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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
22 December 2005 (22.12.2005)

PCT

(10) International Publication Number
WO 2005/121968 A2

(51) International Patent Classification[7]: G06F 12/00

(21) International Application Number:
PCT/US2005/020133

(22) International Filing Date: 7 June 2005 (07.06.2005)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/577,706 7 June 2004 (07.06.2004) US

(71) Applicant *(for all designated States except US)*: **WMS GAMING INC.** [US/US]; 800 South Northpoint Boulevard, Waukegan, IL 60089 (US).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **GAGNER, Mark, B.** [US/US]; 30W515 Diversey Parkway, West Chicago, IL 60185 (US). **KLOES, Robertus, A.** [US/US]; 4655 N. Winthrop Avenue, Apt. 1, Chicago, IL 60640 (US). **SYLLA, Craig, J.** [US/US]; 247 W. Treehouse Lane, Round Lake, IL 60073 (US).

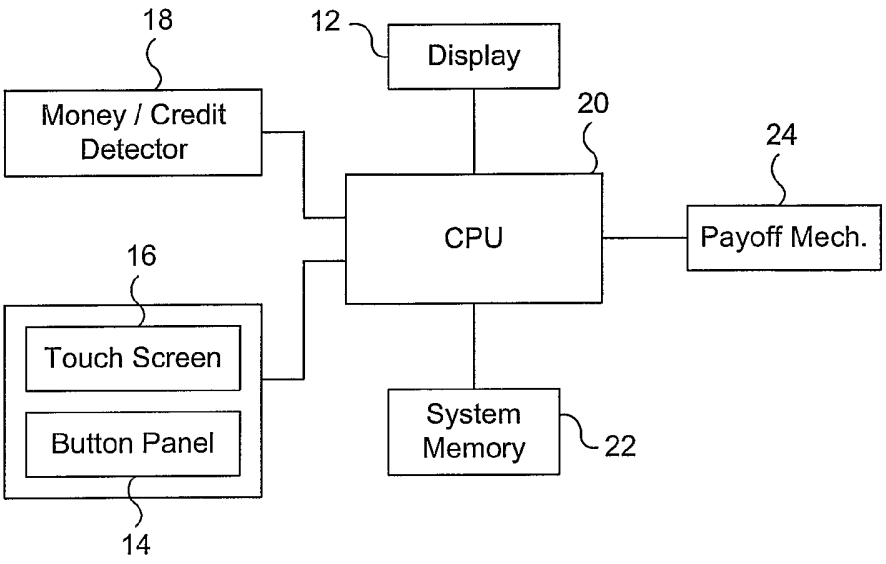
(74) Agents: **STEFFEY, Charles, E.** et al.; P.O. Box 2938, Minneapolis, MN 55402 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, MC, NL, PL, PT, RO,

[Continued on next page]

(54) Title: GAMING DEVICE WITH RESOURCE SWAPPING



(57) Abstract: Embodiments of the invention include a method for conserving memory storage on a gaming machine. The method includes creating a resource file and caching the resource file in RAM on a gaming machine; determining whether the resource file is accessed more than a preselected number of times within a preselected time interval; caching the resource file in a SWAP store as a SWAP file if the resource file has not been accessed the preselected number of times within the preselected time interval; monitoring the number of times the SWAP file is accessed within a preselected period of time; and caching the SWAP file as a resource file in RAM of the gaming machine if the SWAP file is accessed a number of preselected times or more.

WO 2005/121968 A2

SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

GAMING DEVICE WITH RESOURCE SWAPPING

Cross-reference to Related Application

This application is related to United States Provisional Patent Application serial no. 60/577,706, entitled "Gaming Device with Resource Swapping" (Attorney Docket 1842.115PRV), filed on June 7, 2004.

TECHNICAL FIELD

EMBODIMENTS OF THE INVENTION DESCRIBED HEREIN RELATE TO A GAMING DEVICE WITH RESOURCE SWAPPING.

Copyright

A portion of the disclosure of this patent document contains material that is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure, as it appears in the Patent and Trademark Office patent files or records, but otherwise reserves all copyright rights whatsoever. The following notice applies to any software and data as described below and in the drawings that form a part of this document: Copyright 2005, WMS Gaming, Inc. All Rights Reserved.

