The captured data can be transmitted to the at least one electronic device for use in wagering.
REMOTE GAMING SYSTEM AND RANDOM NUMBER GENERATION METHOD

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims priority to U.S. Provisional Application No. 61/829,610, filed May 31, 2013 and U.S. Provisional Application No. 62/004,069, filed May 28, 2014, these applications being fully incorporated by reference herein.

FIELD OF TECHNOLOGY

[0002] The present disclosure relates generally to electronic systems, and more particularly to a system and method of utilizing randomly generated numbers for wagering within a network.

BACKGROUND

[0003] Wagering on games of chance is a common form of entertainment. Most games of chance require that participants be physically present during play. This can be inconvenient and restrict the number of available players. What is needed is a system whereby players can participate remotely in a secure fashion.

BRIEF DESCRIPTION OF THE DRAWINGS

[0004] In order to describe a manner in which features of the disclosure can be obtained, reference is made to specific embodiments that are illustrated in the appended drawings. Based on an understanding that these drawings depict only example embodiments of the disclosure and are not intended to be limiting of scope, the principles herein are described and explained with additional specificity and detail through the use of the accompanying drawings in which:

[0005] Figure 1 is a flow chart of a method 100 of remote gaming in accordance with an example embodiment;

[0006] Figure 2 is a flow chart of another method 200 of remote gaming in accordance with an example embodiment;
Figure 3 is a flow chart of another method 300 of remote gaming in accordance with an example embodiment;

Figure 4 is a block diagram representing an example embodiment of a remote gaming system;

Figure 5 is a block diagram representing an example embodiment of a remote gaming system architecture;

Figure 6 illustrates an example of a vision system that can be used to collect and capture information about dice rolls;

Figure 7 illustrates a flow chart of possible game states of dice games as described within the disclosure;

Figure 8 is a flow chart of an example method of executing a dice game within a remote gaming system as disclosed herein;

Figure 9 is a flow chart of an example method of authenticating communications or transmissions sent into the remote gaming system; and

Figure 10 illustrates a computing device system according to one example.

DETAILED DESCRIPTION

Various embodiments of the disclosure are discussed in detail below. While specific implementations are discussed, it should be understood that this is done for illustration purposes only. Other components and configurations may be used without departing from the scope of the disclosure.

Within this disclosure, the term "computing device" is defined as any device that is capable of at least accepting data, transmitting data, and executing commands. For example, computing devices can include, but are not limited to, programmable logic controllers, portable communication devices, mobile communication devices, mobile computers, smart phones, computing pads, tablet computers, personal computers, desktop computers, laptop computers, netbooks, servers, routers, set-top phones, or other computing device capable of at least accepting data, transmitting data, and executing commands.
The term "matrix barcode" is defined as 2-dimensioned barcode which is a two-dimensional way to represent information. An example of a matrix barcode is a QR Code. The information encoded by a QR code may be made up of four standardized types ("modes") of data (numeric, alphanumeric, byte / binary, Kanji) or, through supported extensions, virtually any type of data. A QR code consists of black modules (square dots) arranged in a square grid on a white background, which can be read by an imaging device (such as a camera) and processed using Reed-Solomon error correction until the image can be appropriately interpreted; data can be extracted from patterns present in both horizontal and vertical components of the image.

According to another broad aspect, there is provided a method of remote gaming, the method comprising: receiving at least one control command from at least one electronic device; transmitting the at least one control command to at least one logic controller; generating at least one random number sequence by at least one mechanical apparatus in response to at least one instruction by the at least one controller; capturing data pertaining to the at least one random number sequence; and transmitting information related to the captured data to at least one remote server. The captured data can be stored of subsequent retrieval and use. The information related to the captured data to the at least one electronic device. The captured data can pertain to a dice roll performed by the mechanical apparatus.

According to another broad aspect, there is provided a method comprising: time-stamping and storing the information relating to the captured data for subsequent retrieval; receiving a request for random number data from at an electronic device; and transmitting to the electronic device, the most recently time-stamped and stored information which was not previously transmitted to an electronic device. Thus it is possible that each packet of captured (dice roll) data might be utilized only one time and/or by only one player.

According to another broad aspect, there is provided a method comprising: comparing data captured by a vision system within a gaming machine and stored on the machine and with corresponding captured data stored on the at least one remote server; and transmitting the corresponding captured data to the at least one electronic device, if captured data stored on the machine and corresponding captured data stored on the at least one remote server is the same.

According to another broad aspect, there is provided a remote-gaming system, the system
including: a processor; a non-transitory computer readable storage medium storing instructions for controlling the processor to perform steps comprising: receiving at least one control command from at least one electronic device; transmitting the at least one control command to at least one logic controller; generating at least one random number sequence by at least one mechanical apparatus in response to at least one instruction by the at least one controller; capturing data pertaining to the at least one random number sequence; and transmitting information related to the captured data to at least one remote server.

According to another broad aspect, there is provided a non-transitory computer-readable storage medium storing instructions which, when executed by an electronic system, cause the electronic system to: receive at least one control command from at least one electronic device; transmit the at least one control command to at least one logic controller; generate at least one random number sequence by at least one mechanical apparatus in response to at least one instruction by the at least one controller; capture data pertaining to the at least one random number sequence; and transmit information related to the captured data to at least one remote server.

Figure 1 is a flow chart of a method 100 within the disclosure. Each block shown in Figure 1 represents one or more processes, methods, or subroutines, carried out in example method 100. Method 100 can begin when a user of a remote device such as a mobile communications device inputs 104 a command indicating that he wishes to play a game or place a wager. A control command can then be sent 108 to a controller such as a programmable logic device. The controller can be in control of a machine which is configured to generate random numbers, such as a device like that described in U.S. Patent No. 6,394,901. The controller can cause the machine to generate a random number 112. Data relating to the generated number can be captured 116 by, for example, a digital camera housed within or connected to the machine. Once the data is captured, the data can be transmitted 120 to appropriate endpoints, such as a memory location within the machine, a remote server, and/or one or more electronic devices, such as, for example, the electronic device into which the original control command was entered 104 by a user. The method 100 can be repeated multiple times so as to enable multiple games to be played or multiple wagers to be placed by multiple users. The method 100 illustrated in Figure 1 is provided by way of example, as there are a variety of ways to carry out the method.
Additionally, while the example method 100 is illustrated with a particular order of steps, it will be understood that Figure 1 and the steps illustrated therein can be executed in any order that accomplishes the technical advantages of the present disclosure and can include fewer or more steps than illustrated. Furthermore, the example method 100 can be combined with other methods, such as those disclosed herein, in order to achieve the technical advantages of this disclosure.

