



US 20120142425A1

(19) **United States**(12) **Patent Application Publication**  
**Scott et al.**(10) **Pub. No.: US 2012/0142425 A1**(43) **Pub. Date: Jun. 7, 2012**(54) **LEGACY GAME DOWNLOAD AND  
CONFIGURATION**

- (75) Inventors: **Thomas Scott**, Henderson, NV  
(US); **Christopher D. Barton**,  
Henderson, NV (US); **Mettu  
Ramakrishna Reddy**, Brookfield,  
WI (US)
- (73) Assignee: **Bally Gaming, Inc.**, Las Vegas, NV  
(US)
- (21) Appl. No.: **13/280,087**
- (22) Filed: **Oct. 24, 2011**

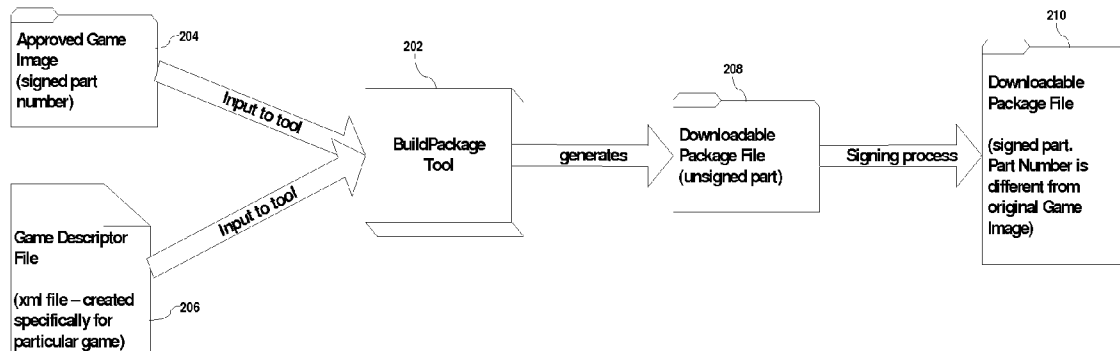
**Related U.S. Application Data**

- (60) Provisional application No. 61/406,028, filed on Oct.  
22, 2010.

**Publication Classification**

- (51) **Int. Cl.**  
**A63F 9/24** (2006.01)
- (52) **U.S. Cl.** ..... **463/42**
- (57) **ABSTRACT**

A system includes a download configuration host server coupled to an electronic gaming machine which, in turn, comprises an operating system. A method for configuring a legacy game application on the electronic gaming machine comprises the following steps. A downloadable package comprising a game image of the legacy game and a game descriptor file is generated. The downloadable package is communicated from the download configuration host server to the electronic gaming machine. The legacy game image is installed on the electronic gaming machine. The installed legacy game is configured by the operating system. The operating system simulates configuration of the installed legacy game by an operator in response to data in the game descriptor file. The installed legacy game is then enabled for game play.



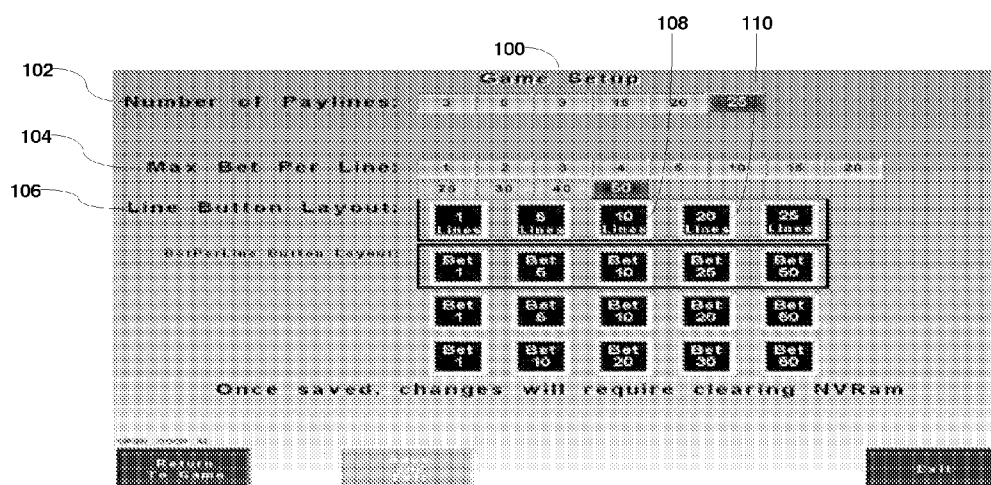


Fig. 1

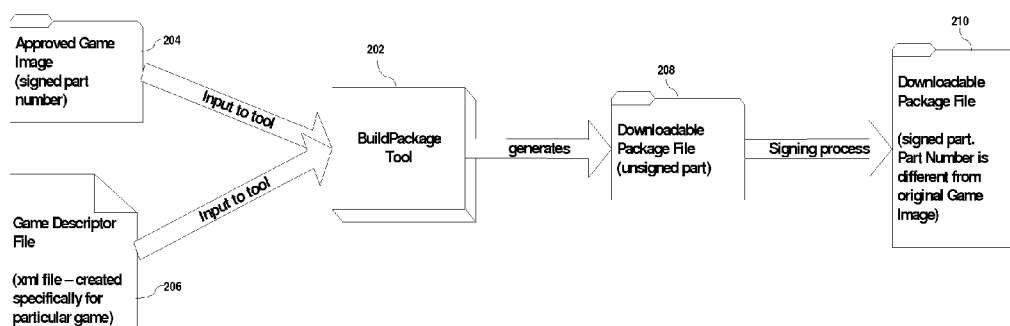
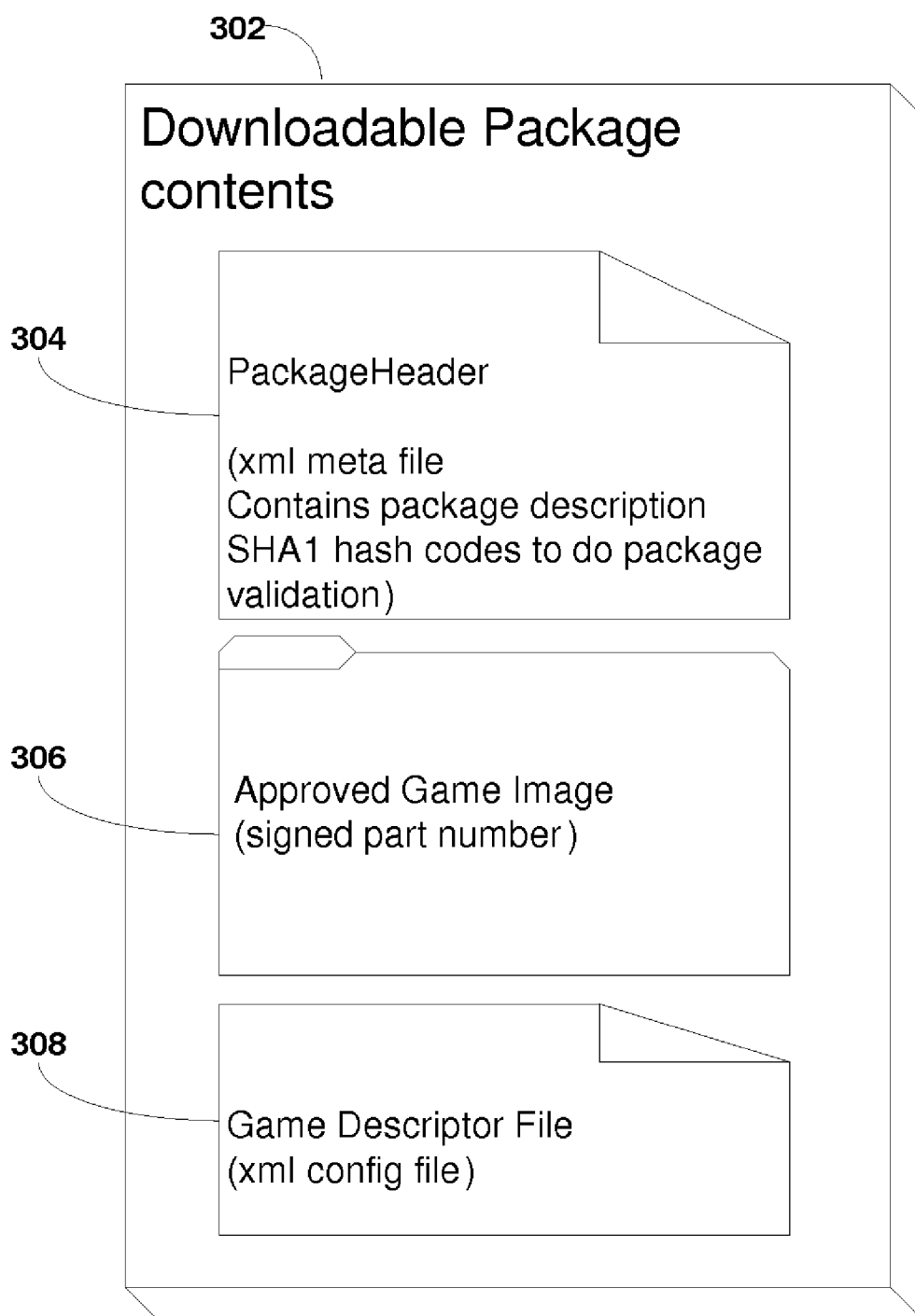