SUMMARY

Embodiments of the invention include a method for conserving memory storage on a gaming machine. The method includes creating a resource file and caching the resource file in RAM on a gaming machine; determining whether the resource file is accessed more than a preselected number of times within a preselected time interval; caching the resource file in a SWAP store as a SWAP file if the resource file has not been accessed the preselected number of times within the preselected time interval; monitoring the number of times the SWAP file is accessed within a preselected period of time; and caching the SWAP file as a resource file in RAM of the gaming machine if the SWAP file is accessed a number of preselected times or more.

Another embodiment includes a method for increasing the number of images, animations, and sounds on a gaming machine. The method includes creating one or more resource files associated with one or more games and

caching the one or more resource files in RAM on the gaming machine; determining whether the one or more resource files is accessed more than a preselected number of times within a preselected time interval; caching one or more of the resource filea in a SWAP store as one or more SWAP files if the one or more resource files has not been accessed the preselected number of times within the preselected time interval; monitoring the number of times the one or more "SWAP files are accessed within a preselected period of time; and caching the one or more SWAP files as one or more resource files in RAM of the gaming machine if the SWAP file is accessed a number of preselected times or more.

Another embodiment includes a gaming machine. The gaming machine includes RAM; a plurality of resource files stored on RAM, wherein the resource files have been accessed a preselected number of times or more over a preselected time interval; a SWAP store; and a plurality of SWAP files cached on the SWAP store.

One other embodiment includes a Computer Readable Medium for conserving memory storage on a gaming machine, the computer readable medium includes creating a resource file and caching the resource file in RAM on a gaming machine; determining whether the resource file is accessed more than a preselected number of times within a preselected time interval; caching the resource file in a SWAP store as a SWAP file if the resource file has not been accessed the preselected number of times within the preselected time interval; monitoring the number of times the SWAP file is accessed within a preselected period of time; and caching the SWAP file as a resource file in RAM of the gaming machine if the SWAP file is accessed a number of preselected times or more.

DESCRIPTION OF THE FIGURES

FIG. 1 illustrates a perspective view of a prior art gaming machine embodiment.

FIG. 2 illustrates a prior art block diagram of a control system embodiment suitable for operating the prior art gaming machine shown in FIG. 1.

FIG. 3 illustrates a block diagram of a *Cached Resource Swap* mechanism that swaps unused Cached Resource Objects out of RAM and onto a secondary high-speed storage device.

DESCRIPTION

In the following detailed description of exemplary embodiments of the invention, reference is made to the accompanying drawings which form a part hereof, and in which is shown by way of illustration specific exemplary embodiments in which the invention may be practiced. These embodiments are described in sufficient detail to enable those skilled in the art to practice the invention, and it is to be understood that other embodiments may be utilized and that logical, mechanical, electrical and other changes may be made without departing from the scope of the present invention.

Some portions of the detailed descriptions which follow are presented in terms of algorithms and symbolic representations of operations on data bits within a computer memory. These algorithmic descriptions and representations are the ways used by those skilled in the data processing arts to most effectively convey the substance of their work to others skilled in the art. An algorithm is here, and generally, conceived to be a self-consistent sequence of steps leading to a desired result. The steps are those requiring physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared, and otherwise manipulated. It has proven convenient at times, principally for reasons of common usage, to refer to these signals as bits, values, elements, symbols, characters, terms, numbers, or the like. It should be borne in mind, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities. Unless specifically stated otherwise as apparent from the following discussions, terms such as "processing" or "computing" or "calculating" or "determining" or "displaying" or the like, refer to the action and processes of a computer system, or similar computing device, that manipulates and transforms data represented as physical (e.g., electronic) quantities within the computer system's registers and memories into other data similarly

represented as physical quantities within the computer system memories or registers or other such information storage, transmission or display devices.

In the Figures, the same reference number is used throughout to refer to an identical component which appears in multiple Figures. Signals and connections may be referred to by the same reference number or label, and the actual meaning will be clear from its use in the context of the description.

The following detailed description is, therefore, not to be taken in a limiting sense, and the scope of the present invention is defined only by the appended claims.