[0024] Figure 2 is a flow chart of a second method 200 within the disclosure. The method 200 is described with reference to the location in which the steps of the method 200 occur. Again, it will be understood that the example method 200 is illustrative and is non-limiting. The steps of method 200 are performed by or on at least one mobile device 204, at least one server 208 and at least one machine 212, such as a dice machine. The method 200 begins when a player makes a bet 214 via at least one mobile device 204. A signal pertaining to the bet is sent to a server 208, where a determination is made 218 as to whether the bet is valid. If the bet is invalid, the method 200 returns to step 214 and the player may make another bet, upon which the method returns to step 218 and so on. If the bet is valid the method 200 can proceed to step 222, where event information, such as information relating to the placing of the bet, is conveyed 222 to all the players. The method 200 can then proceed to step 226 wherein a shooter inputs a signal or command to cause the machine 212 to generate a random number, such as by rolling one or more dice. Once the dice-roll signal has been sent, betting ends 230. Once betting ends 230, event information relating to the ending of betting is conveyed to all the players 234. Also, after betting ends 230, the machine 212 rolls the dice and reads the information on the dice at step 238. The value of the dice roll is then sent to the server 208 where a determination is made as to whether another dice roll is required 242 according to rules of the game being played, such as for example, the game known as "craps." If another dice roll is necessary, the method 200 returns to step 226 where the shooter again rolls the dice and the method 200 continues as described above. If a determination is made 242 that another dice roll is not necessary, the method 200 continues to step 246, where the results of the game are conveyed to all players and bets are paid 246 as appropriate according to the rules of the game being played.

[0025] Figure 3 is a flow chart of a third method 300 within the disclosure. Again it will be understood that method 300 constitutes a non-limiting example and that elements of the method
can be performed in conjunction with or as an alternative to, other methods, such as method 100 and method 200. The method 300 starts at step 304. At step 308 the dice machine 212 invites play. Play can be initiated either locally to the machine 212 or remotely at a remote device or remote application 204. Bets can be placed, either at the machine itself 312 and/or via one or more remote applications/devices 316. A dice roll procedure 320 can then be initiated, either by the pulling of a lever on the machine 212 or other suitable input such as the pressing of a button, or by the inputting 328 of a start command at a remote device 204. A dice roll procedure 320 can include mechanically tumbling dice, dumping the tumbled dice in a suitable receptacle, centering the dice within the receptacle and then elevating the dice. Once the dice roll procedure 320 is completed, the method 300 can proceed to step 332. At step 332, which can be known as the vision process, one or more suitable images of the elevated dice can be captured, by, for example, a digital camera. Once the vision process 332, is complete the results of the dice roll can be save or transmitted 336 to a remote device 204, a server or other suitable endpoints for further use or possible retrieval and/or verification. The method 300 can then return to start 304. Dice results can be transmitted to a remote device 204. In at least one embodiment of the disclosure, the method 300 can require an appropriate electronic "handshake" 344 with such a remote device prior to providing the dice results of step 336. Once the dice results are provided 344 to the appropriate remote devices, the method 300 can then return to start 304.

[0026] Figure 4 is a block diagram representing an example embodiment 400 of a remote gaming system. Machine 410 can comprise any standard commercially available hardware, for example: a server. Machine 410 can be the Dice Master. Machine 410 is electrically connected to a Programmable Logic Controller (PLC) 420. PLC 420 is a digital computer used for automation of electromechanical processes, such as control of machinery, for example PLC 420 can control machine 410. While this example is discussed with reference to a PLC, any suitable computing device can be used. PLC 420 can be an Automation Direct Do-More system based on the 205 rack platform. PLC 420 can use Do-More Software compute sequential logic. PLC 420 can also include an Operator Interface Terminal (OIT) 450. OIT 450 can be a touchscreen, for example a C-More touchscreen. OIT 450 can selectively control the game based on the allowable entries and interface points from a combination of programming in PLC 420 and OIT 450, via an operator input. PLC 420 can also include a network module (not show), such as, a
Local Area Network module, an Ethernet module, a Wireless Fidelity module, or any other network connection module for connection to network 430. Network 430 can be the Internet, a Computer Network, a Telephone Network, an Intranet, an Extranet, the global Telex network, a Local Area Network (LAN), a Wireless Local Area Network (WLAN), a Wide Area Network (WAN), a Metropolitan Area Network (MAN), a Backbone Network, a Campus Area Network (CAN), a Storage Area Network (SAN), a Home Area Network (HAN), a Personal Area Network (PAN), a Wireless Fidelity Network (Wi-Fi), Virtual Private Network (VPN), Virtual Network, Cellular Network, Satellite Network, and Radio Frequency Network (RF). Remote devices 440 can communicate with PLC 420 in order to control Machine 410 via network 430. Remote devices 440 can be any device that is capable of at least accepting data, transmitting data, and executing commands. For example, electronic devices can include, but are not limited to, portable communication devices, mobile communication devices, mobile computers, smartphones, computing pads, tablet computers, personal computers, desktop computers, laptop computers, netbooks, servers, routers, set-top phones, or other electronic devices capable of at least accepting data, transmitting data, and executing commands.

