.

At T5, Machines 1 and 2 are providing a wagering game, Machine 3 has requested a token, and Machines 4 and 5 are in non-wagering mode. At T6, Machines 1 and 2 are still providing a wagering game, and Machine 4 is still in non-wagering mode. Machine 5 has requested a license token 8512. Since no tokens are currently available, the request by Machine 3 has been queued 8312. At T7, Machine 1 stops providing a wagering game and exits the enabled state. The token is still not available for reissue, so Machine 3 is in queue mode 8314, and the request by Machine 5 has been queued 8514. At T8, Machine 1 returns the license token, and at T9 Machine 1 is in non-wagering mode. Also at T9, Machine 2 begins to exit the enabled state (e.g., because the player exited the wagering game). Also, queued Machine 3 is now issued a license token and enters the enabled state.

At T10, Machine 2 returns the license token, and Machine 3 begins providing a wagering game. At T11, queued Machine 5 receives the license token and enters the enabled state. At T12, Machine 3 exits the enabled state 8324, and at T13, Machine 3 returns the license token. Also at T13, Machine 5 begins to exit the enabled state. With the return of the token held by Machine 3, and with no requests in the queue, the Machines 1-4 switch from non-wagering mode to attract mode at T14. Machine 5 returns its token, and by T15 all five machines are in attract mode.

Figure 9 illustrates an example procedure according to an example embodiment of the present invention. The example procedure illustrated in Figure 9 includes two separate and concurrent flows of control, one for a client terminal and one for a host, which together illustrate an example interaction. At 9110, the terminal may begin by initializing in a "non-enabled" state. At 9210, the host may begin by creating, loading, or otherwise activating license tokens. The terminal may wait at 9115 until an enabled state request is made at the terminal (e.g., a player indicates a desire to play a wagering game at the terminal). Once an enabled state request is made, the terminal may request a license token at 9120. License token request 9020 may be any form of communication (e.g., data packets transferred over a network such as the Internet using a protocol such as TCP/IP). The License token request communication 9020 may be a synchronous protocol with multiple messages, asynchronous packet sent to an online transaction processing system and response, automated email queries, transfer of secure communication tokens, communications through proxies, or any other known transmission protocol or method. At 9220, the host may wait for the incoming License token request communication 9020.

After the host receives a request 9020, the host may check to see if a license token is available at 9225. If a license token is not available, the host procedure may queue the request and perform

any other queue maintenance tasks including prioritizing the queued requests at 9227. If a license token is available, the host, at 9230, may issue a license token 9030. This license token 9030 may be received by the terminal at 9130. The host may also send the terminal instructions 9035 along with the license token at 9235. The terminal may receive the instructions at 9135. These instructions could be anything and may include a return condition specifying a condition upon which the terminal should return the license token. Other relevant examples of instructions are listed in other embodiments. The terminal may then enter an "enabled" state and perform enabled state tasks at 9140 (e.g., provide a wagering game). The terminal may also check for any return condition such as the expiration of a timer or a player indicating they would like to cash-out or otherwise exit the enabled state at 9145. If not, the terminal will continue in enabled mode. As well as return conditions local to the terminal and player actions, a return condition may be the receiving of a recall message 9045. After the host sent instructions at 9235, the host may check to see if a recall condition has been met at 9240. Recall conditions could be anything, and relevant examples were listed in other embodiments, such as the length of the request queue. If a recall condition is not met, the host, at 9242, may continue to wait for either the recall condition to be met or for the license token to be returned without a recall at 9242. If a recall condition is met, a recall message 9045 may be sent to the terminal at 9245.

Once the return condition is met at 9145, the terminal may return to a "non-enabled" state at 9150. After exiting the enabled state, the terminal may return the license token at 9160, and go back to waiting for an enabled state request at 9115. The host may receive license token 9060, and return the license token to the license token pool at 9260. The host may then go back to waiting for license token requests at 9220.

Although this example procedure is shown in a sequential order, for the illustration, it will be appreciated that all of the example procedures described herein may execute concurrently, so that many terminals running the terminal procedure may issue requests parallel, and requests are constantly received and processed asynchronously at the host.

In an alternative embodiment, the examples systems and methods described in the present application may be used to enable systems to use licensed content, e.g., protected intellectual property such as well-known game properties, game software, or other licensed content. A game system operator may purchase a license that allows a particular number of game terminals to use a particular type of licensed content. The system may then prevent more than that licensed number of terminals from providing the licensed content at any one time. Assuming

regulations permit multiple wagering games to be offered by a particular machine, machines not currently using the particular licensed content might offer other wagering content, e.g., content owned by the game terminal operator, or content controlled by other licenses. In this case, the system regulates not the number of terminals offering wager, but rather the number of terminals offering a particular licensed game content. It will be appreciated that this alternative embodiment may be combined with the regulatory license embodiments, to control both the number of the terminals offering wagering, and the number of terminals offering a particular licensed game content at the same time, either by having multiple types of tokens, e.g., a token controlling wagering, and a token controlling the particular offered game, or by having multi-property tokens that control both features at the same time.

To allow for auditing and verification, all of the above-described systems and methods may include features to allow reliable verification of how game terminals were operated and/or the allocation of tokens over time in the system. These features may allow a regulatory authority or other entity to audit the above-described methods and systems to verify that no more than the allowed quantity of gaming terminals or other devices operated wagering games (or were in a wagering game enabled mode, or ran a particular wagering game) at a particular time. Approaches to producing an auditable record may include, e.g., writing a record of all token transfers to a write-only or other secure archival media either on a host, at a dedicated remote location, or on the game machines themselves; recording the times that game machines enter or exit particular modes to hard meters or other auditable media on the game machines or in secure records on the host; recording logs of token transfers or machine state changes to a soft auditable record together with cryptographically secure timestamps obtained from an outside source; or random sampling of the states of various machines to produce statistical evidence that the required thresholds were note exceeded. It will be appreciated that the features provided may be altered or varied based on the particular regulatory requirements of a jurisdiction where the systems and methods are employed.

In alternative embodiments, a set of terminals may operate using a distributed control system without the need for a central or shared host. In this case the terminals may distribute the license token pool and queue requests for tokens among themselves in a peer-to-peer fashion. When a terminal exits the enabled state, the terminal may announce the availability of a license token to other terminals, and another terminal may claim the license token directly from the peer terminal.

It will be appreciated that all of the disclosed methods, games, and procedures described herein can be implemented using one or more computer programs or components. These components may be provided as a series of computer instructions on any conventional computer-readable medium, including RAM, ROM, flash memory, magnetic or optical disks, optical memory, or other storage media. The instructions may be configured to be executed by a processor which, when executing the series of computer instructions, performs or facilitates the performance of all or part of the disclosed methods, games, and procedures.