Fig. 2

**Fig. 3**

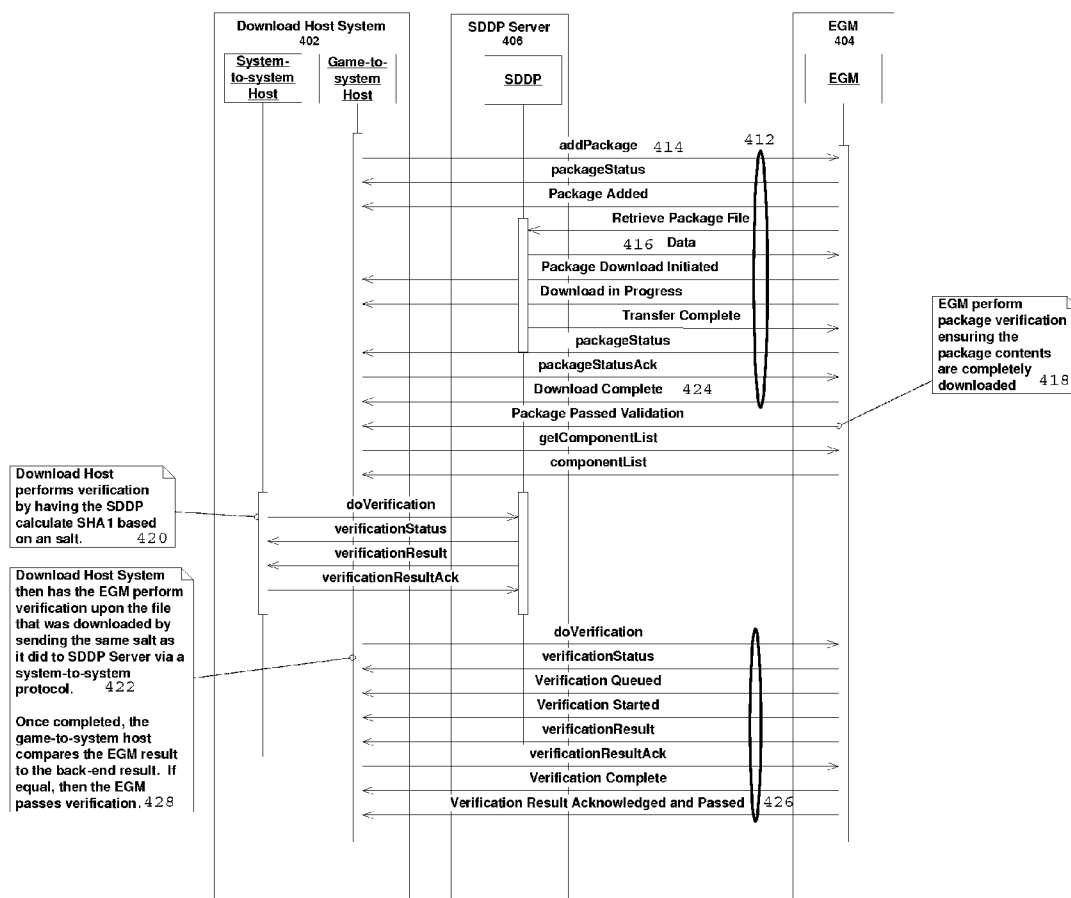


Fig. 4

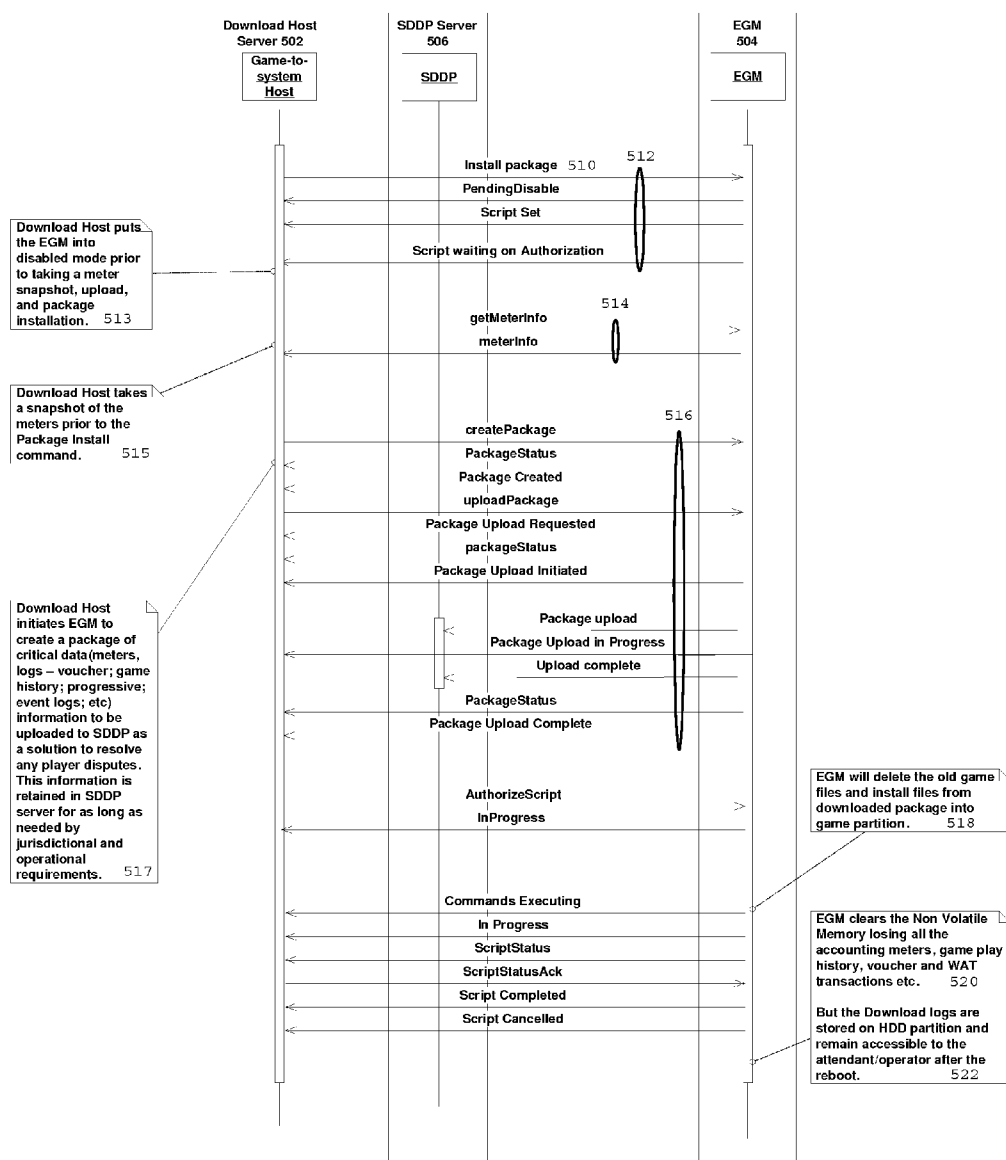


Fig. 5

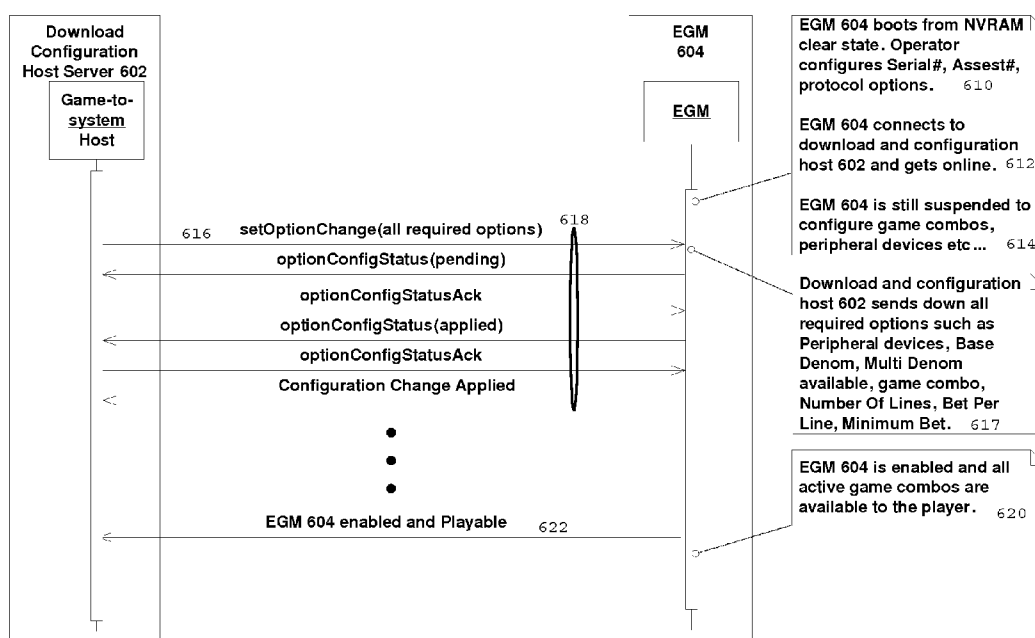


Fig. 6

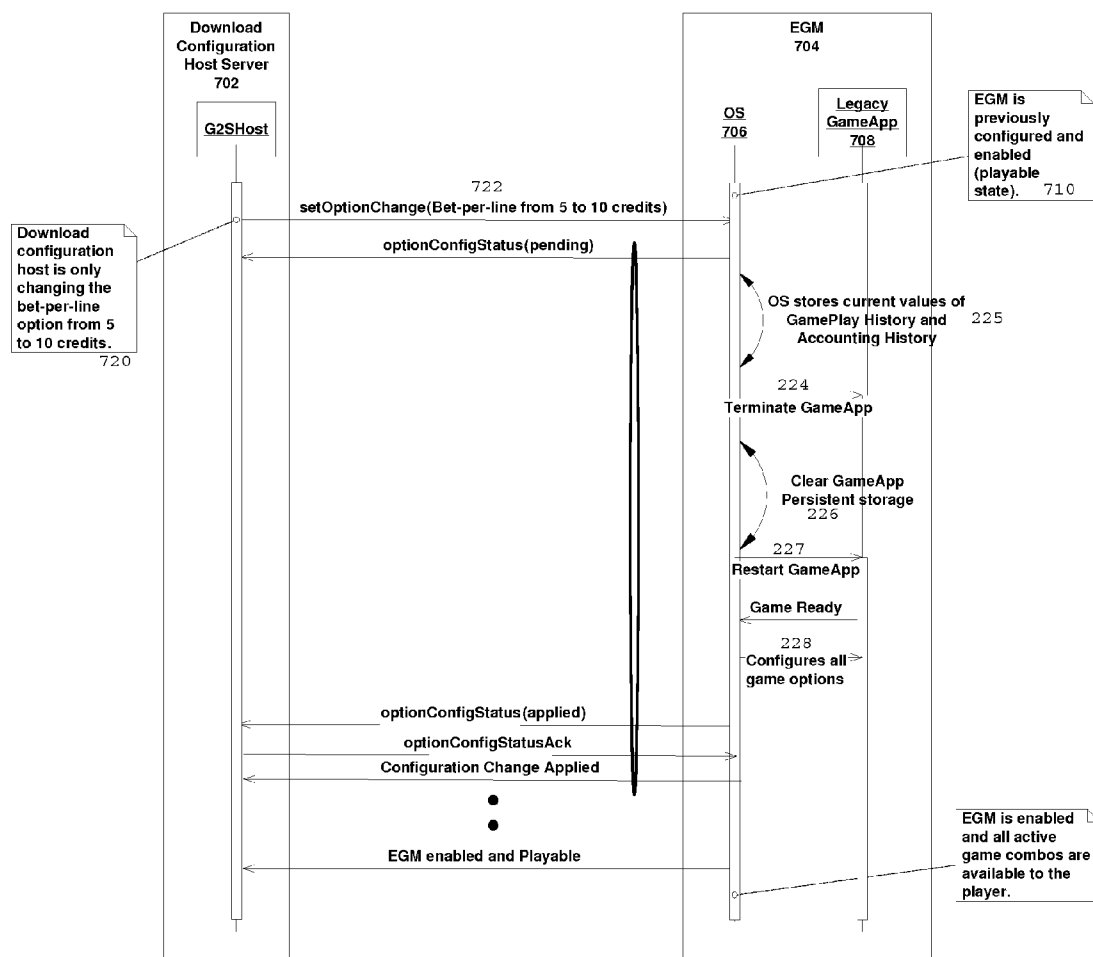


Fig. 7

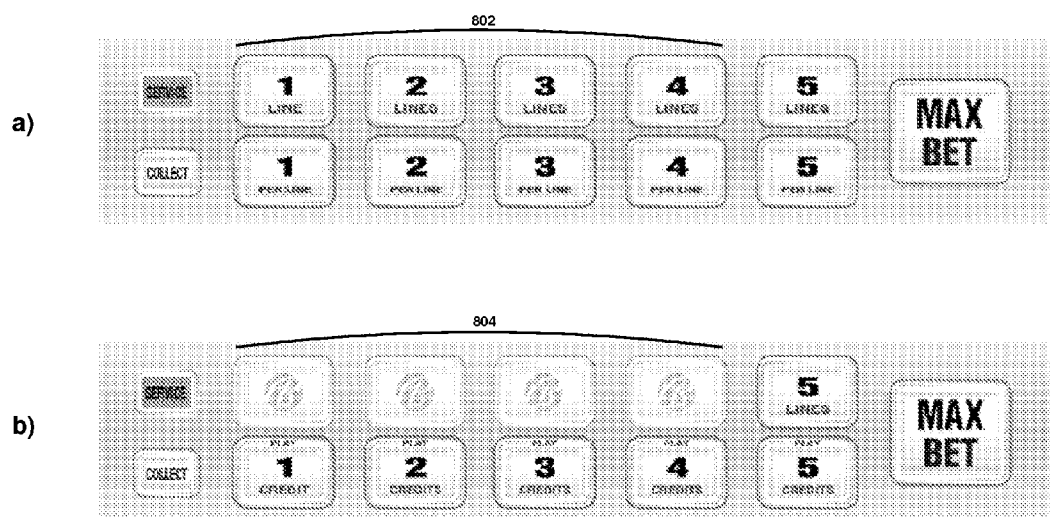


Fig. 8

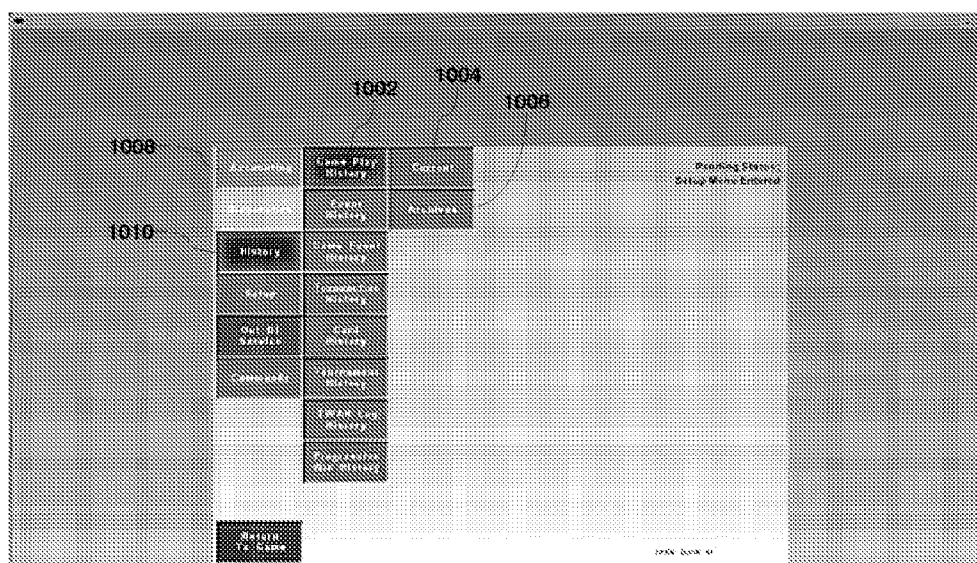
0128442381.f32

↑ Display ID

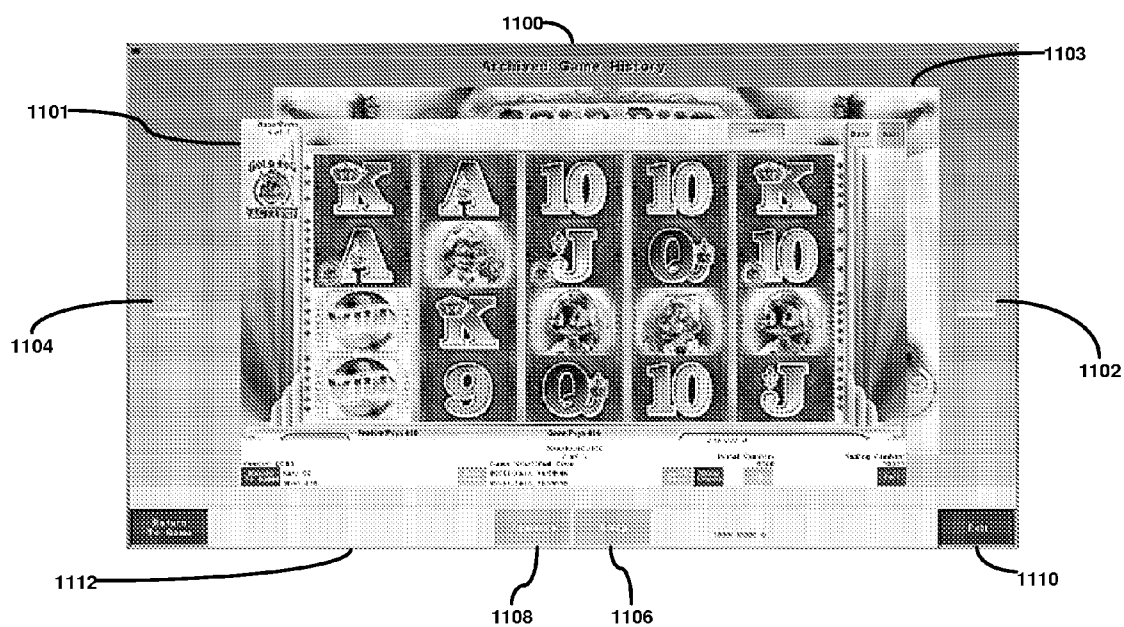
↑ Date/Time Stamp

↑ File extension

Fig. 9



**Fig. 10**



**Fig. 11**

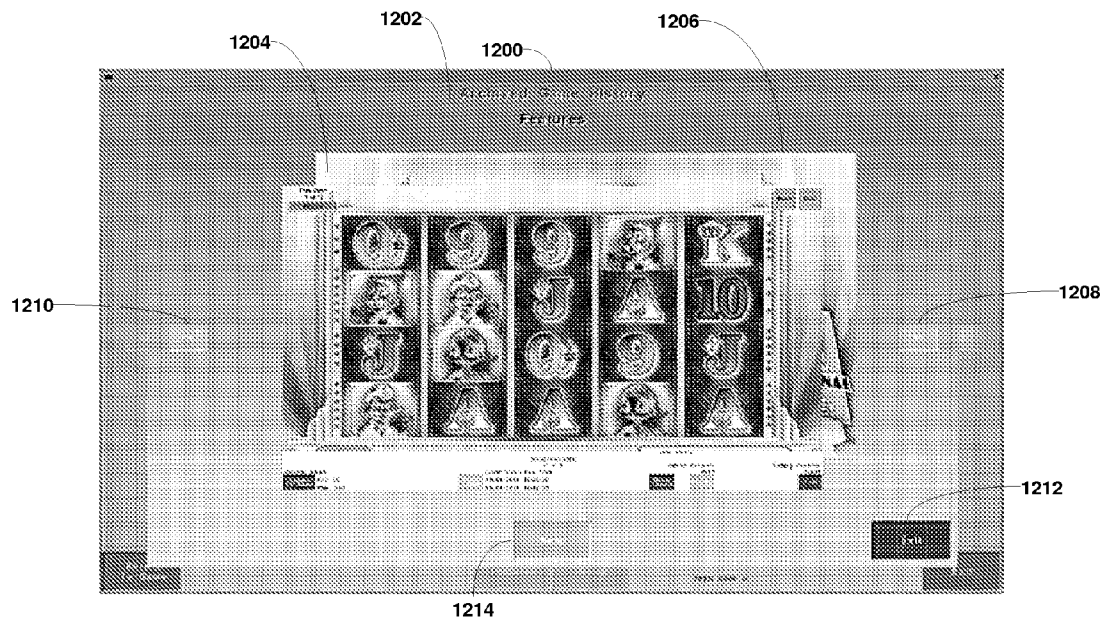


Fig. 12



Fig. 13

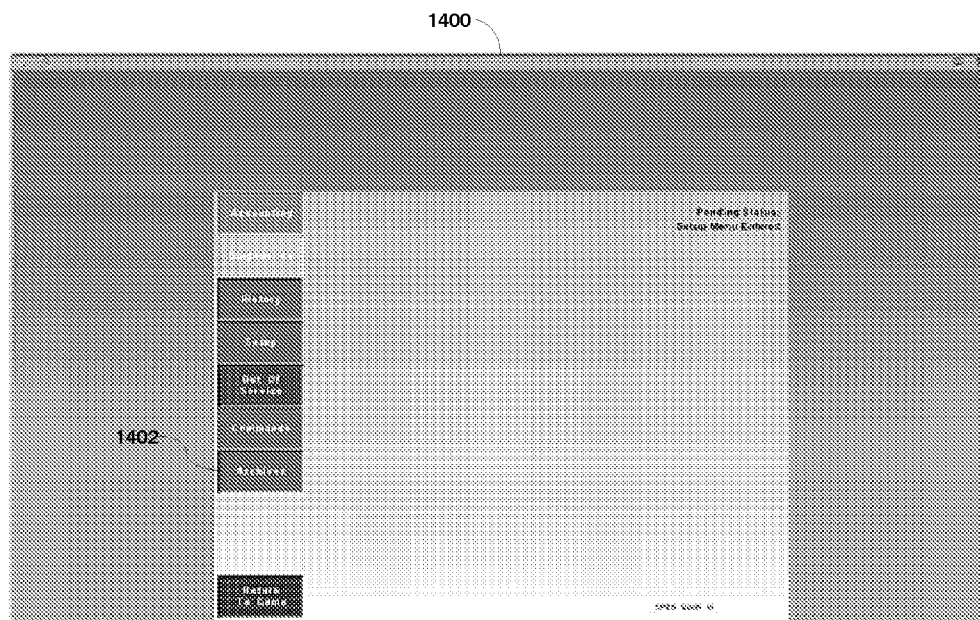


Fig. 14

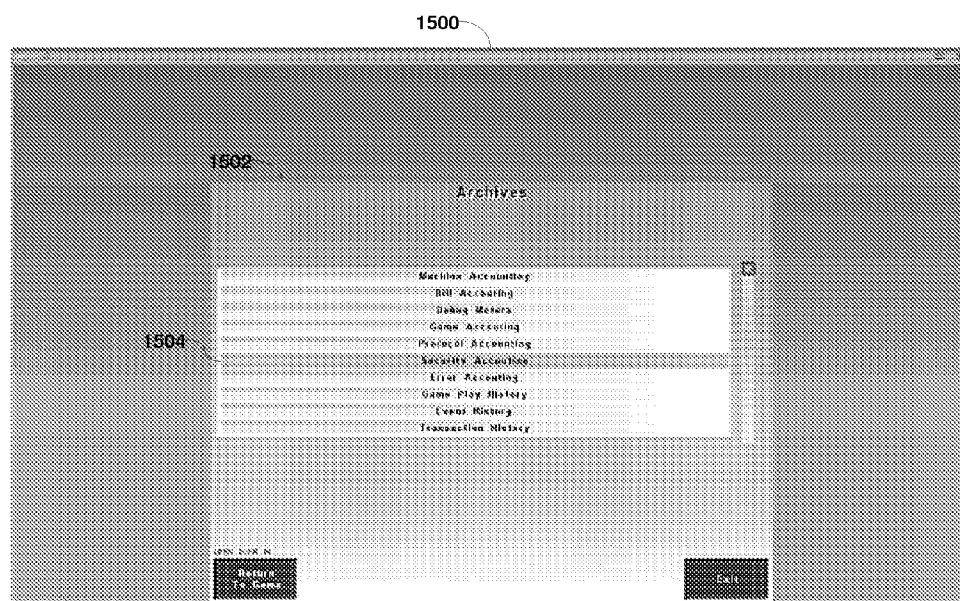
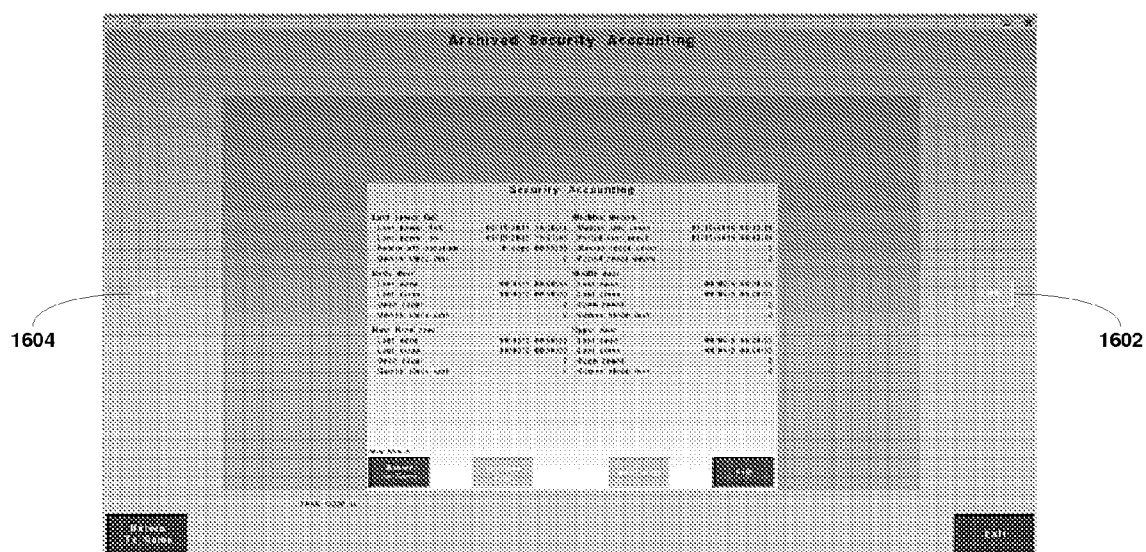


Fig. 15



**Fig. 16**

```

<descriptors version="1.0">
  <descriptor type="game_auto_setup"> 1702
    <parts>
      <part id="AVGHKPK01000-00" /> 1704
    </parts>
    <history> 1706
      <capture start="feature1" displays="2" transition="500">
        <screen id="feature1" type="feature" next="feature2">
          <record type="feature" length="0" />