Prior art FIG. 1 is a perspective view of a gaming machine 10 for conducting a wagering game. The gaming machine 10 includes a display 12 that may, for example, be a mechanical reel slot display or a video display. If implemented in video, the display 12 may include a cathode ray tube (CRT), liquid crystal display (LCD), plasma, or generally any other type of video display known in the art. In the illustrated embodiment, the gaming machine 10 is an "upright" version in which the display 12 is oriented vertically relative to the player. It will be appreciated, however, that any of several other models of gaming machines are within the scope of the present invention including, for example, a "slant-top" version in which the display 12 is slanted at about a thirty-degree angle toward the player. The gaming machine may include a secondary display in top box 13, depending upon the design of the wagering game.

The gaming machine 10 includes a button panel 14 and an optional touch screen 16 for enabling a player to conduct the wagering game. The button panel 14 and the touch screen 16 may have some corresponding buttons for performing the same functions and other buttons for performing different functions. By way of example, a "$" button may be pressed to insert money to increase a player's credits. A "Change Denom" button may be touched to change denomination. A "Collect" button may be touched to collect winnings. A "Pay Table" button may be touched to see the game's pay table. A "Select Lines" button may be touched to change the number of pay lines bet. A "Bet Per Line" button may be touched to change the amount of credits bet on each line. The number of lines to play is

generally selected prior to selecting the amount to bet per line. A "Spin Reels" button may be touched to begin the game. A "Stop Reels" button may be touched to stop the reels while they are spinning. If the player does not touch the "Stop Reels" button, the reels will stop on their own. A "Max Bet Spin" button may be touched to play the maximum credits instantly.

Prior art FIG. 2 is a block diagram of a control system suitable for operating the gaming machine. Money/credit detector 18 signals a central processing unit (CPU) 20 when a player has inserted money or played a number of credits. The money may be provided by coins, bills, tickets, coupons, cards, etc. Using the button panel 14 or touch screen 16, the player may select any game variables (e.g., number of pay lines in a reel slot game if more than one can be selected) and place his/her wager to play the wagering game. In a play of the game, the CPU 20 generates at least one random event using a random number generator (RNG) and provides an award to the player for a winning outcome of the random event. Alternatively, the random event may be generated by a remote computer using an RNG or pooling schema and then transmitted to the gaming machine. The CPU 20 operates the display 12 to represent the random event(s) and outcome(s) in a visual form that can be understood by the player. In addition to the CPU 20, the control system may include one or more additional slave control units for operating one or more peripherals. The terms "processor" and CPU as used herein are each intended to collectively cover the CPU 20 and any and all such control units.

A system memory 22 stores control software, operational instructions and data associated with the gaming machine. In one embodiment, the system memory 22 comprises a separate read-only memory (ROM) and battery-backed random-access memory (RAM). However, it will be appreciated that the system memory 22 may be implemented on any of several alternative types of memory structures or may be implemented on a single memory structure. A payoff mechanism 24 is operable in response to instructions from the CPU 20 to award a payoff to the player. The payoff may, for example, be in the form of a number of credits. The number of credits are determined by one or more math tables stored in the system memory 22.

A wagering game is described below wherein the basic game is a multi-line, multi-coin video reel slot game. It should, however, be understood that the basic game may be any type of wagering game, including but not limited to video or mechanical reel slots (one or more pay lines), poker, keno, bingo, blackjack, or roulette.

Referring back to prior art FIG. 1, to play a basic game in the form of reel slots, a player may select a number of pay lines using the "Select Lines" button, place a wager on each line using the "Bet Per Line" button, and start the reels 26 in motion using the "Spin Reels" button. Alternatively, the player may select all lines, play the maximum bet on each line, and start the reels 26 in motion using the "Max Bet Spin" button. In response to pressing the "Max Bet Spin" button, or a "Spin Reels" button for a wager less than the maximum, the CPU spins and randomly stops the plurality of symbol-bearing reels 26 to place symbols on the reels 26 in visual association with a number of pay lines 28. Other mechanisms, such as a handle, may be used to set the reels 26 in motion. The number of pay lines 28 may, for example, be nine, fifteen, or twenty, but more or less pay lines may be provided is desired. As noted above, the display 12 on which the reels are implemented may be mechanical or video. If the display 12 is video, the reels are simulated with moving graphics. If, however, the display 12 is mechanical, the reels are physical and rotatably driven by stepper motors.