It should be understood that there exist implementations of other variations and modifications of the invention and its various aspects, as may be readily apparent to those of ordinary skill in the art, and that the invention is not limited by specific embodiments described herein. Features and embodiments described above may be combined. It is therefore contemplated to cover any and all modifications, variations, combinations or equivalents that fall within the scope of the basic underlying principles disclosed and claimed herein.

CLAIMS

1. A system, comprising:

a plurality of wagering game terminals each having a token-enabled state and a non-enabled state;

a plurality of tokens associated with a capability, the quantity of tokens less than the quantity of terminals, each of the plurality of tokens configured, when held by one of the terminals, to allow the terminal to operate in the token-enabled state;

wherein the plurality of terminals are each configured to enter the token-enabled state only when a token is received by the terminal and to return to the non-enabled state when the token is surrendered by the terminal; and

wherein each terminal in the token-enabled state is configured to provide at least one wagering game to a player, and wherein each terminal in the non-enabled state is prevented from offering wagering games.

2. The system of claim 1, wherein the terminal in the token-enabled state is configured to provide a selection of one or more of game formats from a plurality of different wagering game formats, the selection based, at least in part, on instructions associated with the token.

3. The system of claim 1 or 2, wherein the instructions associated with the token are based, at least in part, on the time of day.

4. The system of any of claims 1 to 3, wherein the return to the non-enabled state is made responsive to a user input.

5. The system of any of claim 1 to 3, wherein the return to the non-enabled state is made responsive to the terminal being in the token-enabled state for at least a predetermined period of time.

6. The system of claim any of claims 1 to 5, wherein the terminals are each configured to log data including at least one of: revenue generated by a terminal in the token-enabled state, the amount of time the terminal has spent in the token-enabled state, and a count of the number of token requests not immediately fulfilled, or time periods spent in the token-enable stated.

7. The system of any of claims 1 to 6, wherein the plurality of terminals are in network communication with each other, and wherein the terminals are configured to communicate to each other information indicating that a token is available.

8. The system of any of claims 1 to 7, wherein the terminals are configured to queue token requests when token requests exceed token availability.

9. The system of any of claim 7, wherein the order of terminals in the queue is prioritized based, at least in part, on at least one of: the amount of time the terminal has been in the queue, the location of the terminal, the amount of time the terminal has already been in the token-enabled state during a time period, the operator of the terminal, or the profitability of the terminal.

10. The system of claim of any of claims 1 to 9, wherein the wagering game terminal is configured to provide at least one of: a video slot machine game, a video poker game, a keno game, a lottery game, a mechanical slot machine game, a video blackjack game, a video craps game, a video bingo game, or a video roulette game.

11. A method of providing wagering games at a wagering game terminal, comprising:

whenever the game terminal is in a non-enabled state, preventing the game terminal from providing wagering games;

while the game terminal is in the non-enabled state, requesting a token associated with a capability to provide a wagering game;

responsive to the request, receiving the token;

entering a token-enabled state, responsive to the receipt of a token;

providing at least one wagering game to a player at the game terminal while in the token-enabled state; and

returning to the non-enabled state and surrendering the token.

12. The method of claim 11, further comprising:

receiving a player request to play a wagering game; and

responsive to receiving the player request, entering the token-enabled state.

13. The method of claim 11 or 12, wherein the terminal is further configured to provide a plurality of wagering game formats, the method further comprising:

receiving instructions associated with the token; and

selecting one or more game formats to provide from the plurality of wagering game formats based, at least in part, on the instructions.

14. The method of any of claims 11 to 13, wherein the instructions associated with the token depend at least in part on the time of day.

15. The method of any of claims 11 to 14, wherein the return to the non-enabled state is caused by a user exiting the token-enabled state.

16. The method of any of claims 11 to 14, wherein the return to the non-enabled state is caused by the terminal being in the token-enabled state for at least a predetermined period of time.

17. The method of any of claims 11 to 14, wherein the return to the non-enabled state and release of the token is made responsive to the amount lost by the player exceeding a predetermined limit.

18. The method of any of claims 11 to 17, further comprising:

logging data for the terminal including at least one of: revenue generated while in the token-enabled state, time spent waiting for a token, or amount of time spent in the token-enabled state, or time periods spent in the token-enable state.

19. The method of any of claims 11 to 18, further comprising:

responsive to the surrendering of the token, communicating the availability of the token to other terminals.

20. A system, comprising:

a plurality of tokens associated with a capability, wherein each token is configured to be in one of one or more states, the states including at least an issued state and an available state, wherein tokens are configured to enter the issued state responsive to an entity requesting a token and that token being in the available state; and

wherein tokens are configured to enter the available state responsive to being surrendered by the entity the token has been issued to,

the tokens configured, when allocated to a wagering game terminal, to enable a wagering game terminal to provide a wagering game terminal to a game player.

21. The system of 20, wherein each token is associated with a license to operate a single wagering terminal.

22. The system of claim 21, wherein the tokens are further configured to, when surrendered by a game terminal providing wagering games, to cause the game terminals to enter a mode where the play of wagering games is prevented at the terminal until another token is received at the terminal.

23. A method of controlling the operation of plurality of wagering game terminals using a quantity of terminal licenses smaller than the quantity of game terminals, each of the terminals having wager-enabled and non-enabled states, the method comprising:

responsive to a request to allow a game terminal in the non-enabled state to offer a wagering game, allowing the game terminal to enter the enabled state only if the total quantity of terminals currently in the enabled state is less than the quantity of terminal licenses; and

responsive to the game terminal entering the enabled state, offering a wagering game at the terminal.

24. The method of claim 23, further comprising:

preventing the offering of a wagering game at the terminal whenever the terminal is in the non-enabled state.

25. The method of claim 23 or 24, further comprising:

maintaining a count of the quantity of game terminals currently in the wager-enabled state.

26. The method of claim 25, further comprising:

responsive to the quantity of game terminals in the wager-enabled state being greater than or equal to the quantity of terminal licenses, when the request is received, queueing the request.

27. The method of any of claims 23 to 26, further comprising:

responsive to a terminal transitioning from the wager-enabled to the non-enabled state, allowing a terminal associated with a queued request to enter the wager-enabled state.

28. The method of any of claims 23 to 27, further comprising:

receiving a player request to play a wagering game at a terminal in the non-enabled state; and

responsive to the player request making the request to allow the game terminal in the non-enabled state to offer the wagering game.