[0027] Figure 5 is a block diagram representing an example embodiment 500 of remote gaming system architecture. Dice machine 510 can be a gaming apparatus for simulating the play of the dice game craps, for example. Dice machine 510 can have a network module (not shown) to securely connect to server 520 via web service 550. Server 520 can be any physical computer system dedicated to running one or more services, to serve the requests of other computers or electronic devices on a communications network. Server 520 can include a web server 550, one or more game engines 530 and a database 560. Web service 550 can manage incoming and outgoing communication via a secure connection, such as HTTPS or other secure protocols. Game engines 530 can receive a bet from mobile devices 540, determine the validity of a bet, determine the end of betting, and determine if another roll of the dice is necessary. Database 560 can store user profile data, user betting history, user gaming result history, user account status, and user gaming balance. Mobile devices 540 can connect securely to server 520 via web service 550. After a mobile device 540 connects to server 520, mobile device 540 can securely operate dice machine 510.

[0028] Figure 6 illustrates an example of a vision system 600 that can be used to collect and
capture information about dice rolls as described above. A vision sensor, such as an optical sensor, video camera, digital camera or other suitable device capable of capturing visual data can, when suitably placed and configured, be used to photograph or otherwise record the visual data associated with one or more dice rolls within this embodiment. For example, after dice are rolled by a dice rolling machine, the device can center the rolled dice 608 beneath an input portion of the vision sensor 604. When placed in viewing range of the vision sensor 604, the dice can be illuminated by an appropriate light source 612 such as a light-emitting diode. The vision system 600 communicates with a processor such as a PLD via a suitable transfer protocol, such as Modbus TCP. Data captured by the vision system 600 can be communicated to one or more external endpoints, as can relevant usage data, which can be used to statistically verify the results of dice rolls on an ongoing basis.

[0029] To detect the dice and increase the accuracy, the gaming system of the disclosure can use and exception-based inspection method. The method uses process of elimination to determine which dice are present, since a 5, 3, 1 have a similar pattern of dots, as do 2,4,5, etc. This method produces a higher statistical accuracy than pure detection of the dice, since the patterns are very similar. The steps of this process can include the following:

[0030] 1. Count all dots on the dice present. Since the dice are clear and of some color, this step can be relied upon to clearly identify the white dots through filtering. This step provides a total count for the dice, but does not identify the individual dice. From this step, the remaining inspections can be established;

[0031] 2. Find Sixes - finds and counts all dice that match the 6 pattern;

[0032] 3. Find Fives - finds and counts all dice that match the 5 pattern;

[0033] 4. Find Fours - finds and counts all dice that match the 4 pattern;

[0034] 5. Find Threes - finds and counts all dice that match the 3 pattern; and

[0035] 6. Find Twos - finds and counts all dice that match the 2 pattern.

[0036] Because the total of the dice count is known, the following matrix, shown in Table 1 below, can be implemented by the system to determine whether the value on each individual die is correct. This process can improve the speed of the vision system and can reduce the chances
of error in dice data before it is utilized.

<table>
<thead>
<tr>
<th>Dice Total</th>
<th>Inspections</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>Only Solution is two sixes, no further inspection needed</td>
</tr>
<tr>
<td>11</td>
<td>Only Solution is one six and one five, no further inspection needed</td>
</tr>
<tr>
<td>10</td>
<td>Check 6, Check 5, If a 6 is present, then other is 4, if 5 then two fives</td>
</tr>
<tr>
<td>9</td>
<td>Check 6, Check 5, If a 6 is present, then other is 3, if 5 then the other is 4</td>
</tr>
<tr>
<td>8</td>
<td>Check 6, Check 5, Check 4, If 6 is present then other is 2, if 5 then other is 3, else 2 four’s</td>
</tr>
<tr>
<td>7</td>
<td>Check 6, Check 5, Check 4, If 6 is present then other is 1, if 5 then other is 2, else 1 four and 1 three</td>
</tr>
<tr>
<td>6</td>
<td>Check 5, Check 4, If 5 is present then other is 1, if 4 then other is 2, else 2 three’s</td>
</tr>
<tr>
<td>5</td>
<td>Check 4, If 4 is present then other is 1, else 1 three and 1 two</td>
</tr>
<tr>
<td>4</td>
<td>Check 3, If 3 is present then other is 1, else 2 two’s</td>
</tr>
<tr>
<td>3</td>
<td>Only Solution is one two and a one</td>
</tr>
<tr>
<td>2</td>
<td>Only Solution is two ones.</td>
</tr>
</tbody>
</table>

Table 1

[0037] By checking in the order of 6, 5, 4, 3, 2, 1 as in Table 1, the possibility of confusing a 4 as two 2's or a 6 as two 3's can be eliminated, for instance. This procedure can be made necessary by the clear aspect of "official" dice, which require a high degree of filtering and fast shutter speeds accurately detect the dots on the surface and while not capturing the dots opposite side. Such filtering can eliminate the edges of the dice, leaving the dots on the dice as the only visible element. Thus, the proximate nature of the rolled dice could confuse the system as the relative location of the edges is not known, causing a misread. The method of testing first for the higher number eliminates the possibility of erroneous detection of smaller counts, since the higher counts (6 for instance) have only one possible alignment to equal the spacing pattern, whereas a 2 can have multiple alignments which satisfy its pattern. Thus, accuracy of the captured values of dice rolls is enhanced.

[0038] The vision system, once it receives the results of the dice roll in question, communicates with the logic controller, which in turn tabulates the results and reports them locally and/or remotely, as discussed herein.

[0039] Figure 7 illustrates a flow chart of possible game states of dice games as described within the disclosure. At state 700, a shooter, either local to a dice machine or via a signal input into a
remote electronic device, places a bet. Thereafter additional players are able to bet against the shooter 704. Players may then enter into side bets 708 amongst themselves relating to the possible outcome of a forthcoming dice roll initiated by the shooter. Once all bets have been entered, the shooter causes the dice machine to roll the dice and the results of the dice roll are determined 712 and transmitted to relevant end points, such one or more remote servers or one or more electronic devices. Thereafter, players who won their bets, as determined by the outcome of the dice roll(s) in state 712, are paid 716 as appropriate.