The CPU may use a random number generator to select a game outcome (*e.g.*, "basic" game outcome) corresponding to a particular set of reel "stop positions." The CPU then causes each of the reels 26 to stop at the appropriate stop position. Symbols are displayed on the reels 26 to graphically illustrate the reel stop positions and indicate whether the stop positions of the reels 26 represent a winning game outcome. Winning basic game outcomes (*e.g.*, symbol combinations resulting in payment of coins or credits) are identifiable to the player by a pay table. The pay table may be affixed to the machine 10 and/or displayed by the display 12 in response to a command by the player (*e.g.*, by pressing the "Pay Table" button). A winning basic game outcome occurs when the symbols appearing on the reels 26 along an active pay line 28 correspond to

one of the winning combinations on the pay table. A winning combination could, for example, be three or more matching symbols along an active pay line 28, where the award is greater as the number of matching symbols along the active pay line 28 increases. If the displayed symbols stop in a winning combination, the game credits the player an amount corresponding to the award in the pay table for that combination multiplied by the number of wagered credits. The player may collect the amount of accumulated credits by pressing the "Collect" button.

Included among the plurality of basic game outcomes are one or more start-bonus outcomes for triggering play of bonus games. A start-bonus outcome may be defined in any number of ways. For example, a start-bonus outcome may occur when a special start-bonus symbol or a special combination of symbols appears on one or more of the reels 26. A start-bonus outcome may require the combination of symbols to appear along an active pay line 28, or may alternatively require that the combination of symbols appear anywhere on the display regardless of whether the symbols are along an active pay line 28. The appearance of a start-bonus outcome causes the CPU to shift operation from the basic slot game to the bonus game associated with that start-bonus outcome.

A modern gaming machine has a large number of Resource files, which store individual images, animations, sounds and so forth. Most jurisdictions require that these Resource files be stored on read-only media in a secure, verifiable format. This type of media is generally much slower than system RAM.

To speed rendering, most Resource files are verified, decompressed, converted to a native display format and cached in RAM. This is typically done at start-up. All Resources, even those that are not currently visible, are cached in system RAM. This approach improves performance during game play but limits the number and complexity of the Resources to what will fit into RAM after decompression.

On multi-game platforms the number and size of the Resources exceeds the capacity of system RAM and may therefore limit a manufacturer's product offerings.

Since most Resources are not visible at any given moment, a mechanism is needed for moving the decompressed, cached images from system RAM to a high-speed storage device when they are not needed.

Referring to FIG. 3, there is shown a *Cached Resource Swap* mechanism that swaps unused Cached Resource Objects out of RAM and onto a secondary high-speed storage device. This method conserves limited RAM space and allows more elaborate and complex games to be created.

In the example shown at 30 in FIG. 3, Flash memory 32 is shown as the secure *Resource Store*. However, any secure, verifiable media (such as ROMs, CD-ROMs or DVDs) might be used. Further, in this example a Hard Drive (HD) 34 is shown as the high-speed *Swap Store* but any type of high-speed media may be used in an actual embodiment.

When a Resource is to be instantiated a *Cached Resource Object* (CImage in this example) 36 is created. The specified Resource file 38 is read from the Resource Store, verified, decompressed, converted to the native display format and cached into RAM. The Cached Resource Object 36 maintains a reference to this *Cached Image*. Note that the Cached Image may be deleted without destroying the Cached Resource Object. In this case the associated reference will be NULL.

If swapping is enabled then a corresponding swap file 40 is created on the high-speed Swap Store 34 and the cached image is copied into the swap file.

A particular image may appear in more than one location on a particular display or set of displays. For example, a particular symbol such as a cherry may appear in multiple locations on one or more reels in a video slot game. Each of these potential appearances has a reference to the associated Cached Resource Object (CImage in this example). The Cached Resource Object keeps a *Reference Count.*

Each reference to a Cached Resource Object is independent and may be visible or invisible at any given moment. If the reference is visible then it instructs the corresponding Cached Resource Object to *Show*. Conversely, if the reference is not visible it instructs the Cached Resource Object to *Hide*. The

Cached Resource Object maintains a *Show Count*, which keeps track of the number of references that wish to show the image.

If the Show Count becomes zero and swapping is enabled then the Cached Resource Object will delete the Cached Image from memory, which frees valuable RAM space. Thus, only those images that are actually visible are retained in limited system RAM.