29. The method of any of claims 23 to 28, further comprising:

providing a non-wagering game to a player at a game terminal in the non-enabled state.

30. The method of any of claims 23 to 28, further comprising:

operating a terminal not currently used by a player in attract mode.

31. The method of 30, wherein information displayed by the terminal in attract mode depends at least in part on the number of game terminals that are currently in wager-enabled mode.

32. The method of any of claims 23 to 31, further comprising:

changing the state of a terminal in wager-enabled mode to non-enabled mode.

33. The method of claim 32, wherein the change to wager non-enabled mode is made responsive to at least one of: the terminal being idle for at least a predetermined period of time, the total amount lost by a player at the terminal, a request made by another terminal to enter the wager-enabled mode, the time of day, or the utilization of the terminal during a predetermined time interval.

34. A system, comprising:

a server configured to control the allocation of a plurality of tokens associated with a capability;

the server configured to receive a request from a terminal for a token, wherein the number of terminals configured to issue a request for a token to the server is greater than the number of tokens the server is configured to control;

the server configured to issue a token, responsive to a request for a token and a token being available; and

the server configured to queue requests for tokens, if no tokens are currently available.

35. The system of 34, wherein the number of tokens the server is configured to control allocation of corresponds to a quantity of licenses available for the operation of wagering terminals.

36. The system of claim 34 or 35, wherein the server is configured to control token allocation for at least one location, and wherein the maximum number of tokens the server will issue to terminals located within any particular location depends on the number of wagering game machine licenses issued for that location.

37. The system of any of claims 34 to 36, wherein the server is configured to send instructions to the terminal associated with issuing a token to that terminal.

38. The system of 37, wherein the instructions include at least one of the following: which wagering game formats will be available at the terminal, the minimum wager allowed at the terminal, the maximum wager allowed at the terminal, or the maximum amount of time the terminal will be issued the token.

39. The system of any of claims 34 to 38, wherein the server is configured to recall an issued token.

40. The system of any of claims 34 to 39, wherein the queue of terminals which have requested a token is prioritized according to one or more of the following criteria: the amount of time the terminal has been in the queue, the location of the terminal, the amount of time the terminal has already been in the token-enabled state during a time period, the operator of the terminal, or the profitability of the terminal.

41. The system of any of 34 to 40, wherein the server is configured to log data including any one or more of the following: revenue generated by each terminal that requests a token, number of issued tokens per time period, and number of token requests not immediately fulfilled.

42. A method comprising:

controlling the allocation of a plurality of tokens associated with a capability;

receiving a request from a terminal for a token, wherein the number of terminals configured to issue a request for a token is greater than the number of tokens in the plurality of tokens;

issuing a token responsive to a request for a token if a token is available; and

queueing requests for tokens if no tokens are currently available.

43. The method of claim 42, wherein the number of tokens in the plurality of tokens corresponds to a number of licenses for the operation of a wagering terminal.

44. The method of claim 43, wherein the controlling the allocation includes controlling the allocation of tokens for at least one jurisdiction, and wherein the maximum number of tokens the method will issue to terminals located within any particular jurisdiction depends on the number of licenses issued for that jurisdiction.

45. The method of claim 43 or 44, further comprising:

sending instructions to the terminal associated with the token.

46. The method of any of claims 43 to 45, wherein the instructions include at least one of the following: which wagering game formats will be available at the terminal, the minimum wager allowed at the terminal, the maximum wager allowed at the terminal, or the maximum amount of time the terminal will be issued the token.

47. The method of any of claims 43 to 46, further comprising:

recalling an issued token.

48. The method of any of claims 42 to 47, further comprising:

prioritizing the queue of terminals which have requested a token according to one or more of the following criteria: the amount of time the terminal has been in the queue, the location of the terminal, the amount of time the terminal has already been in the token-enabled state during a time period, the operator of the terminal, or the profitability of the terminal.

49. The method of any of claims 43 to 48, further comprising:

logging data including any one or more of the following: revenue generated by each terminal that requests a token, number of issued tokens per time period, and number of token requests not immediately fulfilled.

50. A method comprising:

controlling the allocation of a plurality of tokens to a plurality of wagering game terminals, wherein a token is a license for the operation of a wagering game, and wherein the number of terminals in the plurality of wagering game terminals is greater than the number of tokens in the plurality of tokens;

logging data associated with the allocation of the plurality of tokens, including at least the number of tokens allocated to wagering game terminals at any particular time; and

providing an account of token allocations.

51. The method of claim 50, wherein the providing an account of token allocations includes providing a report, which satisfies a reporting or compliance requirement associated with the licenses.

52. A system for sharing authorization tokens, comprising:

a plurality of terminals, each configured to operate in both a first state, where a regulated action is prevented, and a second state where a regulated action is permitted;

at least one server configured to control the allocation of at least one token, wherein the token is configured to provide an authorization grant to a terminal, and wherein the number of tokens available in the system corresponds to a number of licenses;

the at least one server configured to control token allocation for at least one location, wherein the number of tokens the at least one server will issue to terminals located in a particular location is based on the number of licenses available for that particular location;

a first terminal configured, when in the first state, to request a token, responsive to a request by a user to perform a regulated action;

the first terminal configured, responsive to receipt of a token, to enter the second state;

the second state configured to have at least two sub-states, wherein the at least one server determines which sub-states are available for the first terminal to enter;