[0040] As discussed above, aspects of the disclosure pertain to a gaming machine, (which can be known as the "Dice Master"), that is partially or wholly controlled by one or more remote electronic devices. For remote communications with the Dice Master, a suitable transmission Control Protocol, such as Modbus TCP with a direct-connect fixed IP address can be utilized. The Dice Master can be considered the Server and the remote application (or remote electronic device running such an application) is the client, meaning the remote application must initiate and control the communication interface, and that the PLC (programmable logic controller) will only answer and respond to a request from the remote application.

[0041] It can be advantageous to provide robust handshaking to control the frequency and safe control of the application. If the handshaking fails within a set time, the Dice Master will revert to a local mode and the remote interface may need to reinitialize the conversation. A Free running clock from the machine can be used to assist in this handshaking procedure.

[0042] The gaming machine has at least three modes of operation: a local mode, a remote mode and a hybrid mode. In the local mode, in which an operator or player at the machine is in control of the dice roll and his own betting. In the local mode, a remote application can monitor actions of the machine for remote betting and interface. However, in the local mode, the roll by the local operator/player can occur whenever the local operator/player chooses. In the remote mode, operation of the machine is completely remotely controlled; the local system is "off" and there is no local betting or dice rolling possible. In the hybrid mode, operation of the machine is local and controlled by the pull handle. However, the system will wait for up to 60 seconds after all local betting is complete to allow remote betting to be complete. If the betting is complete before the 60 seconds has expired, then the remote application will indicate via a command to release the machine for operation. In this mode, the betting locally is updated to the remote
application through registers indicating the current bank and winnings for the local operator.

[0043] For each mode, there are defined commands and status flags and words to indicate mode and type of operation. Various commands are listed in Tables 2-4, shown below.

<table>
<thead>
<tr>
<th>Tagname</th>
<th>MB Address</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CmdRemote</td>
<td>400000, Bit 0</td>
<td>BOOL</td>
<td>When on, the machine will look for the initiate command to effect a dice roll. If the machine is currently in play locally, the operator will have to acknowledge the request and relinquish control to the remote app. Please look for the StsMessage as it will include a message indicator from the local operator like “Please Wait” or “Let’s play together”, which would enable the remote app to switch to the Hybrid Mode.</td>
</tr>
<tr>
<td>CmdInitiate</td>
<td>400000, Bit 1</td>
<td>BOOL</td>
<td>When in the Remote mode, this command will effect a dice roll. It must be only be set when the step = 4. Each set will only effect one roll</td>
</tr>
<tr>
<td>CmdClear</td>
<td>400000, Bit 2</td>
<td>BOOL</td>
<td>This command will clear all the bets and current state of the Dice Master machine. (This is a remote diagnostic tool only and is not currently used)</td>
</tr>
<tr>
<td>CmdHandshake</td>
<td>400000, Bit 3</td>
<td>BOOL</td>
<td>The handshake corresponds to the Status, if both are off then the handshake command should be turned on, if both are on, then the handshake command should be turned off. Using this system allows both the Remote App and the Dice Master to effect their own alarms.</td>
</tr>
<tr>
<td>CmdHybrid</td>
<td>400000, Bit 4</td>
<td>BOOL</td>
<td>If this command is on, then the Dice Master will delay the Pull of the handle by up to 60 seconds to allow for remote betting.</td>
</tr>
<tr>
<td>CmdHybridDone</td>
<td>400000, Bit 5</td>
<td>BOOL</td>
<td>If set and the Hybrid command is on, then the machine will be allowed to roll locally.</td>
</tr>
<tr>
<td>CmdLogoff</td>
<td>400000, Bit 15</td>
<td>BOOL</td>
<td>This command is used to tell the Dice Master that the remote app is operable and that no current activity is require remotely.</td>
</tr>
<tr>
<td>CmdClock</td>
<td>400002</td>
<td>INT</td>
<td>Signed integer that should be echoed from the Dice Master Value sent in the status area. Both Sides can use this to check for communication timeouts or packet errors since the update rate is relative to the value. The Dice Master will update this value every 10msec, therefore message timing can be indicated by the value delta.</td>
</tr>
<tr>
<td>CmdMessage</td>
<td>400003</td>
<td>INT</td>
<td>Register based messaging at the local app, controlled by the remote app. This enables the remote app to cause messages to be displayed on the local machine, like “Yes! I won!” or “Craps-Craps-Craps”. Up to 1000 messages can be stored on the Dice Master OIT.</td>
</tr>
<tr>
<td>CmdWinnings</td>
<td>400004</td>
<td>INT</td>
<td>Remote Players winnings for display (0-65534)</td>
</tr>
</tbody>
</table>