When the Show Count becomes non-zero then the cached image is restored from the swap file. Since the swap file has already been verified and decompressed and exists on high-speed storage, restoration is much faster than retrieving and decompressing the original resource image from the Resource Store.

Swapping shall be enabled / disabled on an individual object basis. In other words, individual objects may be declared "non-swappable". For example, critical items such as tilt displays or frequently used items such as fonts may be declared "non-swappable."

If any single reference is non-swappable then the object is non-swappable.

When the Cached Resource Object is destroyed it deletes the cached image from RAM and deletes the corresponding swap file from the Swap Store. Since the swap files are not validated (i.e. verified) they must be reconstructed each time the corresponding Resource file is loaded to satisfy jurisdictional requirements.

The above technique is not limited to graphical images but can also be applied for any type of Resource that may be stored in slow media. For example, the technique may also be applied to audio files or scripts.

Although the foregoing invention has been described in some detail for purposes of clarity of understanding, it will be apparent that certain changes and modifications may be practiced within the scope of the appended claims. For instance, while the gaming machines of this invention have been depicted as upright models having top box mounted on top of the main gaming machine cabinet, the use of gaming devices in accordance with this invention is not so

limited. For example, gaming machine may be provided without a top box or the gaming machine may be of a slant-top or a table top design.

IN THE CLAIMS

1. A method for conserving memory storage on a gaming machine, comprising:

Creating a resource file and caching the resource file in RAM on a gaming machine;

Determining whether the resource file is accessed more than a preselected number of times within a preselected time interval;

Caching the resource file in a SWAP store as a SWAP file if the resource file has not been accessed the preselected number of times within the preselected time interval;

Monitoring the number of times the SWAP file is accessed within a preselected period of time; and

Caching the SWAP file as a resource file in RAM of the gaming machine if the SWAP file is accessed a number of preselected times or more.

2. The method of claim 1 wherein the resource file comprises one or more of an image, animation or a sound.

3. The method of claim 1 wherein the resource file is stored in a resource store.

4. The method of claim 1 wherein the resource file is instantiated to create a cached resource object.

5. The method of claim 4 wherein the cached resource object maintains a reference to a cached image.

6. The method of claim 1, further comprising verifying the SWAP file.

7. The method of claim 1, further comprising decompressing the SWAP file.

8. The method of claim 1, further comprising converting the SWAP file to native display format.

9. The method of claim 5, further comprising copying the cached image into the SWAP file.

10. The method of claim 9, further comprising restoring the cached image from the SWAP file.

11. The method of claim 4, wherein the cached resource object keeps a reference count.

12. The method of claim 4, wherein the cached resource object tracks a number of references that wish to show an image associated with the cached resource object.

13. The method of claim 12, wherein when the number of references is non-zero, a cached image is restored from the SWAP file.

14. The method of claim 1, wherein the Resource file is stored on read-only media.

15. The method of claim 5 wherein the cached image appears in more than one location on a display or set of displays.

16. The method of claim 4 wherein when the cached resource object is destroyed, the cached resource object deletes the cached image and corresponding SWAP file.

17. A method for increasing the number of images, animations, and sounds on a gaming machine, the method comprising:

Creating one or more resource files associated with one or more games and caching the one or more resource files in RAM on the gaming machine;

Determining whether the one or more resource files is accessed more than a preselected number of times within a preselected time interval;

Caching one or more of the resource filea in a SWAP store as one or more SWAP files if the one or more resource files has not been accessed the preselected number of times within the preselected time interval;

Monitoring the number of times the one or more "SWAP files are accessed within a preselected period of time; and

Caching the one or more SWAP files as one or more resource files in RAM of the gaming machine if the SWAP file is accessed a number of preselected times or more.

18. The method of claim 17 wherein the resource file comprises one or more of an image, animation or a sound.

19. The method of claim 17 wherein the resource file is stored in a resource store.

20. The method of claim 17 wherein the resource file is instantiated to create a cached resource object.

21. The method of claim 20 wherein the cached resource object maintains a reference to a cached image.

22. The method of claim 17, further comprising verifying the SWAP file.

23. The method of claim 17, further comprising decompressing the SWAP file.

24. The method of claim 17, further comprising converting the SWAP file to native display format.

25. The method of claim 21, further comprising copying the cached image into the SWAP file.

26. The method of claim 25, further comprising restoring the cached image from the SWAP file.

27. The method of claim 20, wherein the cached resource object keeps a reference count.

28. The method of claim 20, wherein the cached resource object tracks a number of references that wish to show an image associated with the cached resource object.