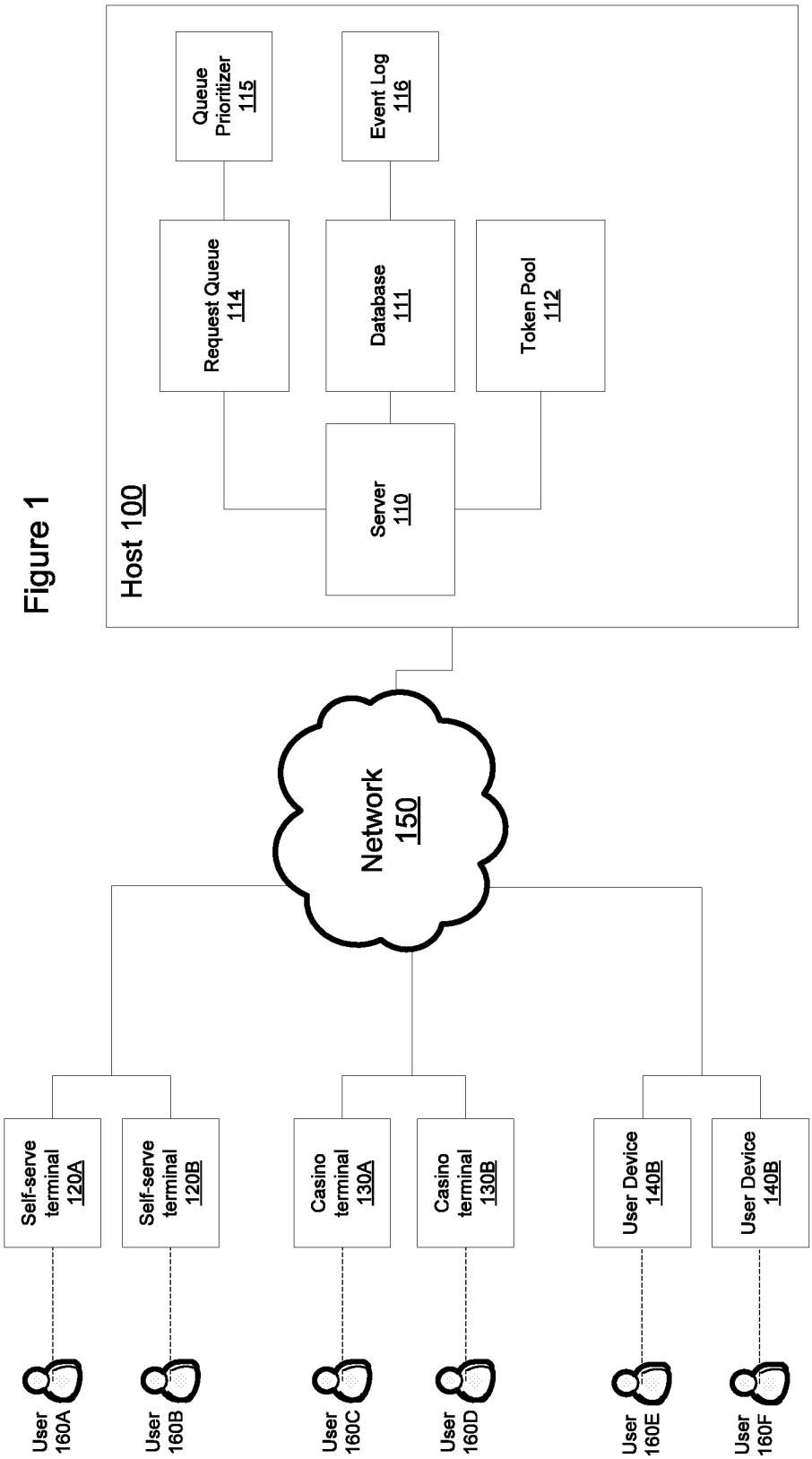
the first terminal configured, responsive to returning to the first state, to return the token to the server;

the at least one server configured to queue a request for a token from a second terminal when no tokens are currently available;

the at least one server configured to ensure that the number of terminals in the second state at any one time does not exceed the number of terminals legally allowed to operate a wagering game in the location where the terminals are located.

53. A computer-readable storage medium encoded with instructions configured to be executed by a processor, the instructions which, when executed by the processor, cause the performance of the method of any of claims 11-19, 23-33, or 42-51.

Figure 1


Host 100
Server 110
Request Queue 114
Queue Prioritizer 115
Database 111
Event Log 116
Token Pool 112
Network 150
Self-serve terminal 120A
Self-serve terminal 120B
Casino terminal 130A
Casino terminal 130B
User Device 140B
User Device 140B
User 160A
User 160B
User 160C
User 160D
User 160E
User 160F