Table 2
<table>
<thead>
<tr>
<th>Tagname</th>
<th>MB Address</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>StsRemote</td>
<td>400101, Bit 0</td>
<td>BOOL</td>
<td>When on, the machine accepts the request for Remote Play. If local play is underway, the operator is prompted for the request and must acknowledge the request to enable the remote app to take over, alternately the operator may request the hybrid mode to play jointly.</td>
</tr>
<tr>
<td>StsInitiate</td>
<td>400101, Bit 1</td>
<td>BOOL</td>
<td>Echoes the Initiate Command, once the remote app sees the status update, it can clear the command.</td>
</tr>
<tr>
<td>StsClear</td>
<td>400101, Bit 2</td>
<td>BOOL</td>
<td>Echoes the command that all betting is clear.</td>
</tr>
<tr>
<td>StsHandshake</td>
<td>400101, Bit 3</td>
<td>BOOL</td>
<td>The handshake corresponds to the Command, if the command is on, then this bit will be turned on, if the Command is off, then this bit will be turned off.</td>
</tr>
<tr>
<td>StsHybrid</td>
<td>400101, Bit 4</td>
<td>BOOL</td>
<td>Echoes the Command for Hybrid mode and indicates that the request was accepted</td>
</tr>
<tr>
<td>StsHybridDone</td>
<td>400101, Bit 5</td>
<td>BOOL</td>
<td>Echoes the command indicating that local control is restored</td>
</tr>
<tr>
<td>StsLogoff</td>
<td>400101, Bit 15</td>
<td>BOOL</td>
<td>Echoes the command for Logoff.</td>
</tr>
<tr>
<td>StsClock</td>
<td>400102</td>
<td>INT</td>
<td>Signed integer that should be incremented by one every 10msec. Can be used to time message transactions or used as a free running heartbeat.</td>
</tr>
<tr>
<td>StsMessage</td>
<td>400103</td>
<td>INT</td>
<td>Register based messaging at the Dice Master. Used to respond to the message from the Remote App. A local message selection system is used, if a request to switch to remote mode is made, a response of “Go away, I am enjoying the game” could be sent, indicated by a message number.</td>
</tr>
<tr>
<td>StsWinnings</td>
<td>400104</td>
<td>INT</td>
<td>Current winnings for the Local User (0-65,534)</td>
</tr>
<tr>
<td>StsStep</td>
<td>400105</td>
<td>INT</td>
<td>Current Step number (Steps explained below)</td>
</tr>
<tr>
<td>StsDice1</td>
<td>400106</td>
<td>INT</td>
<td>Score of Dice 1, 1-6</td>
</tr>
<tr>
<td>StsDice2</td>
<td>400107</td>
<td>INT</td>
<td>Score of Dice 2, 1-6</td>
</tr>
<tr>
<td>StsDice1LocX</td>
<td>400108</td>
<td>INT</td>
<td>Location of Dice 1 &quot;x&quot; location</td>
</tr>
<tr>
<td>StsDice1LocY</td>
<td>400109</td>
<td>INT</td>
<td>Location of Dice 1 &quot;y&quot; location</td>
</tr>
<tr>
<td>StsDice1Rotation</td>
<td>400110</td>
<td>INT</td>
<td>Degree rotation of the Dice</td>
</tr>
<tr>
<td>StsDice2LocX</td>
<td>400111</td>
<td>INT</td>
<td>Location of Dice 2 &quot;x&quot; location</td>
</tr>
<tr>
<td>StsDice2LocY</td>
<td>400112</td>
<td>INT</td>
<td>Location of Dice 2 &quot;y&quot; location</td>
</tr>
<tr>
<td>StsDice2Rotation</td>
<td>400113</td>
<td>INT</td>
<td>Degree rotation of the Dice</td>
</tr>
</tbody>
</table>

**Table 3**

12
As indicated in the preceding tables, the gaming machine responds with a corresponding status to the command with an "equal to" model, if CmdLogon is on, then StsLogon will echo that the machine has accepted the command. Since the model is asynchronous, we allow three retries before alarming and forcing a reinitialize of the communication link. Since the Dice Master will revert to the local mode, the remote interface must first begin the handshaking routine before attempting to logon or initiate a roll.

Within this disclosure, the remote application is always "Command" or "Cmd" and the Dice Master is "Status" or "Sts". The remote application can write to the Commands, but not to Status and the Dice Master will do the opposite. Any error in the command words, such as setting a wrong flag or sending a value outside or range is rejected at the Dice Master, causing it to start a retry. If the retries expire, the Dice Master will revert to local mode and the remote application will need to reinitialize to return to commanding the application.

The same steps are followed in the remote application, however the idle step becomes 4 since the local betting is not required. If in the hybrid mode, the operator locally will control
step 2, however a timeout mode will be active, so if the local operator leaves to get an adult beverage, the remote application can continue betting, however the remote application cannot initiate a roll in the remote application.

[0047] Figure 8 is a flow chart of an example method of executing a dice game within a remote gaming system. The method 800 illustrated is provided by way of example, as there are a variety of ways to carry out the method. Additionally, while the example method is illustrated with a particular order of steps, the steps illustrated therein can be executed in any order that accomplishes the technical advantages of the present disclosure and can include fewer or more steps than illustrated.

[0048] Method 800 can begin at block 805. At block 805, a player places a bet on the "pass" or the "don't pass." After the bet is placed, the method can move to block 810. At block 810, a shooter rolls two die. If the combined addition of the resulting two die faces is equal to "7" or "11" the method can proceed to block 815. If the combined addition of the resulting two die faces is equal to "2," "3" or "12" the method can proceed to block 830. If the combined addition of the resulting two die faces is equal to "4," "5," "6," "8," "9" or "10" the method can proceed to 820. At block 815, any bets placed on the "pass" win their bet and any bets placed on the "don't pass" lose their bets. After the bets are paid, the method can end. At block 830, any bets placed on the "don't pass" win their bet and any bets placed on the "pass" lose their bets. After the bets are paid, the method can end. At block 820, the "point" is assigned to the number equal to the addition of the dice. After the "point" is assigned, the method can proceed to block 835. At block 835, the shooter (which can include a mechanical device within this disclosure) rolls the dice. If the combined addition of the resulting two die faces is equal to the assigned "point" the method can proceed to block 815. If the combined addition of the resulting two die faces is equal to "7" the method can proceed to block 830. If the combined addition of the resulting two die faces is equal to any other number, the method returned to block 835 and the shooter rolls again.

[0049] Figure 9 is a flow chart of an example method of authenticating communications or transmissions sent into the remote gaming system from one or more remote devices. The method 900 illustrated in Figure 9 is provided by way of example, as there are a variety of ways to carry out the method. Additionally, while the example method is illustrated with a particular...
order of steps, the steps illustrated therein can be executed in any order that accomplishes the technical advantages of the present disclosure and can include fewer or more steps than illustrated.