          <button type="enter" text="feat1" x="0.777187" y="0.032564" /> 1710
        </screen>
        <screen id="feature2" type="feature" next="main">
          <record type="cycle_wins" length="5" />
          <button type="next" text="Next" x="0.979411" y="0.045577" /> 1712
          <button type="return" text="Return" x="0.057307" y="0.070320" /> 1714
        </screen>
        <screen id="main" type="game" next="feature1">
          <record type="cycle_wins" length="5" />
          <button type="next" text="Prior" x="0.348485" y="0.964870" /> 1716
        </screen>
      </capture>
    </history> 1708
    <options>
      <option name="Max Lines" type="enum">
        <description>
          Select the maximum number of paylines to play.
        </description>
        <enums>
          <enum value="20" actions="20lines" />
          <enum value="25" actions="25lines" />
        </enums>
      </option>
      <option name="Max Bet Per Line" type="enum">
        <description>
          Select the maximum bet per payline to play.
        </description>
        <enums>
          <enum value="40" actions="40BPL" />
          <enum value="50" actions="50BPL" />
        </enums>
      </option>
      <option name="Min Bet" type="integer">
        <description>
          Select the minimum bet per game play.
        </description>
        <enums />
      </option>
    </options>
    <actions>
      <action id="_init">
        <touch x="0.106244" y="0.368341" /> <!-- setup -->
        <touch x="0.226234" y="0.246701" /> <!-- game setup -->
        <!-- select the layout, language from the iDeck -->
        <option name="Language" category="/os/ds/ibd" value="English" />
        <option name="Layout" category="/os/ds/ibd" value="15-button 1-plug" />
        <!-- setup the non-changing buttons -->
        <option name="Label" category="/os/ds/ibd/btn/btn01" value="Small-Service" />
        <option name="Label" category="/os/ds/ibd/btn/btn02" value="Small-Collect" />
        <option name="Label" category="/os/ds/ibd/btn/btn13" value="Large-MaxBet" />
      </action>
    </actions>
  </descriptor>
</descriptors>

