US 20040038740A1

(19) **United States**

(12) **Patent Application Publication** (10) **Pub. No.: US 2004/0038740 A1**

Muir (43) **Pub. Date: Feb. 26, 2004**

(54) **MULTI-PLATFORM GAMING ARCHITECTURE**

(76) Inventor: **Robert Linley Muir**, Artarmon (AU)

Correspondence Address:
KATTEN MUCHIN ZAVIS ROSENMAN
575 MADISON AVENUE
NEW YORK, NY 10022-2585 (US)

(21) Appl. No.: **10/648,178**

(22) Filed: **Aug. 26, 2003**

Related U.S. Application Data

(62) Division of application No. 09/238,535, filed on Jan. 27, 1999.

(30) **Foreign Application Priority Data**

Jan. 27, 1998 (AU) PP1499

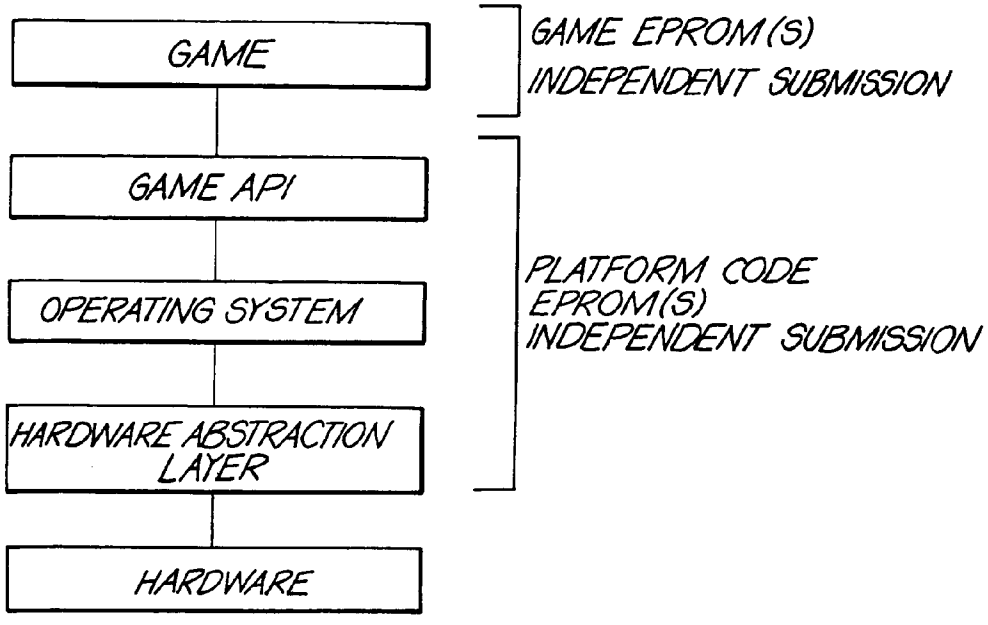
Publication Classification

(51) **Int. Cl.**[7] **A63F 9/24**

(52) **U.S. Cl.** **463/40**

(57) **ABSTRACT**

A gaming console architecture includes a game platform interface and a game program, the game program including a plurality of functional modules, such as "combinations" and "graphics/audio" which interact via the platform interface. The game program may include a user interface module and a combinations module and communication of combinations to be displayed, are conveyed from the combinations module to the user interface module via the platform interface. The architecture allows the creation of games which run on a variety of platforms of different architectures which may be stored on a generic "game server" which stores games for execution and distribution to various platforms including electronic gaming machines, internet consoles and televisions.