[0050] Method 900 can begin at block 905 or block 915. At block 905 a user who does not have an account can create an account. After a sign up in requested, the method can proceed to block 910. At block 910, a user can create a username and password. A user can also create a profile and store personal and financial information. After a user has created a username and password the method can proceed to block 920. At block 915, a user who already has an account can log into their account. When a user wants to login in, the method can proceed to block 920. At block 920, the system determines if DYCRAVTM is used. DYCRAVTM is computer software for dynamic challenge-response, namely identification, authentication, verification, and used to ensure secure creation, storage and exchange of compressed and/or encrypted data. If DYCRAVTM is used the method can proceed to block 935. If DYCRAVTM is not used the method can proceed to block 925. At block 925, a user enters their username and password. After a user enters their username and password, the method can proceed to block 930. At block 930, the username and password is mutually authenticated using transport layer security (TLS). If a user cannot be authenticated method 900 will end. After a user is authenticated, the method can proceed to block 945. At block 935, the user enters DYCRAVTM requirements. After the user enters DYCRAVTM, the method can proceed to block 940. At block 940, the user DYCRAVTM requirements are mutually authenticated using DYCRAVTM. If DYCRAVTM are not authenticated method 900 ends. After authentication using DYCRAVTM is successful, the method can proceed to block 945. At block 945, a user can interact with the remote gaming system until an application timeout or the user manually logs out. After the timeout has been reached or the user manually logs out, the method can proceed to block 950. At block 950, a user is logged out and method 900 can end. It will be understood that other methods of verifying user and device information are possible within this disclosure.

[0051] As discussed above, the machine controller is commonly referred to as a Programmable Logic Controller (PLC) and it provides control of devices through inputs and outputs based on the configuration of the system. The PLC can be a rack-based system, which enables flexibility in the control components needed to operate the gaming machine. The PLC for this machine is
can consist of an AutomationDirect Do-More system based on the 205 rack platform. The processor can be configured using Do-More Software (Ver. 1.4) in standard rung based sequential logic. The PLC can include Ethernet communications using both the Do-More interface protocol and Modbus TCP. An Ethernet system can support several communication connections simultaneously.

[0052] The Operator Interface Terminal (OIT) for the gaming machine is can consist of a C-More touch screen. The C-More touch screen can be configured using C-More Software (Ver. 2.7.5) with standard graphical components from included libraries. The configuration of the touch screen enables the operator to selectively control the game based on the allowable entries and interface points from a combination of programming in the PLC and the OIT. The PLC responds to inputs and controls the machine, while outputting data to the OIT for display and controlling the graphics.

[0053] Several components of the gaming machine require control from the PLC, mainly the stepper motors that control the tumbler, dumper, elevator, and swiper. The PLC has configuration tools to enable programming of the patterns that the stepper motors will follow in order to transport the dice through the machine. The output from the PLC's stepper control cards go to a stepper "driver" that then controls the number of steps that the stepper motor will move and in what direction. The stepper motors will respond to the driver and move the required distance and speed.

[0054] As discussed above, the included vision sensor is also controlled through the PLC. The PLC communicates with the vision system through a Modbus TCP interface with the PLC serving as the Client, while the Vision system is the Server. When the vision system is needed, the PLC initiates the triggers for the camera to capture images, then the vision system returns the results to the PLC for processing and display on the OIT and to the remote apps. In some embodiments, the vision system cannot be operated independently of the PLC.

[0055] Within this disclosure, remote applications can interface with the machine through Modbus TCP and a "driver" software package on a remote computer. The remote application uses a command structure in concert with the PLC to control the machine and report results remotely. The machine can be configured to deny remote request if a loss of handshaking or
connection is detected, this adds to the safety of the machine and the ability of a local player to gain access to the machine when the remote app is idle.

[0056] Although embodiments within this disclosure have been discussed in the context of dice-based games, it will be understood that aspects of this disclosure can pertain to any wagering game that utilizes random numbers generated by mechanical means or other suitable means.

[0057] The disclosure now turns to a brief description of a computing device, such as a remote electronic device, as shown in Figure 10, which can be employed to practice the concepts is disclosed herein. The components disclosed herein can be incorporated in whole or in part into tablet computers, personal computers, handsets, transmitters, servers, and any other electronic or other computing device.

[0058] With reference to Figure 10, an example system 1000 includes a general-purpose computing device 1000, including a processing unit (CPU or processor) 1020 and a system bus 1010 that couples various system components including the system memory 1030 such as read only memory (ROM) 1040 and random access memory (RAM) 1050 to the processor 1020. The system 1000 can include a cache 1022 of high speed memory connected directly with, in close proximity to, or integrated as part of the processor 1020. The system 1000 copies data from the memory 1030 and/or the storage device 1060 to the cache 1022 for quick access by the processor 1020. In this way, the cache provides a performance boost that avoids processor 1020 delays while waiting for data. These and other modules can control or be configured to control the processor 1020 to perform various actions. Other system memory 1030 may be available for use as well. The memory 1030 can include multiple different types of memory with different performance characteristics. It can be appreciated that the disclosure may operate on a computing device 1000 with more than one processor 1020 or on a group or cluster of computing devices networked together to provide greater processing capability. The processor 1020 can include any general purpose processor and a hardware module or software module, such as module 1 1062, module 2 1064, and module 3 1066 stored in storage device 1060 and configured to control the processor 1020 as well as a special-purpose processor where software instructions are incorporated into the actual processor design. The processor 1020 may essentially be a completely self-contained computing system, containing multiple cores or processors, a bus, memory controller, cache, etc. A multi-core processor may be symmetric or asymmetric.
The system bus 1010 may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures. A basic input/output system (BIOS) stored in ROM 1040 or the like, may provide the basic routine that helps to transfer information between elements within the computing device 1000, such as during start-up. The computing device 1000 further includes storage devices 1060 such as a hard disk drive, a magnetic disk drive, an optical disk drive, tape drive or the like. The storage device 1060 can include software modules 1062, 1064, 1066 for controlling the processor 1020. Other hardware or software modules are contemplated. The storage device 1060 is connected to the system bus 1010 by a drive interface. The drives and the associated computer readable storage media provide nonvolatile storage of computer readable instructions, data structures, program modules and other data for the computing device 1000. In one aspect, a hardware module that performs a particular function includes the software component stored in a non-transitory computer-readable medium in connection with the necessary hardware components, such as the processor 1020, bus 1010, display 1070, and so forth, to carry out the function. Appropriate variations are contemplated depending on the type of device, such as whether the device 1000 is a small, handheld computing device, a desktop computer, or a computer server.