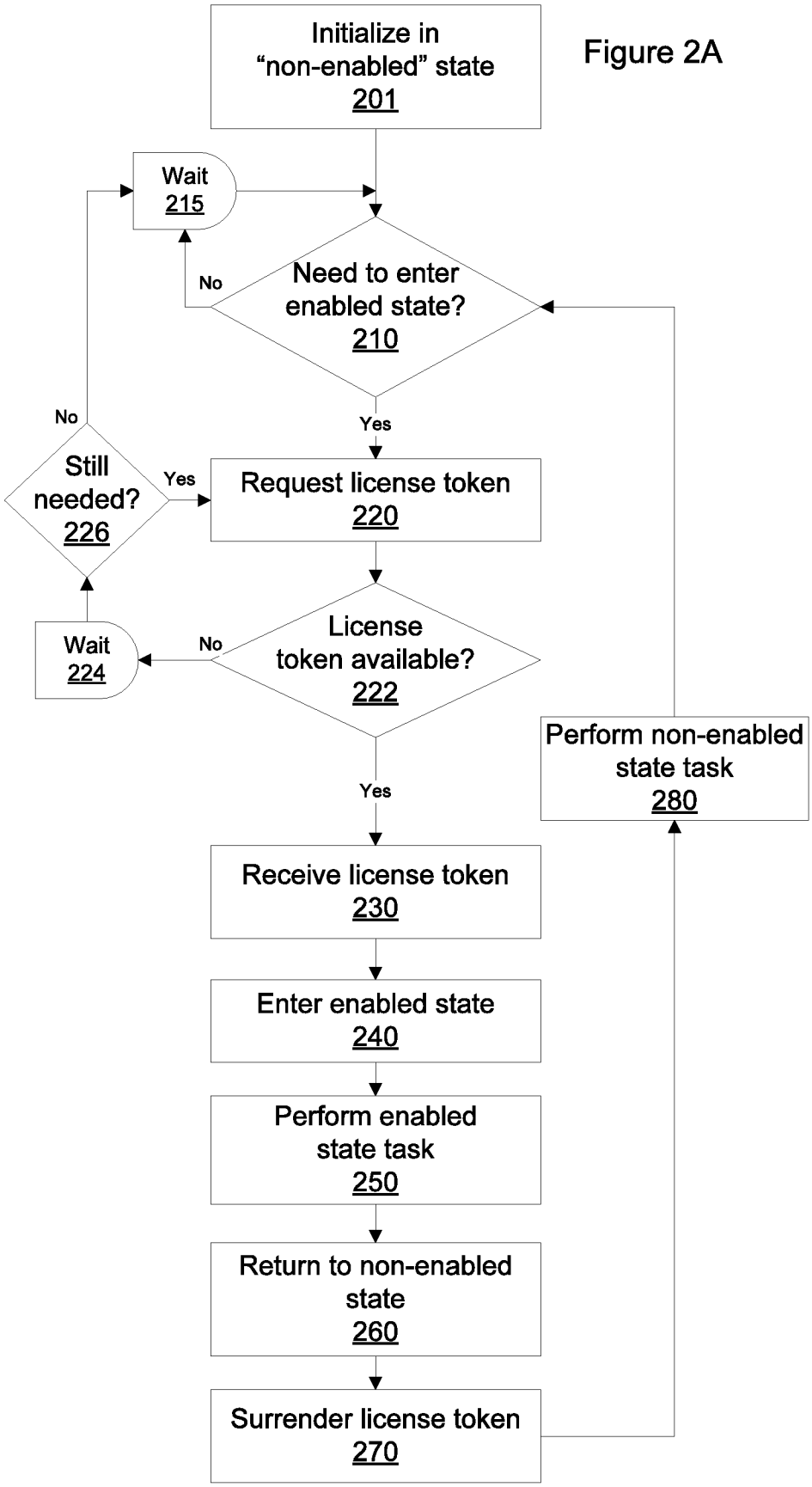
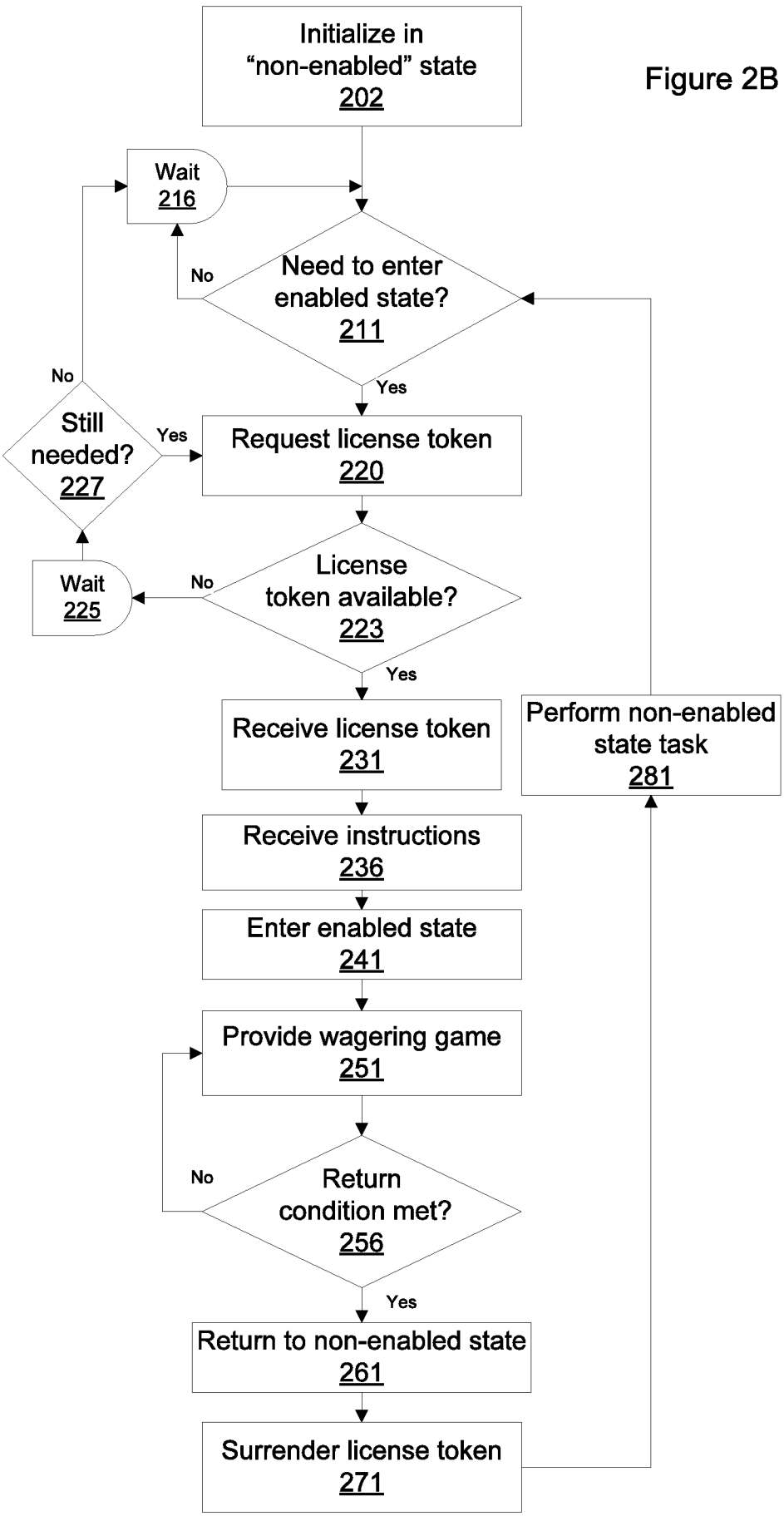

Initialize in "non-enabled" state 201
Wait 215
No
Need to enter enabled state? 210
Yes
No
Still needed? 226
Yes
Request license token 220
Wait 224
No
License token available? 222
Yes
Perform non-enabled state task 280
Receive license token 230
Enter enabled state 240
Perform enabled state task 250
Return to non-enabled state 260
Surrender license token 270

Figure 2A

Figure 2B


Initialize in "non-enabled" state 202
Wait 216
Need to enter enabled state? 211
No
Yes
Still needed? 227
No
Yes
Request license token 220
Wait 225
License token available? 223
No
Yes
Receive license token 231
Perform non-enabled state task 281
Receive instructions 236
Enter enabled state 241
Provide wagering game 251
Return condition met? 256
No
Yes
Return to non-enabled state 261
Surrender license token 271

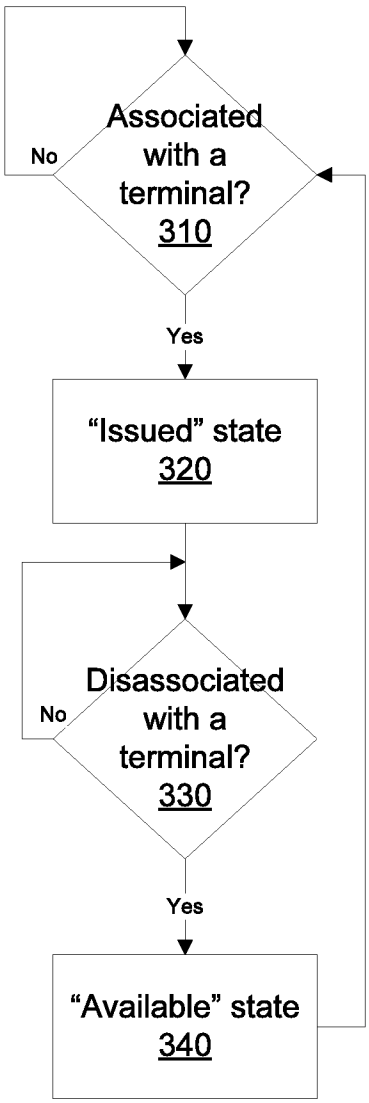

Associated with a terminal? 310
No
Yes
"Issued" state 320
Disassociated with a terminal? 330
No
Yes
"Available" state 340