```

Fig. 17

```
<action id="_exit">
  <touch x="0.348729" y="0.930046" /> <!-- save -->
  <touch x="0.132454" y="0.945015" /> <!-- return to game -->
</action>
<action id="20lines">
  <touch x="0.632454" y="0.088343" />
  <option name="Label" category="/os/ds/ibd/btn/btn03" value="Medium-Line" />
  <option name="Text" category="/os/ds/ibd/btn/btn03" value="1" />
  <option name="Label" category="/os/ds/ibd/btn/btn04" value="Medium-Lines" />
  <option name="Text" category="/os/ds/ibd/btn/btn04" value="5" />
  <option name="Label" category="/os/ds/ibd/btn/btn05" value="Medium-Lines" />
  <option name="Text" category="/os/ds/ibd/btn/btn05" value="10" />
  <option name="Label" category="/os/ds/ibd/btn/btn06" value="Medium-Lines" />
  <option name="Text" category="/os/ds/ibd/btn/btn06" value="15" />
  <option name="Label" category="/os/ds/ibd/btn/btn07" value="Medium-Lines" />
  <option name="Text" category="/os/ds/ibd/btn/btn07" value="20" />
</action>
<action id="25lines">
  <touch x="0.701246" y="0.085044" />
  <option name="Label" category="/os/ds/ibd/btn/btn03" value="Medium-Line" />
  <option name="Text" category="/os/ds/ibd/btn/btn03" value="1" />
  <option name="Label" category="/os/ds/ibd/btn/btn04" value="Medium-Lines" />
  <option name="Text" category="/os/ds/ibd/btn/btn04" value="5" />
  <option name="Label" category="/os/ds/ibd/btn/btn05" value="Medium-Lines" />
  <option name="Text" category="/os/ds/ibd/btn/btn05" value="10" />
  <option name="Label" category="/os/ds/ibd/btn/btn06" value="Medium-Lines" />
  <option name="Text" category="/os/ds/ibd/btn/btn06" value="20" />
  <option name="Label" category="/os/ds/ibd/btn/btn07" value="Medium-Lines" />
  <option name="Text" category="/os/ds/ibd/btn/btn07" value="25" />
</action>
<action id="40BPL">
  <touch x="0.503727" y="0.290017" />
  <option name="Label" category="/os/ds/ibd/btn/btn08" value="Medium-Credit" />
  <option name="Text" category="/os/ds/ibd/btn/btn08" value="1" />
  <option name="Label" category="/os/ds/ibd/btn/btn09" value="Medium-Credits" />
  <option name="Text" category="/os/ds/ibd/btn/btn09" value="5" />
  <option name="Label" category="/os/ds/ibd/btn/btn10" value="Medium-Credits" />
  <option name="Text" category="/os/ds/ibd/btn/btn10" value="10" />
  <option name="Label" category="/os/ds/ibd/btn/btn11" value="Medium-Credits" />
  <option name="Text" category="/os/ds/ibd/btn/btn11" value="25" />
  <option name="Label" category="/os/ds/ibd/btn/btn12" value="Medium-Credits" />
  <option name="Text" category="/os/ds/ibd/btn/btn12" value="40" />
</action>
<action id="50BPL">
  <touch x="0.573741" y="0.286718" />
  <option name="Label" category="/os/ds/ibd/btn/btn08" value="Medium-Credit" />
  <option name="Text" category="/os/ds/ibd/btn/btn08" value="1" />
  <option name="Label" category="/os/ds/ibd/btn/btn09" value="Medium-Credits" />
  <option name="Text" category="/os/ds/ibd/btn/btn09" value="5" />
  <option name="Label" category="/os/ds/ibd/btn/btn10" value="Medium-Credits" />
  <option name="Text" category="/os/ds/ibd/btn/btn10" value="10" />
  <option name="Label" category="/os/ds/ibd/btn/btn11" value="Medium-Credits" />
  <option name="Text" category="/os/ds/ibd/btn/btn11" value="25" />
  <option name="Label" category="/os/ds/ibd/btn/btn12" value="Medium-Credits" />
  <option name="Text" category="/os/ds/ibd/btn/btn12" value="50" />
</action>
<action id="2ndLayout">
  <touch x="0.584983" y="0.580034" />
</action>
</actions>
<dependencies />
</descriptor>
</descriptors>
```

Fig. 18

## LEGACY GAME DOWNLOAD AND CONFIGURATION

### FIELD OF THE INVENTION

**[0001]** The present invention relates to creating and downloading game packages, and in particular for creating and downloading game packages for legacy games.

### BACKGROUND OF THE INVENTION

**[0002]** Advances in the design and implementation of gaming machines sometimes prevent older games from operating in newer gaming machines. New hardware and/or software is developed for gaming machines to provide new and/or improved game play experiences for players, and capabilities for owners and operators. For example, operating systems controlling the operation of the processor or processors in gaming machines are updated to provide new such capabilities. New games may then be designed and implemented to take advantage of these new capabilities. However, in some cases, older games, termed legacy games in the present application, which were not designed to use the new capabilities, are unable to take advantage of them. In such a situation, a legacy game may need to be reprogrammed to take advantage of the new operating system.

**[0003]** More specifically, an improved operating system, and cooperating new slot machine game program, may permit configuration options of the game to be dynamically and/or remotely changed without requiring a reinstallation of the game. However, in a legacy slot machine game, configuration data, such as maximum lines, maximum wager per line, and so forth, is set during installation of the game and is unchangeable without uninstalling (deleting) and then reinstalling the game in the gaming machine. Thus, legacy games are currently not able to cooperate with the new operating system to provide dynamic configuration option change.

**[0004]** Often, however, the legacy games are still popular and viable, and require no changes in game play operation. In addition, in most jurisdictions, game software must be submitted to and approved by regulatory bodies, which then require that the approved software be cryptographically signed to prevent modification. If a new game program is implemented, recreating the legacy game, which operates with the new operating system, it must be submitted to the regulatory bodies, which must analyze and approve it. This typically takes time and money. For a game not requiring any changes to the game play this is a waste of time and money.

**[0005]** It is, therefore, desirable to permit a legacy game to be able to be installed and operated in a gaming machine including new and/or improved features for which that game was not designed; more specifically a new operating system including features with which the legacy game was not designed to cooperate. Such a system will permit the use of already existing and approved games on gaming machines using new hardware and/or software. This will permit continued use of popular and viable legacy games without requiring redesign, reimplementation and resubmission to regulatory bodies of reprogrammed legacy games.

### BRIEF DESCRIPTION OF THE INVENTION

**[0006]** In accordance with principles of the present invention, a system includes a download configuration host server coupled to an electronic gaming machine which, in turn, comprises an operating system. A method for configuring a

legacy game application on the electronic gaming machine comprises the following steps. A downloadable package comprising a game image of the legacy game and a game descriptor file is generated. The downloadable package is communicated from the download configuration host server to the electronic gaming machine. The legacy game image is installed on the electronic gaming machine. The installed legacy game is configured by the operating system. The operating system simulates configuration of the installed legacy game by an operator in response to data in the game descriptor file. The installed legacy game is then enabled for game play.

### BRIEF DESCRIPTION OF THE DRAWING

**[0007]** FIG. 1 is a screen display illustrating a game setup menu for a legacy game;

**[0008]** FIG. 2 is a flow diagram illustrating the process for creating a download package for legacy games according to principles of the present invention;

**[0009]** FIG. 3 is a content diagram illustrating the contents of a downloadable package for a legacy game according to principles of the present invention;

**[0010]** FIG. 4 is a sequence diagram illustrating the process of downloading a downloadable package as illustrated in FIG. 3 to a gaming machine according to principles of the present invention;

**[0011]** FIG. 5 is a sequence diagram illustrating the process of installing a legacy game contained in a downloadable package as illustrated in FIG. 3 in a gaming machine according to principles of the present invention;

**[0012]** FIG. 6 is a sequence diagram illustrating the process of performing initial configuration of a legacy game previously installed in a gaming machine from a downloadable package as illustrated in FIG. 3 according to principles of the present invention;

**[0013]** FIG. 7 is a sequence diagram illustrating the process of reconfiguring a legacy game previously installed in a gaming machine from a downloadable package as illustrated in FIG. 3 according to principles of the present invention;

**[0014]** FIG. 8*a* and *b* are screen displays illustrating a button panel displayed by a legacy game previously installed in a gaming machine from a downloadable package as illustrated in FIG. 3 in a first and second configuration, respectively, as reconfigured according to principles of the present invention

**[0015]** FIG. 9 is a diagram illustrating the format of a file name for archiving game play history for a legacy game according to principles of the present invention;

**[0016]** FIG. 10 is a screen display illustrating the location of icons on a game play history selection screen on a legacy game according to principles of the present invention;

**[0017]** FIG. 11 is a screen display illustrating images representing the contents of a game play history file according to principles of the present invention;

**[0018]** FIG. 12 is a screen display illustrating images representing the features active in main screen in a game play history file according to principles of the present invention;

**[0019]** FIG. 13 is a screen display illustrating images representing the features active in an upper screen in a game play history file according to principles of the present invention;

**[0020]** FIG. 14 is a screen display illustrating the location of icons on an operating system status screen according to principles of the present invention;

[0021] FIG. 15 is a screen display illustrating the location of buttons on an operating system archive selection screen according to principles of the present invention;

[0022] FIG. 16 is a screen display illustrating security accounting information according to principles of the present invention; and

[0023] FIG. 17 and FIG. 18 combined is a listing of a game descriptor file which is part of a downloadable package according to principles of the present invention.

#### DETAILED DESCRIPTION

[0024] It is desirable to create downloadable packages for legacy games which permit those legacy games to operate with a later, more advanced, operating systems and/or game download and configuration systems. Such download and configuration systems permit configuration changes to be made to electronic gaming machines (EGMs) on a casino floor in real time. Previously approved legacy game image files are used to create downloadable packages.