Although the example embodiment described herein employs a hard disk 1060, it should be appreciated that other types of computer readable media storing data that are accessible by a computer may also be used in this example operating environment, such as magnetic cassettes, flash memory cards, digital versatile disks, cartridges, random access memories (RAMs) 1050, read only memory (ROM) 1040, a cable or wireless signal containing a bit stream and the like. Non-transitory computer-readable storage media expressly exclude media such as energy, carrier signals, electromagnetic waves, and signals per se.

To enable user interaction with the computing device 1000, an input device 1090 represents any number of input mechanisms, such as a microphone for speech, a touch-sensitive screen for gesture or graphical input, keyboard, mouse, motion input, speech and so forth. An output device 1070 can also be one or more of a number of output mechanisms. In some instances, multimodal systems enable a user to provide multiple types of input to communicate with the computing device 1000. The communications interface 1080 generally governs and
manages the user input and system output. There is no restriction on operating on any particular hardware arrangement and therefore the basic features here may be substituted for improved hardware or firmware arrangements as they are developed.

[0062] For clarity of explanation, the embodiment of Figure 10 is presented as including individual functional blocks including functional blocks labeled as a "processor" or processor 1020. The functions these blocks represent may be provided through the use of either shared or dedicated hardware, including, but not limited to, hardware capable of executing software and hardware, such as a processor 1020, that is purpose-built to operate as an equivalent to software executing on a general purpose processor. For example, the functions of one or more processors presented in Figure 10 may be provided by a single shared processor or multiple processors. Use of the term "processor" should not be construed to refer exclusively to hardware capable of executing software. Illustrative embodiments may include a microprocessor and/or digital signal processor (DSP) hardware, read-only memory (ROM) 1040 for storing software performing the operations discussed below, and random access memory (RAM) 1050 for storing results. Very large scale integration (VLSI) hardware embodiments, as well as custom VLSI circuitry in combination with a general purpose DSP circuit, may also be provided.

[0063] The logical operations of the various embodiments are implemented as: (1) a sequence of computer implemented steps, operations, or procedures running on a programmable circuit within a general use computer, (2) a sequence of computer implemented steps, operations, or procedures running on a specific-use programmable circuit; and/or (3) interconnected machine modules or program engines within the programmable circuits. The system 1000 shown in Figure 10 can practice all or part of the recited methods, can be a part of the recited systems, and/or can operate according to instructions in the recited non-transitory computer-readable storage media. Such logical operations can be implemented as modules configured to control the processor 1020 to perform particular functions according to the programming of the module. For example, Figure 10 illustrates three modules Mod 1 1062, Mod 2 1064, and Mod 3 1066, which are modules configured to control the processor 1020. These modules may be stored on the storage device 1060 and loaded into RAM 1050 or memory 1030 at runtime or may be stored in other computer-readable memory locations.

[0064] Embodiments within the scope of the present disclosure may also include tangible and/or
non-transitory computer-readable storage media for carrying or having computer-executable
instructions or data structures stored thereon. Such non-transitory computer-readable storage
media can be any available media that can be accessed by a general purpose or special purpose
computer, including the functional design of any special purpose processor as discussed above.

By way of example, and not limitation, such non-transitory computer-readable media can include
RAM, ROM, EEPROM, CD-ROM or other optical disk storage, magnetic disk storage or other
magnetic storage devices, or any other medium which can be used to carry or store desired
program code means in the form of computer-executable instructions, data structures, or
processor chip design. When information is transferred or provided over a network or another
communications connection (either hardwired, wireless, or combination thereof) to a computer,
the computer properly views the connection as a computer-readable medium. Thus, any such
connection is properly termed a computer-readable medium. Combinations of the above should
also be included within the scope of the computer-readable media.

[0065] Computer-executable instructions include, for example, instructions and data which cause
a general purpose computer, special purpose computer, or special purpose processing device to
perform a certain function or group of functions. Computer-executable instructions also include
program modules that are executed by computers in stand-alone or network environments.
Generally, program modules include routines, programs, components, data structures, objects,
and the functions inherent in the design of special-purpose processors, etc. that perform
particular tasks or implement particular abstract data types. Computer-executable instructions,
associated data structures, and program modules represent examples of the program code means
for executing steps of the methods disclosed herein. The particular sequence of such executable
instructions or associated data structures represent examples of corresponding acts for
implementing the functions described in such steps.

[0066] Various embodiments of the disclosure may be practiced in network computing
environments with many types of computer system configurations, including personal
computers, hand-held devices, multi-processor systems, microprocessor-based or programmable
consumer electronics, network PCs, minicomputers, mainframe computers, and the like.
Embodiments may also be practiced in distributed computing environments where tasks are
performed by local and remote processing devices that are linked (either by hardwired links,
wireless links, or by a combination thereof) through a communications network. In a distributed computing environment, program modules may be located in both local and remote memory storage devices.

[0067] The various embodiments described above are provided by way of illustration only and should not be construed to limit the scope of the disclosure.
CLAIMS

1. A method of remote gaming, comprising:
   receiving at least one control command from at least one electronic device;
   transmitting the at least one control command to at least one logic controller;
   generating at least one random number sequence by at least one mechanical apparatus in
   response to at least one instruction by the at least one controller;
   capturing data pertaining to the at least one random number sequence; and
   transmitting information related to the captured data to at least one remote server.

2. The method of claim 1, further comprising transmitting the information related to the
   captured data to the at least one electronic device.

3. The method of claim 1, further comprising time-stamping and storing the information
   relating to the captured data for subsequent retrieval.

4. The method of claim 3, further comprising:
   receiving a request for random number data from at least one second electronic device;
   and
   transmitting to the at least one second electronic device, the most recently time-stamped
   and stored information which was not previously transmitted to an electronic device.

5. The method of claim 1, wherein the at least one electronic device is one of a mobile
   communications device, a personal computer and a laptop computer.

6. The method of claim 1, wherein the mechanical apparatus includes at least two six-sided
   dice and the at least one random number sequence is displayed on the dice.