29. The method of claim 28, wherein when the number of references is non-zero, a cached image is restored from the SWAP file.

30. The method of claim 17, wherein the Resource file is stored on read-only media.

31. The method of claim 21 wherein the cached image appears in more than one location on a display or set of displays.

32. The method of claim 21 wherein when the cached resource object is destroyed, the cached resource object deletes the cached image and corresponding SWAP file.

33. A gaming machine comprising:

RAM;

A plurality of resource files stored on RAM, wherein the resource files have been accessed a preselected number of times or more over a preselected time interval;

A SWAP store; and

A plurality of SWAP files cached on the SWAP store.

34. The gaming machine of claim 33 wherein the resource file comprises one or more of an image, animation or a sound.

35. The gaming machine of claim 33 wherein the resource file is stored in a resource store.

36. The gaming machine of claim 33 wherein the resource file is instantiated to create a cached resource object.

37. The gaming machine of claim 36 wherein the cached resource object maintains a reference to a cached image.

38. The gaming machine of claim 33, further comprising a secondary storage device for storing files that have been accessed less than the preselected number of times.

39. A Computer Readable Medium for conserving memory storage on a gaming machine, the computer readable medium comprising:

Creating a resource file and caching the resource file in RAM on a gaming machine;

Determining whether the resource file is accessed more than a preselected number of times within a preselected time interval;

Caching the resource file in a SWAP store as a SWAP file if the resource file has not been accessed the preselected number of times within the preselected time interval;

Monitoring the number of times the SWAP file is accessed within a preselected period of time; and

Caching the SWAP file as a resource file in RAM of the gaming machine if the SWAP file is accessed a number of preselected times or more.

40. The Computer Readable Medium of claim 39 wherein the resource file comprises one or more of an image, animation or a sound.

41. The Computer Readable Medium of claim 39 wherein the resource file is stored in a resource store.

42. The Computer Readable Medium of claim 39 wherein the resource file is instantiated to create a cached resource object.

43. The Computer Readable Medium of claim 42 wherein the cached resource object maintains a reference to a cached image.

44. The Computer Readable Medium of claim 39, further comprising verifying the SWAP file.

45. The Computer Readable Medium of claim39, further comprising decompressing the SWAP file.

46. The Computer Readable Medium of claim 39, further comprising converting the SWAP file to native display format.

47. The Computer Readable Medium of claim 42, further comprising copying the cached image into the SWAP file.

48. The Computer Readable Medium of claim 47, further comprising restoring the cached image from the SWAP file.

49. The Computer Readable Medium of claim 42, wherein the cached resource object keeps a reference count.

50. The Computer Readable Medium of claim 42, wherein the cached resource object tracks a number of references that wish to show an image associated with the cached resource object.

51. The Computer Readable Medium of claim 50, wherein when the number of references is non-zero, a cached image is restored from the SWAP file.

52. The Computer Readable Medium of claim 39, wherein the Resource file is stored on read-only media.

53. The Computer Readable Medium of claim 43 wherein the cached image appears in more than one location on a display or set of displays.

54. The Computer Readable Medium of claim 43 wherein when the cached resource object is destroyed, the cached resource object deletes the cached image and corresponding SWAP file.