Figure 3

Figure 4

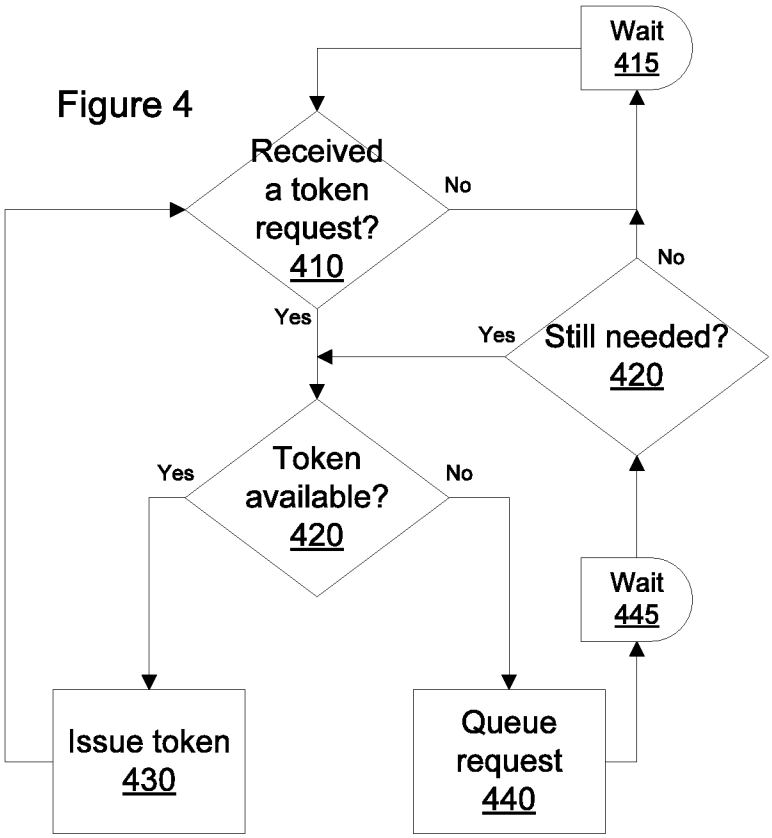

Wait
415
Received a token request?
410
No
Yes
No
Yes
Still needed?
420
Yes
Token available?
420
No
Wait
445
Issue token
430
Queue request
440

Figure 5

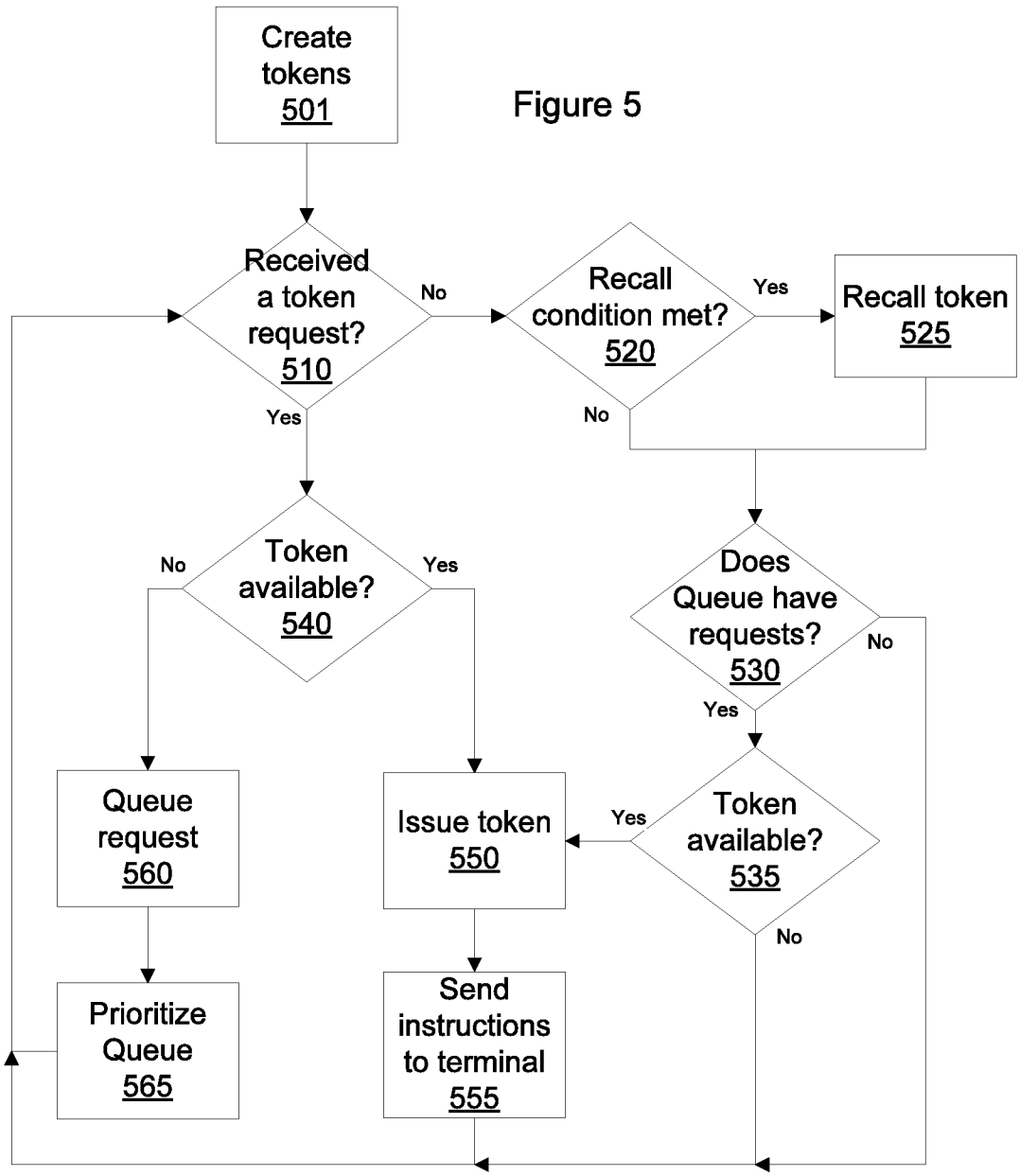

Create tokens 501
Received a token request? 510
No
Recall condition met? 520
Yes
Recall token 525
Yes
No
No
Token available? 540
Yes
Does Queue have requests? 530
No
Yes
Queue request 560
Issue token 550
Yes
Token available? 535
No
Prioritize Queue 565
Send instructions to terminal 555