7. The method of claim 1, wherein the capturing of data is performed by a system which
   includes at least one visual or optical sensor.
8. The method of claim 1, wherein the capturing of data is performed by a system which includes at least one camera apparatus.

9. The method of claim 1, wherein accuracy of the captured data is verified by an electro-mechanical device prior to transmitting that the captured data.

10. The method of claim 1, wherein the captured data is stored locally, for subsequent retrieval, on a machine housing the at least one mechanical apparatus and housing the system which includes the at least one camera apparatus.

11. The method of claim 10, further comprising:
   comparing the captured data stored on the machine and the corresponding captured data stored on the at least one remote server; and
   transmitting the corresponding captured data to the at least one electronic device, if captured data stored on the machine and corresponding captured data stored on the at least one remote server is the same.

12. A system for remote gaming, the system including:
   a processor;
   a non-transitory computer readable storage medium storing instructions for controlling the processor to perform steps comprising:
   receiving at least one control command from at least one electronic device;
   transmitting the at least one control command to at least one logic controller;
   generating at least one random number sequence by at least one mechanical apparatus in response to at least one instruction by the at least one controller;
   capturing data pertaining to the at least one random number sequence; and
   transmitting information related to the captured data to at least one remote server.
13. The system of claim 12, the non-transitory computer readable storage medium further storing instructions for controlling the processor to perform steps comprising:

transmitting the information related to the captured data to the at least one electronic device.

14. The system of claim 12, the non-transitory computer readable storage medium further storing instructions for controlling the processor to perform steps comprising:

time-stamping and storing the information relating to the captured data for subsequent retrieval.

15. The system of claim 14, the non-transitory computer readable storage medium further storing instructions for controlling the processor to perform steps comprising:

receiving a request for random number data from at least one second electronic device; and

transmitting to the at least one second electronic device, the most recently time-stamped and stored information which was not previously transmitted to an electronic device.

16. The method of claim 1, wherein the at least one electronic device is one of a mobile communications device, a personal computer and a laptop computer.

17. The system of claim 12, wherein the mechanical apparatus includes at least two six-sided dice and the at least one random number sequence is displayed on the dice.

18. The system of claim 12, wherein the capturing of data is performed by a system which includes at least one visual or optical sensor.

19. The system of claim 12, wherein the capturing of data is performed by a system which includes at least one camera apparatus.

20. The system of claim 12, wherein accuracy of the captured data is verified by an electro-mechanical device prior to transmitting that the captured data.
21. The system of claim 12, wherein the captured data is stored locally, for subsequent retrieval, on a machine housing the at least one mechanical apparatus and housing the system which includes the at least one camera apparatus.

22. The system of claim 21, the non-transitory computer readable storage medium further storing instructions for controlling the processor to perform steps comprising:
   comparing the captured data stored on the machine and the corresponding captured data stored on the at least one remote server; and
   transmitting the corresponding captured data to the at least one electronic device, if captured data stored on the machine and corresponding captured data stored on the at least one remote server is the same.

23. A non-transitory computer-readable storage medium storing instructions which, when executed by an electronic system, cause the electronic system to:
   receive at least one control command from at least one electronic device;
   transmit the at least one control command to at least one logic controller;
   generate at least one random number sequence by at least one mechanical apparatus in response to at least one instruction by the at least one controller;
   capture data pertaining to the at least one random number sequence; and
   transmit information related to the captured data to at least one remote server.

24. The non-transitory computer-readable storage medium of 23, further storing instructions which, when executed by the electronic system, cause the electronic system to:
   transmit the information related to the captured data to the at least one electronic device.

25. The non-transitory computer-readable storage medium of 23, further storing instructions which, when executed by the electronic system, cause the electronic system to:
   time-stamp and store the information relating to the captured data for subsequent retrieval.
26. The non-transitory computer-readable storage medium of 25, further storing instructions which, when executed by the electronic system, cause the electronic system to:
   receive a request for random number data from at least one second electronic device; and
   transmit to the at least one second electronic device, the most recently time-stamped and stored information which was not previously transmitted to an electronic device.

27. The non-transitory computer-readable storage medium of 23, wherein the at least one electronic device is one of a mobile communications device, a personal computer and a laptop computer.

28. The non-transitory computer-readable storage medium of 23, wherein the mechanical apparatus includes at least two six-sided dice and the at least one random number sequence is displayed on the dice.

29. The non-transitory computer-readable storage medium of 23, wherein the capturing of data is performed by a system which includes at least one visual or optical sensor.

30. The non-transitory computer-readable storage medium of 23, wherein the capturing of data is performed by a system which includes at least one camera apparatus.

31. The non-transitory computer-readable storage medium of 23, wherein accuracy of the captured data is verified by an electro-mechanical device prior to transmitting that the captured data.

32. The non-transitory computer-readable storage medium of 23, wherein the captured data is stored locally, for subsequent retrieval, on a machine housing the at least one mechanical apparatus and housing the system which includes the at least one camera apparatus.
33. The non-transitory computer-readable storage medium of 32, the non-transitory computer-readable storage medium further storing instructions, which when executed by the electronic system, cause the system to:

- compare the captured data stored on the machine and the corresponding captured data stored on the at least one remote server; and
- transmit the corresponding captured data to the at least one electronic device, if captured data stored on the machine and corresponding captured data stored on the at least one remote server is the same.

34. The method of claim 1, wherein the mechanical apparatus includes at least one multi-sided die and the at least one random number sequence is displayed on the die.

35. The method of claim 1, wherein the mechanical apparatus includes at least one multi-sided object displaying a matrix barcode on at least one of its sides and the at least one random number sequence is displayed as information contained within the at least one matrix barcode.
FIG. 1

100
2

Control Command Input on Remote Device

Control Command Sent to PLC

Machine Generates Random Number

Random Number Data Captured

Captured Data Sent to Endpoints
FIG. 2
FIG. 3
FIG. 5
FIG. 7

1. Shooter places bet
2. Players bet against shooter
3. Players may place side bets amongst each other
4. Shooter rolls dice until result is determined
5. Bets are paid to winners
FIG. 8
FIG. 9