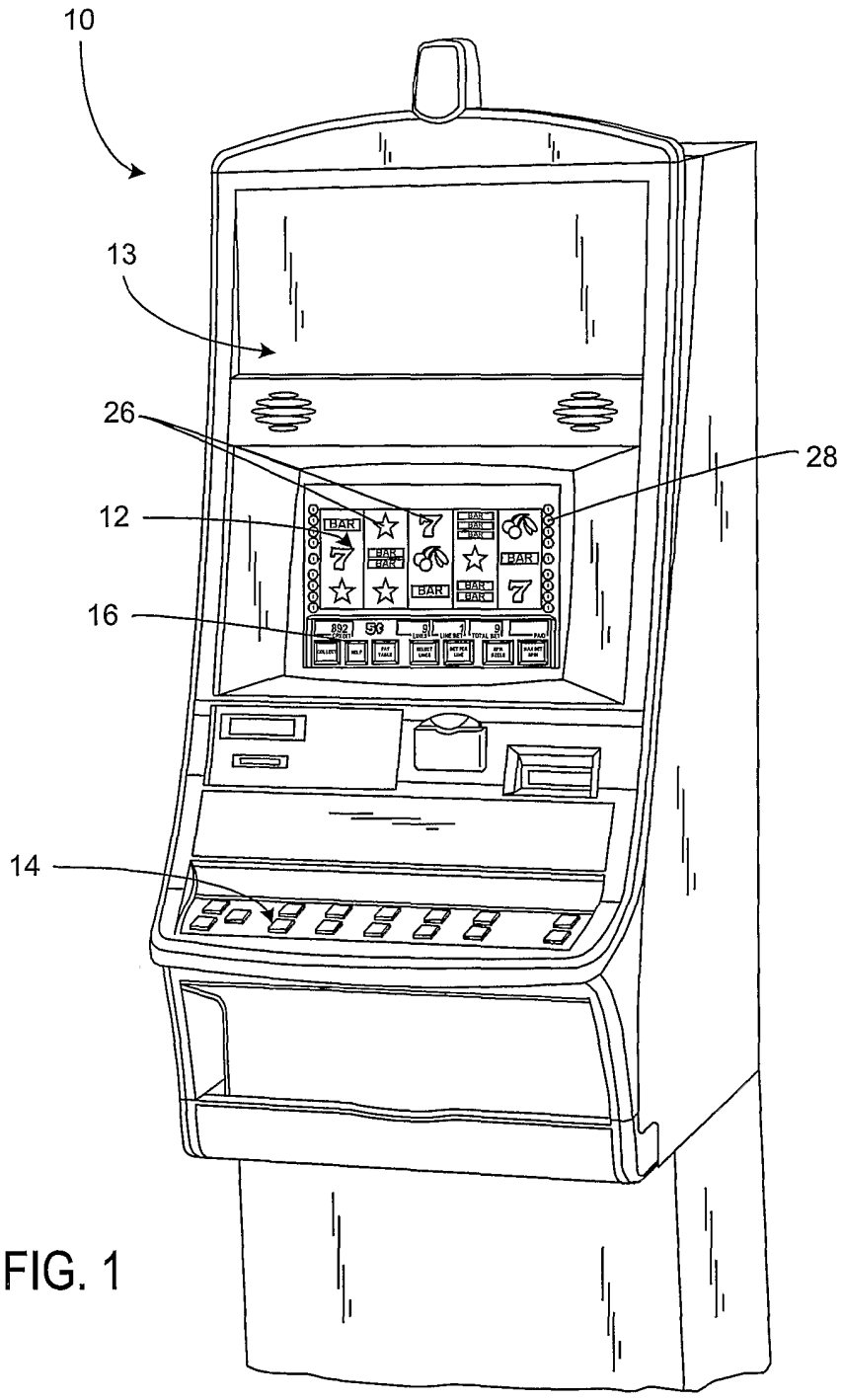


FIG. 1

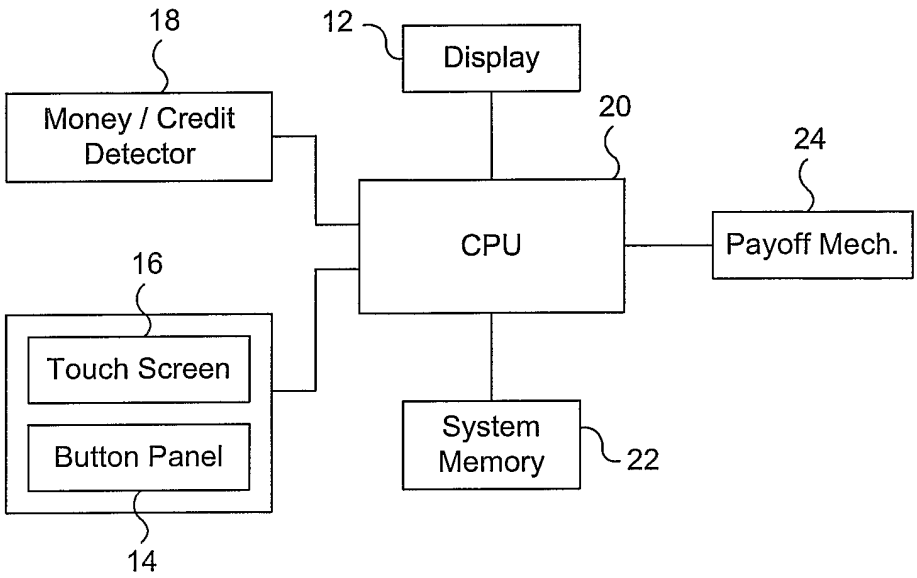

18
12
Display
Money / Credit Detector
20
24
CPU
Payoff Mech.
16
Touch Screen
Button Panel
System Memory
22
14

FIG. 2

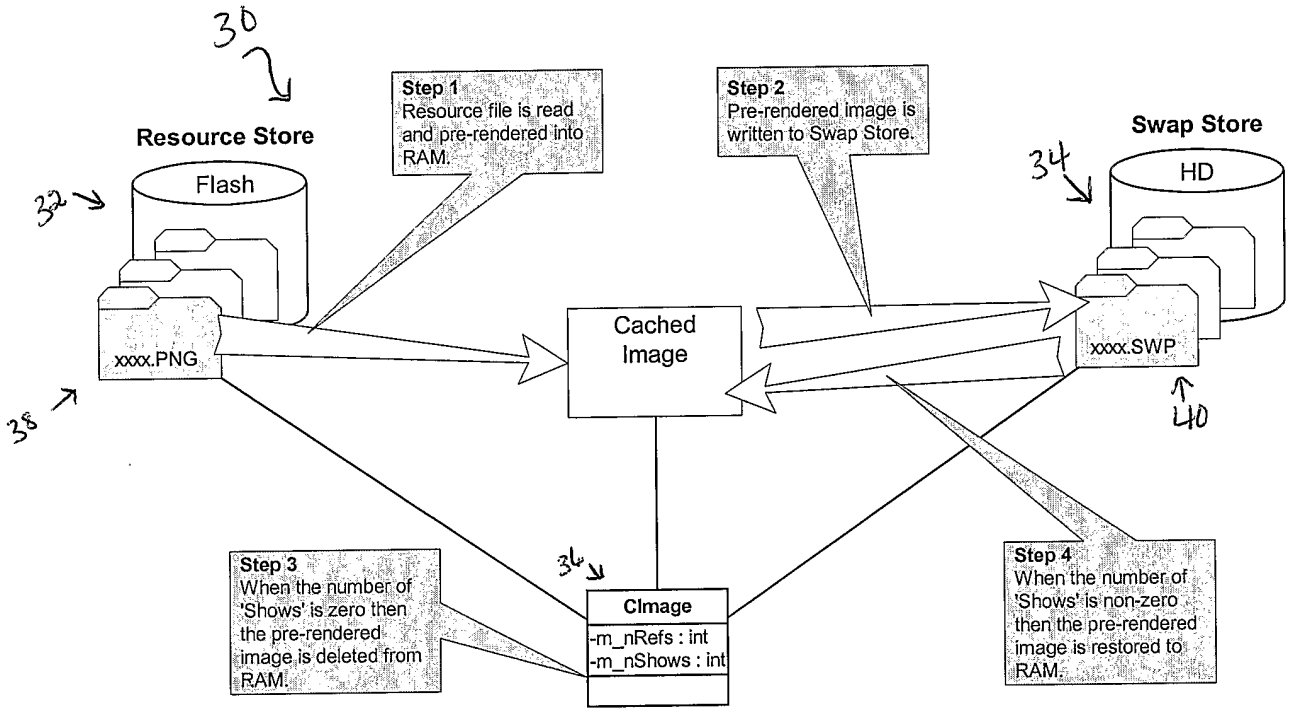

30
Resource Store
Flash
32
xxxx.PNG
38
Step 1
Resource file is read and pre-rendered into RAM.
Step 2
Pre-rendered image is written to Swap Store.
Swap Store
34
HD
xxxx.SWP
40
Cached Image
Step 3
When the number of 'Shows' is zero then the pre-rendered image is deleted from RAM.
36
CImage
-m_nRefs : int
-m_nShows : int
Step 4
When the number of 'Shows' is non-zero then the pre-rendered image is restored to RAM.

FIG. 3