[0025] In general, a game play system, e.g. in a casino, includes a download configuration host server coupled to an electronic gaming machine which, in turn, comprises an operating system. A method for configuring a legacy game application on the electronic gaming machine comprises the following steps. A downloadable package comprising a game image of the legacy game and a game descriptor file is generated. The downloadable package is communicated from the download configuration host server to the electronic gaming machine. The legacy game image is installed on the electronic gaming machine. The installed legacy game is configured by the operating system. The operating system simulates configuration of the installed legacy game by an operator in response to data in the game descriptor file. The installed legacy game is then enabled for game play.

[0026] In the present application, the term 'game image file' refers to a file which includes information defining a game which has been approved by regulatory agencies and has been cryptographically signed to prevent alteration. A game image file for a legacy game is combined with a game option xml file called "Game Descriptor File" (GDF) to create a file called a "Download Package".

[0027] Data in the GDF permits the operating system to interact with the legacy game. Repacking the legacy game image into a downloadable package file with a GDF gives the ability to release legacy popular game titles with a new and/or improved download and configuration host to provide game content that is downloadable and remotely configurable from a backend system. The present application describes creating a downloadable package from a previously approved game image, installing a game package in an EGM, and remotely configuring game options.

[0028] As is well known, electronic gaming machines include one or more computers, each including a processor, memory, and input/output devices, coupled together via a computer bus, and operating under control of one or more executable programs. The memory is a combination of read-write memory (RAM), read-only memory (ROM), and mass storage devices, such as disk drives; optical drives such as CDs or DVDs; semiconductor mass storage; tape drives; and so forth. Output devices include display screens on which images are displayed representing game play. Other output devices include lights and sound producing devices such as speakers, buzzers, bells, etc. Input devices include buttons for receiving instruction from a player related to game play.

Other input devices include coin receivers, bill receivers, and/or ticket receivers for receiving wagers from players. Another input device is one or more touch screens, which may be placed atop one or more of the output devices, or atop a player panel, for receiving instruction from a player related to game play. The computer may also include communication input/output devices such as network interfaces which can receive data from a network and transmit data to the network, or wireless communication devices.

[0029] An executable program, when executed by the processor, receives input instructions from the player through the input devices, performs calculations in the processor according to the approved game executable program, and generates output displays and sounds representing the results of game play on the output devices. A control executable program, generally termed an operating system, controls access to the computer hardware from other executable programs, such as game programs, and other generally available functions. Other systems in a gaming system, such as download configuration host servers, progressive servers, administration servers, accounting servers, and so forth, also include one or more computers which are constructed and operated as described just above.

[0030] Typically, a legacy game requires its options to be configured during initial installation on an EGM before the game to become enabled and playable. The value of these options are typically displayed by the legacy game on an operator menu screen. FIG. 1 is a screen display 100 illustrating an example of such a menu screen entitled "Game Setup". As illustrated in FIG. 1, typical options which may be configured include "Number of Paylines" 102, "Max Bet per Line" 104, "Line Button Layout" 106, and so forth. Other options, not shown to simplify the figure, may include "Max Lines", "Double up" and "Video Output", etc. The game specific menu screen 100 is displayed and controlled by the legacy game software directly. In legacy game machines, these options are accessed by an operator by touching the screen at the location of the button desired to operate.

[0031] For the new operating system (OS) to support download and configuration of legacy games from e.g. a remote host, the OS needs information related to the display of the option selectors on the game setup display screens such as screen 100. This permits access to those game setup display screens to change of these options as desired. The Game Descriptor File (GDF) is an xml file that is created for the legacy game and contains information related to the display of game options on the Game Setup menu. FIG. 17 and FIG. 18 display a listing of a sample Game Descriptor File, and will be described in more detail below.

[0032] The Game Descriptor File contains co-ordinates (x, y) of the buttons and/or other widgets shown in the Game Setup Menu 100. The co-ordinates of a button relate to the location of that button on the game setup display. Typically, this is measured from the left edge for the x co-ordinate, and from the top edge for the y co-ordinate. The measurements are also typically in pixels, though other measurement units may be used instead. The Game Descriptor File may also contain other data related to the buttons and/or widgets such as name, identifiers, etc.

[0033] If, for example, the download and configuration system desires to change the "Max Bet Per Line" 104 option value from 10 (108) to 20 (110), the OS retrieves the location of "Max Bet Per Line" 20 button 110 from the GDF file. The OS then simulates a touch on the game setup display screen

**100** at that location. This activates the “Max Bet Per Line” **20** button **110**, as if it were being manually touched by an operator at the EGM. This configuration activity is hidden from players by displaying a splash screen over the Game Setup screen **100** which includes a status message indicating that option changes are being applied.

**[0034]** Legacy games were developed, approved, and released prior to the development of new OSs and/or new download and configuration systems coupled between the OS and the slot machine game application. The new OS and/or new download and configuration system allow dynamic access to, and remote configuration of, game options. The download and configuration system provides the capability for the game to register its configuration options to the OS at runtime. The OS compiles a list of options for the legacy game and sends them to the download and configuration system through a communication link. In one embodiment, this communication link may use the game-to-system (G2S) protocol. The G2S communication standard protocol is promulgated by the Gaming Standards Association (GSA). More specifically, in such an embodiment, the option configuration device (OptionConfig), which is part of the G2S protocol, may be used to configure, or reconfigure options in a gaming machine.

**[0035]** Because a legacy game is not designed or implemented to cooperate with a download and configuration system, or to communicate using the G2S communication protocol, the legacy game, by itself, cannot be used with such a system. To bridge this gap between the legacy game and the OS, Game Descriptor File (GDF), with the appropriate options for each legacy game, is generated and included within a download package. The legacy game image (i.e. the data approved by regulatory bodies and cryptographically signed) is combined with the GDF file to create a downloadable package. There are no changes made to the legacy game image in the downloadable package. The legacy game image is reproduced exactly and compressed to reduce the size of download package file. Because no changes are made to the legacy game image, that game image may legitimately be used to control the EGM.

**[0036]** Referring to FIG. 2, a build package tool **202** receives an approved legacy game image **204** and the game descriptor file **206** as input data. The build package tool **202** generates an unsigned downloadable package file **208** as output data. This download package is then cryptographically signed to prevent undetected modification to produce a signed downloadable package file **210**. The signed downloadable package file **210** is downloaded to, and installed on, an EGM to permit the legacy game represented by the approved game image **204** to be played on the EGM.

**[0037]** Referring to FIG. 3, a downloadable package file **302** includes data **304** representing a package header. The package header **304** includes data representing a package description, hash codes for the downloadable package and other portions of the downloadable package. These hash codes are used to perform package verification of the package in a known manner. The downloadable package file also includes data representing the approved legacy game image **306** and the game descriptor file **308** associated with the legacy game image **306**.

**[0038]** After the downloadable package file **302** is created and signed, it is communicated to a Software Download and Distribution Point (SDDP) server where it is available to be downloaded to gaming machines. Creating and scheduling

the download and install tasks are done through a user interface to a download management host.

**[0039]** Referring to FIG. 4, various request, reply, transfer, command, status, and acknowledgement messages are exchanged among the download host system **402**, the SDDP server **406** and the EGM **404** to implement the legacy game package download, verification and operation, as described in detail below. When a download task is scheduled in the download host system **402**, it initiates the download **412** by sending a request **414** to add a download package (i.e. a legacy game) to the EGM **404**, i.e. to transfer a specific game package from the SDDP server **406** to the EGM **404**. The EGM **404** starts downloading the file **416** from the SDDP server **406**. In one embodiment, this download may be performed through secure http (https) connection. Other protocols and connections may also be used. One skilled in the art understands how to select and implement a transfer protocol.

**[0040]** After the game package is downloaded **424**, the EGM **404** performs a validation **418** of the package received from the SDDP server **406** to ensure that the download package is downloaded to the EGM **404** completely and without errors. In the illustrated embodiment, this is done using the known technique of calculating a result of a one-way function, e.g. a cryptographic hash function, applied to the downloaded file and comparing it to the expected value of the one-way function found in the package header **304** (of FIG. 3). If they are the same, then the file was downloaded in full and accurately.

**[0041]** The download host system **402** then performs an independent verification **420** of the download package on the SDDP server **406**. In the illustrated embodiment, this verification uses the known technique of generating a salt value (consisting of a predetermined number of random bits), combining the salt with the contents of the download package, then calculating a result of applying a one-way function, e.g. a cryptographic hash function, to the combination. The value of the resulting one-way function of the combination is kept.

**[0042]** The download host system **402** then directs the EGM **404** to perform a similar verification **422** of the file downloaded to the EGM **404**. The download host system **402** transmits the salt to the EGM **404** so the same salt is used by the EGM **404** to verify the contents of the download package at the EGM **404** as was used by the download host system **402** to verify the contents of the download package at the SDDP server **406**. When the EGM **404** has calculated the result of the hash function on the combination of the salt received from the download host system **402** and the download package received from the SDDP server, that result is transmitted back to the download host system. The download host system compares the result received from the EGM **404** with the result previously calculated at the download host system **402**. If the two results are equal, then the contents of the download package at the EGM are considered verified **428**.

**[0043]** In a legacy game package download process, the EGM stores the legacy game image, i.e. the approved signed part, **306** (of FIG. 3) in the game partition of the hard-disk drive (not shown) in the EGM. The game descriptor file, i.e. the xml file that describes game options **308** is stored in the critical-data partition, e.g. /critical\_data/descriptors of the hard-disk in the EGM **504**. The installation process involves retrieving the legacy game image from the hard-disk drive and loading it into the memory of the EGM in anticipation of being executed.

[0044] Referring to FIG. 5, after the game package is downloaded, the download host server 502 will send an install command 510 to the EGM 504 to install the package. Based on jurisdictional requirements of game idle time and zero credit conditions and so forth, EGM 504 will start installation of the package when those requirements are satisfied. When the conditions are met, the download host server 502 places the EGM 504 into the disabled mode 512, 513.