Figure 6

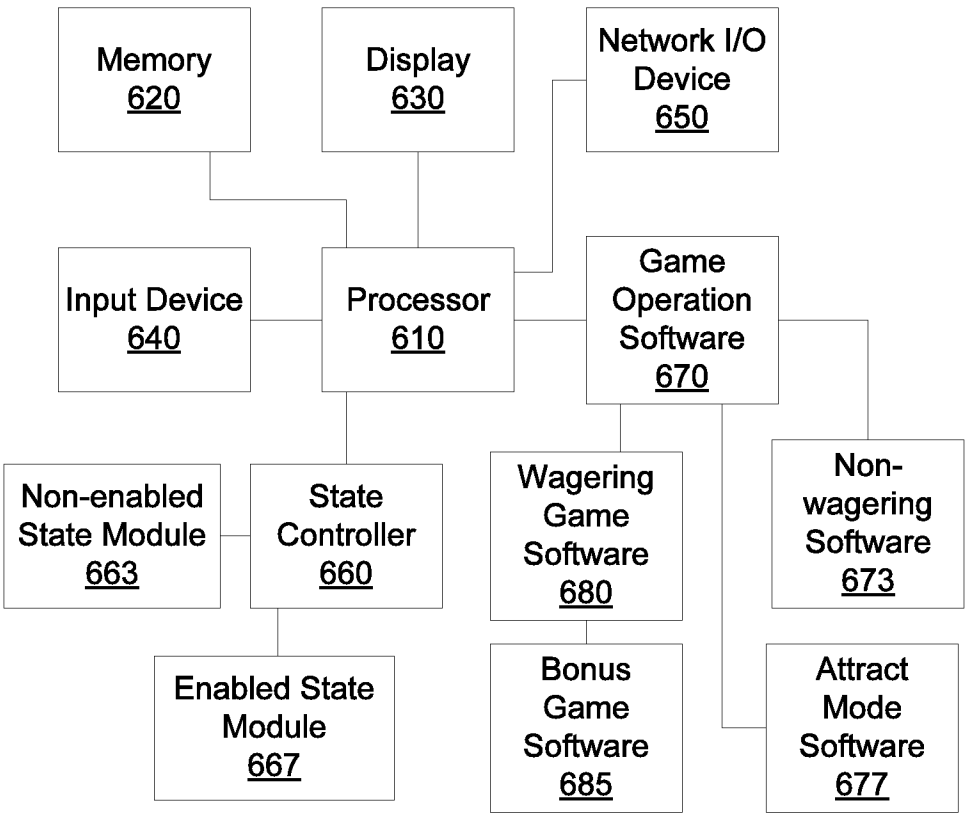
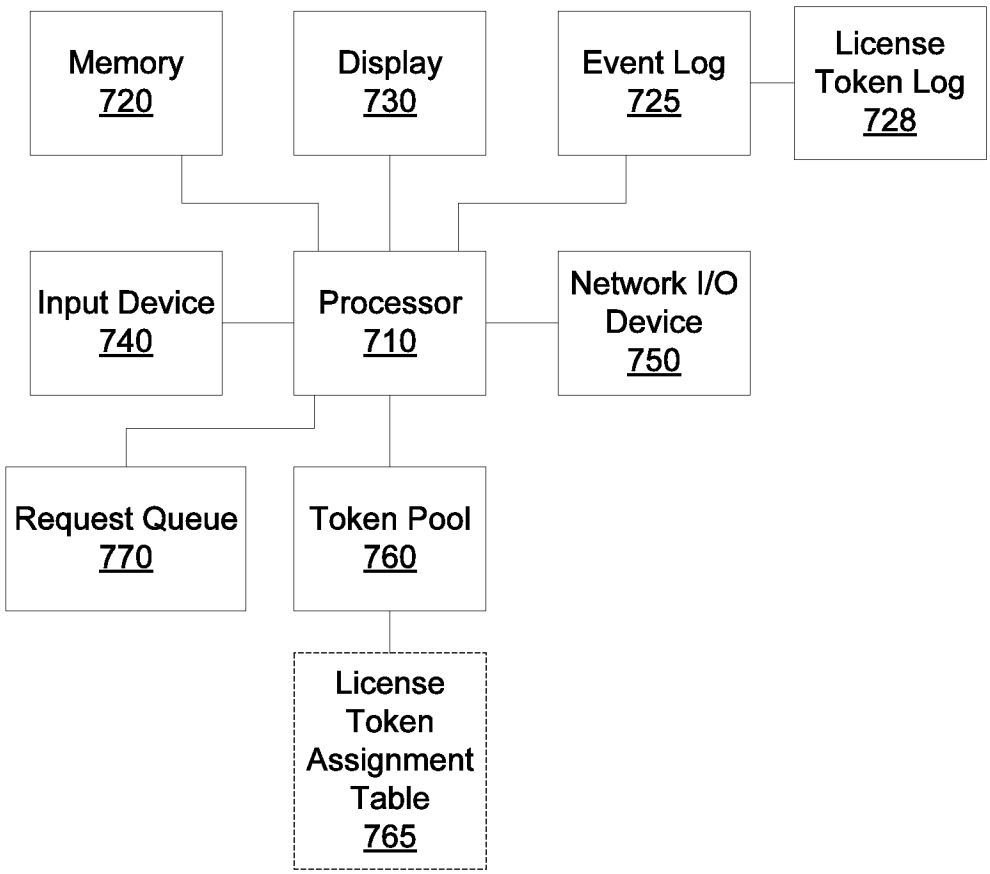

Memory
620
Display
630
Network I/O
Device
650
Input Device
640
Processor
610
Game
Operation
Software
670
Non-enabled
State Module
663
State
Controller
660
Wagering
Game
Software
680
Non-
wagering
Software
673
Enabled State
Module
667
Bonus
Game
Software
685
Attract
Mode
Software
677

Figure 7


Memory
720
Display
730
Event Log
725
License
Token Log
728
Input Device
740
Processor
710
Network I/O
Device
750
Request Queue
770
Token Pool
760
License
Token
Assignment
Table
765