[0045] Before installation of the downloaded package, EGM 504 provides a snapshot 514, 515 of meter data related to game play history, as described below in FIG. 9, FIG. 10, FIG. 11, FIG. 12, and FIG. 13 and the associated written description below. The EGM 504 also requests 516, 517 a package of data related to accounting and/or transaction history, as described in FIG. 14, FIG. 15, and FIG. 16 and the associated written description below. The data is uploaded to the SDDP server 506, and archived in relatively long term storage. It may be used to resolve player disputes after the legacy game is installed in the EGM 504. The files associated with the downloaded legacy game are then retrieved from the hard disk drive and installed 518 into the memory of the EGM 504.

[0046] After the download of the legacy game and the retrieval and storage of history, meter and transaction data, and in preparation for the initiation of operation of the downloaded legacy game, the EGM 504 automatically clears 518 the contents of a non-volatile memory device (not shown) in the EGM 504 erasing previously stored critical information such as accounting meters, game history, and logs of vouchers, events, progressive contributions and wins, bills-in, etc. This critical information is stored 522 locally in the EGM 504 in a critical-data partition of the hard disk drive and retained to meet jurisdictional requirements.

[0047] To activate the newly downloaded legacy game, the EGM 504 performs an automatic processor reset. During the reboot process, the OS locates the game descriptor file (FIG. 3, 308) associated with the legacy game to be installed on the hard disk drive, and reads all the options from the xml file. Because data representing the screens presenting the configuration data, and the locations of buttons and/or other widgets on those screens is specified in the game descriptor file 308, the EGM 504 gets access to the game options. In one embodiment, this may be done through use of the G2S OptionConfig device, as is known. Other methods for performing this function exist, and one skilled in the art understands how to select, and to design and implement such methods, as appropriate.

[0048] FIG. 6 illustrates initially configuring an EGM 604. Referring to FIG. 6, the download host server 602 communicates with the EGM 604 to remotely configure the EGM 604 options. In one embodiment, the download host server 602 may use the G2S optionConfig device to configure the EGM 604. However, any protocol capable of remotely configuring the EGM 604 options may be used. One skilled in the art knows how to select, and to design and implement such a configuration protocol, as appropriate.

[0049] Some of the options that are expected to be configured initially from the download host server 602 include: device setup, credit setup, game combos, etc. Other options that are expected to be configured include game options such as: number of lines and bet per line, minimum bet, double up and any other game specific options. The minimum bet option is expected to change more frequently than other options in order to perform yield management. After configuration of

the legacy game options, the EGM 604 is enabled and the legacy game, or game combos, are made available to a player.

[0050] When the EGM 604 performs a reset, or reboot, it clears 610 the non-volatile RAM (NVRAM), as described above. The EGM 604 then connects 612 to the download configuration host server 602. Operation of the EGM 604 is suspended 614 giving the operator the opportunity to configure the EGM 604. Once connected, the download configuration host server 602 sends an option change message 616 to the EGM 604. This option change message 616 contains desired values for all the required options. The EGM updates the specified options, and informs the download configuration host server 602 when the configuration changes are applied 618. When the options received from the download configuration host server 602 have been applied and the EGM is ready to be playable 620, the EGM 604 sends a message 622, informing the download configuration host server 602.

[0051] As described above, it is desirable to support configuration and reconfiguration of legacy game options remotely and/or in real-time. The legacy game 708, itself, is not designed or implemented in a manner to allow such reconfiguration. However, the new OS 706 in the new EGM 704 is designed and implemented to handle configuration changes in legacy games 708.

[0052] FIG. 7 illustrates reconfiguring an operative EGM 704; that is, reconfiguring the EGM 704 after it had been configured and enabled 710 to allow play of a legacy game. Referring to FIG. 7, after initial configuration (FIG. 6), the download configuration server 702 may be controlled to change one or more game options. In the embodiment illustrated in FIG. 7, the download configuration host server 702 is controlled to change one option: the bet-per-line option from 5 to 10 credits 720 for the legacy game 708 in the EGM 704. As described above, the legacy game 708 is designed to set options only on installation, and not during operation. Consequently, to change options, the legacy game must be uninstalled and reinstalled, and the new options set at reinstallation time.

[0053] The download and configuration host 702 sends a bet-per-line change request 722 to the EGM 704. The OS 706 in the EGM 704 receives the request and detects that this option is meant for a legacy game 708 and, thus, is settable only at installation. In order to apply this option change request, the legacy game 708 is uninstalled and reinstalled. During this process, the legacy game 708 persistent data, e.g. NVRAM, is cleared, then the previous options are set again, along with the new value for the minimum bet option. Changes to configuration options are made during "Idle" conditions which meet jurisdictional and operational requirements. To verify this condition, the OS checks if the EGM 704 is in the "Game\_Idle" state and if so, proceeds doing the following steps:

- [0054] 1. Before terminating the current installation of the legacy game 708, the OS 706 in the EGM 704 stores 725 current values of game play history logs, and accounting information in archive storage to preserve the current critical information needed to resolve any player disputes. This is described in more detail in FIG. 9, FIG. 10, FIG. 11, FIG. 12, FIG. 13, FIG. 14, FIG. 15, FIG. 16 and the associated written description, below.
- [0055] 2. Terminates 724 the current legacy game 708.
- [0056] 3. Removes 726 all the persistent data storage generated and used by the legacy game 708.
- [0057] 4. Restarts 727 the same legacy game 708.

[0058] 5. After the game boots completely 728, the OS 706 in the EGM 704 apply 729 all the game options for the legacy game 708, including the new bet-per-line option value, as described below.

[0059] 6. The OS 706 in the EGM 704 reports 730 the proper option change status and event to the download and configuration host 702

[0060] 7. Enables 731, 732 the legacy game 708 after the post configuration idle period set by jurisdiction or operational requirements.

[0061] Entries in the game descriptor file (FIG. 3, 308) allow configuration of the options of the legacy game, as described below. More specifically, an entry in the game descriptor file 308 related to the minimum bet option allows the download and configuration host server 702 to specify this option and the OS 706 to change this option in the legacy game 708. The minimum bet option is an integer type option with a maximum range defined in credits. Changing the value of this option in the EGM 704 may require that certain "Number of Lines" or "Bet per Lines" buttons on the button deck panel be disabled and force the player to bet a different minimum wager per game cycle.

[0062] For example, referring to FIG. 8a, if the initial configuration sets the minimum bet to \$0.01, all the number-of-lines and bet-per-line buttons are enabled. Referring now to FIG. 8b, if an operator at the download and configuration host changes the value of the minimum bet option from \$0.01 to \$0.05, the OS 706 disables the four lowest bet-per-line buttons, based on an algorithm in the OS 706, so that the bet-per-line buttons displayed and enabled are the only choices players can select. In the algorithm in the OS 706, the game display defaults to maximum number of lines in this case. As illustrated in FIG. 8b, the 1-4 line buttons 802 are disabled and blank icons 804 are displayed instead.

[0063] During the course of game play (i.e. after the legacy game 708 starts and before it ends) the icon buttons 804 may be enabled and displayed e.g. based on a need for a game feature or bonus round. For example, if the legacy game 708 has a feature where the player needs to make a selection of 5 cards, then the number of lines buttons 802 (e.g. the four number of lines buttons 802) become enabled again for the player to make a selection. After the end of the game feature or bonus round state, the number of lines buttons 802 return to the disabled state 804.

[0064] As described above, before installation of a downloaded game or a game configuration change, the game play history for the current game is captured to one or more video files. One frame of the file history is stored in a portion of the video file corresponding to a video frame. In one embodiment, the history files are stored as delta compressed .f32 format video in the /critical\_data/gamehistory partition in the hard disk drive (not shown) on the EGM. However, other video file formats may also be used to store the game history. One skilled in the art understands how to select a video file format and to store the game history frames as corresponding video frames in the format of the selected video file.

[0065] These files are named as illustrated in FIG. 9. The display ID is 0 for a primary display, and 1 for a secondary display, and so forth; i.e. further digits may be assigned to represent other displays. The data/time stamp represents the date and time of creation of the file. The file extension, ".f32" indicates that the file is an .f32 movie so appropriate programs may be invoked to properly read and process it.

[0066] The process of capturing the game history screens is similar to that of setting game options of a legacy game by the OS. That is, touch coordinates for the screen buttons and/or other widgets operative to display screens containing game history are stored in the game descriptor file (FIG. 3, 308). These coordinates are used by the OS to simulate navigating to the game history screens displayed by the legacy game. When a history item is displayed, frame-by-frame screen captures are performed and written to the named .f32 file. The "History" section of the Game Descriptor File describes the information necessary for performing the history navigation and recoding process.

[0067] In general, access to the display screens available for saving the game play history and accounting history of the EGM begin with screen 1000, illustrated in FIG. 10. On screen 1000, activating the "Archives" button 1006 displays a different screen, e.g. screen 1100 as illustrated in FIG. 11. Screen 1100 is typical of those screens containing history information. On screen 1100, a base game screen 1101 is shown in the foreground while a top screen 1003 is shown layered in the background. The buttons required to navigate and view the archived history are displayed on screen 1100 as follows. A Next button 1102 is activated to display to the next game history record. The Next button 1102 is grayed out, and inactivated, if no more game history records exist past the current one. A Previous button 1104 is activated to display the previous game history record. The Previous button 1104 is grayed out, and inactivated, if no more game history records exist prior to the current one. A Swap button 1106 is activated to swap between displaying the base game screen 1101 in the foreground and displaying the top screen 1103 in the foreground, if both are available. The Swap button 1106 is grayed out if no top screen history was captured for this game. A Features button 1108 is activated to show the current game's feature history if it exists. The Features button 1108 is grayed out if no features were played for this game history record. An Exit button 1110 is activated to close the Archived Game History screen 1100 and return to the screen 1000 of FIG. 10.