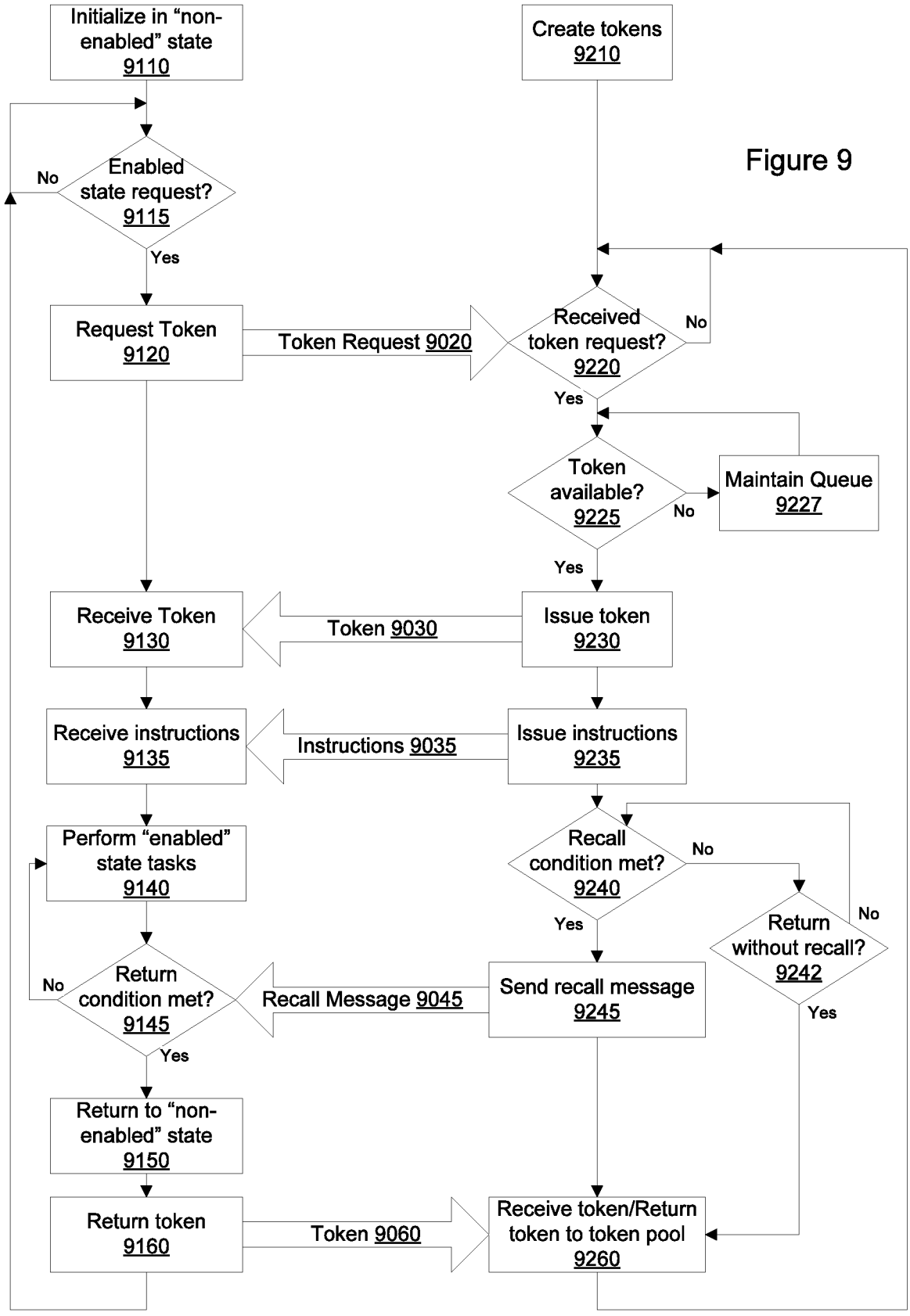

Initialize in "non-enabled" state 9110
Create tokens 9210
Enabled state request? 9115
No
Yes
Request Token 9120
Token Request 9020
Received token request? 9220
No
Yes
Token available? 9225
No
Maintain Queue 9227
Yes
Receive Token 9130
Token 9030
Issue token 9230
Receive instructions 9135
Instructions 9035
Issue instructions 9235
Perform "enabled" state tasks 9140
Recall condition met? 9240
No
Yes
Return without recall? 9242
No
Yes
Return condition met? 9145
No
Yes
Recall Message 9045
Send recall message 9245
Return to "non-enabled" state 9150
Return token 9160
Token 9060
Receive token/Return token to token pool 9260

Figure 9

Figure 8

	Machine 1	Machine 2	Machine 3	Machine 4	Machine 5
T1	Initialize Non-wager Mode 8102	Initialize Non-wager Mode 8202	Initialize Non-wager Mode 8302	Initialize Non-wager Mode 8402	Out of Order 8502
T2	Attract Mode 8104	Request License Token 8204	Attract Mode 8304	Attract Mode 8404	Out of Order 8504
T3	Request License Token 8106	Receive License Token Enter Enabled State 8206	Attract Mode 8306	Attract Mode 8406	Out of Order 8506
T4	Receive License Token Enter Enabled State 8108	Provide Wagering Game 8208	Non-wagering mode 8308	Non-wagering mode 8408	Fixed: initializing in non-wagering mode 8508
T5	Provide Wagering Game 8110	Provide Wagering Game 8210	Request License Token 8310	Non-wagering mode 8410	Non-wagering mode 8510
T6	Provide Wagering Game 8112	Provide Wagering Game 8212	Queue Request 8312	Non-wagering mode 8412	Request License Token 8512
T7	Exit Enabled State 8114	Provide Wagering Game 8214	Queue Mode 8314	Non-wagering mode 8414	Queue Request 8514
T8	Return License Token 8116	Provide Wagering Game 8216	Queue Mode 8316	Non-wagering mode 8416	Queue Mode 8516
T9	Non-wagering mode 8118	Exit Enabled State 8218	Receive License Token Enter Enabled State 8318	Non-wagering mode 8418	Queue Mode 8518
T10	Non-wagering mode 8120	Return License Token 8220	Provide Wagering Game 8320	Non-wagering mode 8420	Queue Mode 8520
T11	Non-wagering mode 8122	Non-wagering mode 8222	Provide Wagering Game 8322	Non-wagering mode 8422	Receive License Token Enter Enabled State 8522
T12	Non-wagering mode 8124	Non-wagering mode 8224	Exit Enabled State 8324	Non-wagering mode 8424	Provide Wagering Game 8524
T13	Non-wagering mode 8126	Non-wagering mode 8226	Return License Token 8326	Non-wagering mode 8426	Exit Enabled State 8526
T14	Attract Mode 8128	Attract Mode 8228	Attract Mode 8328	Attract Mode 8428	Return License Token 8528
T15	Attract Mode 8130	Attract Mode 8230	Attract Mode 8330	Attract Mode 8430	Attract Mode 8530