[0068] In automatic operation under control of the OS 706 (FIG. 7), the OS 706 simulates operation of the Archives button 1006 (FIG. 10), which controls the legacy game to display the archived game history screen 1100, illustrated in FIG. 11. The OS 706 captures data representing the image of the screen, then generates data representing a video frame containing the captured image, and appends the generated video frame to the video file designated as illustrated in FIG. 9. If a top screen 1103 exists, the OS 706 simulates activation of the swap button 1106, displaying the top screen in the foreground. Data representing the image of the displayed top frame is captured and appended to the video file in the same manner. The OS 706 then simulates activation of the Next button 1102, causing the next archived screen image, or images, to be displayed. These screen images are appended to the video file as just described. This continues until there are no further archived game play history screens to display, signified by the Next button 1102 being made inactive.

[0069] In order to save the features history of the legacy game, the OS 706 (of FIG. 7) then simulates activating the features button 1108 (of FIG. 11). This controls the EGM to display a features history screen 1200, illustrated in FIG. 12, including a main screen 1204 in the foreground and another screen 1206 layered behind the main screen 1204. The OS 706 captures the displayed screen 1204 and appends it to the video file. In some games the top screen 1206 is used to

display additional game history information. In such cases, the OS 706 displays screen 1206 by simulating activation of the Swap button 1214. For example, FIG. 13 illustrates a display of the top screen 1306 of feature game history. The OS 706 captures screen 1306 and appends it to the video file. The OS 706 then advances to the next screen in the history by simulating activation of the Next button 1208. Subsequent feature history screens are captured and appended to the video file as just described. This continues until there are no more feature history screens. At this point the OS 706 simulates activation of the Exit button 1110, closing the Feature History screen 1100 and reopening the screen of FIG. 10.

[0070] The OS 706 then simulates activation of the current history button 1004 (of FIG. 10) which displays a current history screen similar to the archived history screen 1100 of FIG. 11 displaying the current game history. The same procedure of repeated screen capturing, appending to the video file, and swapping is performed to append frames representing the current game play history, and game feature history, to the video file. When there are no further current game play history screens to display, the Exit button is activated by the OS 706 closing the Current Game History screen and reopening the screen of FIG. 10.

[0071] After storing the game history information, and before installation of a downloaded legacy game, OS information is also captured to archive files, which in the illustrated embodiment are also video files. The following OS information is archived:

[0072] Accounting data (i.e. machine, bill, game, protocol, security, and error Accounting);

[0073] History data (i.e. event, transaction, card, and progressive history); and

[0074] Version information.

[0075] Screens representing OS information are captured by the same technique described above: the OS 706 (FIG. 7) simulates activation of appropriate on-screen buttons and/or widgets, captures screen images and appends those screen images to a portion of a video file representing a frame. Referring again to FIG. 10, current OS information is accessed the OS 706 simulating activation of the "Accounting" button 1008 for the accounting data, and "History" button 1010 for the history data.

[0076] When the OS 706 (FIG. 7) simulates activation of the Accounting Button 1008 (FIG. 10), a screen 1400, illustrated in FIG. 14, is displayed. Archived OS information is then accessed by the OS 706 simulating activation of the "Archives" button 1402. In response to activating the "Archives" button 1402, a screen 1500, illustrated in FIG. 15, is displayed including a list box 1502. One of the items from the list box may be selected to view. For example, in order to display the accounting information, the OS 706 simulates activating the Security Accounting entry 1504. In response, a screen, as illustrated in FIG. 16, is displayed. This screen displays a portion of the accounting data. The OS 706 captures image data representing this screen and appends it to the video file, as described above. The OS 706 (FIG. 7) simulates activation of a Next button 1602. In response, the EGM is controlled to display the next screen of accounting information. This screen is captured and appended to the video format file. This is repeated until the next button 1602 is grayed out, indicating that no more accounting data exists past the current screen.

[0077] All information desired to be archived before installation of the legacy game is stored in this manner: simulating

activation of appropriate on-screen buttons; capturing data representing desired screens; and appending the captured screens to a video format file.

[0078] Archived .f32 video files, containing game and OS data, are protected from unauthorized alteration. More specifically, a hash (e.g. SHA1) is calculated on the data in the .f32 file and is added to the game history log file(s) stored in /critical-data/Logs/partition of the hard disk drive (not shown) of the EGM. The game history log file(s) with the SHA1 hash is authenticated after power cycles or machine reboots. If any data changes in the f32 files or game history log itself, the EGM will fault and the RAM must be cleared to proceed.

[0079] FIG. 17 and FIG. 18, together, are a listing of an exemplary game descriptor file (308 of FIG. 3). Data in this file is used by the OS 706 (FIG. 7) to simulate activation of appropriate screen buttons, list entries, and/or other widgets. In general, when an operator activates an on-screen button, the operating system generates data representing the button activation. This data is stored in a construct sometimes called an event. The data in the event includes data representing the location touched, which button was activated, and so forth. The event data is supplied to different software routines which react to the received event by performing some processing. For example, pressing the Game Play History button 1002 (FIG. 10) generates an event which causes the legacy game software to display the game play history screen 1100 (FIG. 11) to be displayed. The OS 706 simulates activation of the Game Play History button 1002 by generating an event which contains the same data as an event which would have been generated by an operator activating the button. This event is then distributed to the rest of the system in the same manner as a normal, operator generated, event would be. The system then responds as if the Game Play History button 1002 had been activated by an operator.

[0080] The game descriptor file of FIG. 17 and FIG. 18, contains the information required by the OS 706 (FIG. 7) to generate the events which simulate activation of the required buttons, list entries, and/or other widgets. Referring specifically to FIG. 17, as may be seen, after identifying the descriptor type 1702 and part number 1704, the game descriptor file lists a series of features of the legacy game to be accessed by the OS. For example, "history" 1706 and "options" 1708. Within the history record 1706 are items 1710, 1712, 1714 and 1716, indicating button types and locations. More specifically, item 1710 describes a button having the type "enter", text "feat1", and screen location x=0.777187 and y=0.032564. When the OS 706 desires to activate this button, it assembles a button-press event using this data from the GDF, and passes that event to the legacy game, simulating activation of this button. Other items in the game descriptor file FIG. 17 and FIG. 18 provide required data to the OS 706 for it to simulate activation of the other buttons, list entries, etc. This data is used by the OS 706 to navigate through the legacy game menu screens as described above.

What is claimed is:

1. In a system including a download configuration host server coupled to an electronic gaming machine comprising an operating system, a method for configuring a legacy game application on the electronic gaming machine, comprising:
  - generating a downloadable package comprising a game image of the legacy game and a game descriptor file;

communicating the downloadable package from the download configuration host server to the electronic gaming machine;  
 installing the legacy game image on the electronic gaming machine;  
 configuring the installed legacy game by the operating system simulating operator configuration of the installed legacy game in response to data in the game descriptor file; and  
 enabling the installed legacy game for game play.

2. The method of claim 1, wherein the step of generating a downloadable package comprises including a game image of the legacy game approved by regulators and cryptographically signed to detect modifications.

3. The method of claim 2 wherein the step of generating a downloadable package comprises cryptographically signing the downloadable package to detect modifications.

4. The method of claim 1, wherein the legacy game displays configuration screens for operator control, and the game descriptor file comprises data representing the configuration screens of the legacy game; and wherein the step of configuring the installed legacy game on the electronic gaming machine comprises:

the operating system reading the configuration screen representative data from the game descriptor file; and  
 the operating system simulating operator control of the configuration screens of the legacy game in response to the configuration screen representative data.

5. The method of claim 4 wherein the configuration screen representative data comprises data representing screen locations of buttons and/or other widgets displayed on the configuration screens of the legacy game, and the step of configuring the installed legacy game image comprises:

the operating system reading the screen location representative data from the game descriptor file; and  
 the operating system simulating activation of buttons and/or other widgets in response to the screen location representative data appropriate for configuring the legacy game.

6. In a system including a download configuration host server coupled to an electronic gaming machine comprising an operating system, a method for dynamically and remotely reconfiguring a legacy game application, comprising:

generating a downloadable package comprising a game image of the legacy game image and a game descriptor file;

communicating the downloadable package from the download configuration host server to the electronic gaming machine;

installing the legacy game image on the electronic gaming machine;

configuring the installed legacy game by the operating system simulating operator activation of the legacy game in response to data in the game descriptor file;

enabling the installed legacy game for game play;

communicating a configuration change from the download configuration host server to the electronic gaming machine;

disabling the installed legacy game for game play;

reinstalling the legacy game image on the electronic gaming machine;

configuring the reinstalled installed legacy game by the operating system simulating operator activation of the legacy game in response to data in the game description file and the configuration change; and

enabling the legacy game for game play.

7. The method of claim 6, wherein the legacy game displays configuration screens for operator control, and the game descriptor file comprises data representing the configuration screens of the legacy game; and wherein the step of configuring the reinstalled legacy game on the electronic gaming machine comprises:

the operating system reading the configuration screen representative data from the game descriptor file; and

the operating system simulating operator control of the configuration screens of the legacy game in response to the configuration screen representative data.

8. The method of claim 7 wherein the configuration screen representative data comprises data representing screen locations of buttons and/or other widgets displayed on the configuration screens of the legacy game, and the step of configuring the reinstalled legacy game image comprises:

the operating system reading the screen location representative data from the game descriptor file; and

the operating system simulating activation of buttons and/or other widgets in response to the screen location representative data appropriate for configuring the legacy game.

\* \* \* \* \*