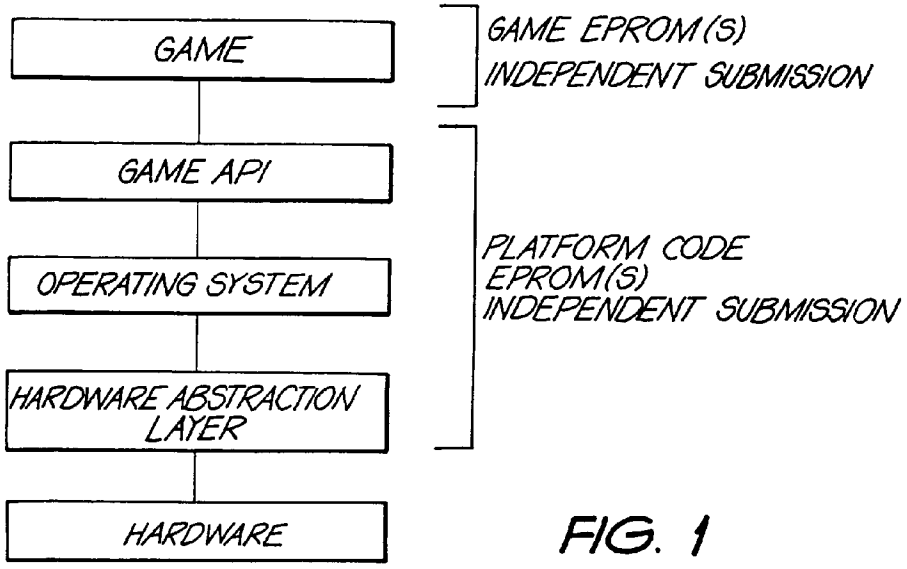

GAME
GAME API
OPERATING SYSTEM
HARDWARE ABSTRACTION LAYER
HARDWARE
GAME EPROM(S) INDEPENDENT SUBMISSION
PLATFORM CODE EPROM(S) INDEPENDENT SUBMISSION

FIG. 1

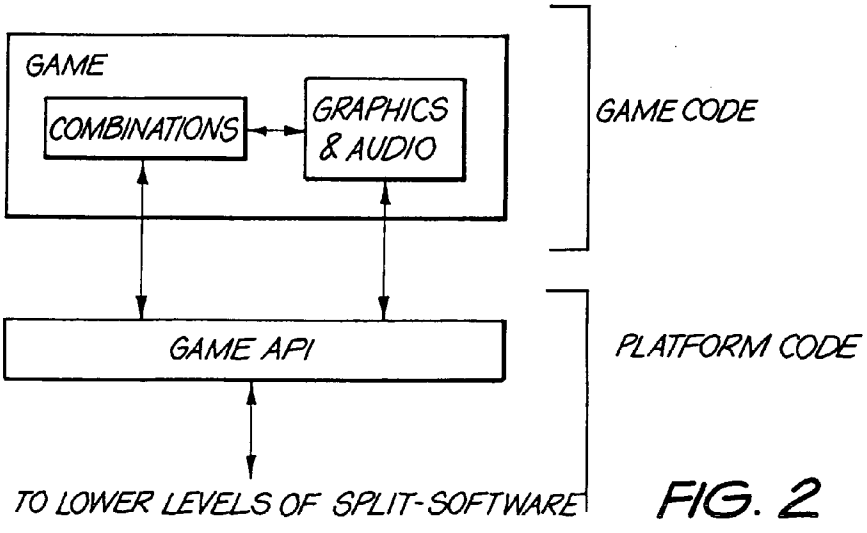

GAME
COMBINATIONS
GRAPHICS & AUDIO
GAME API
TO LOWER LEVELS OF SPLIT-SOFTWARE
GAME CODE
PLATFORM CODE

FIG. 2

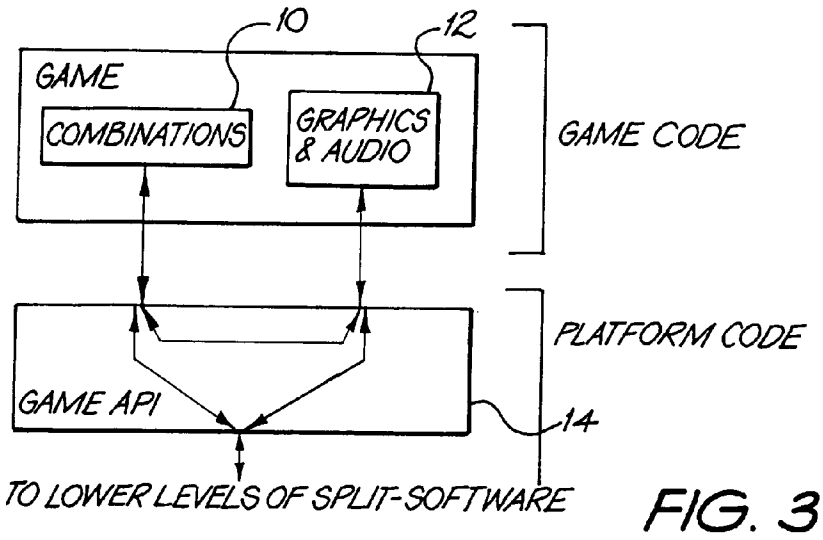

10
12
GAME
COMBINATIONS
GRAPHICS & AUDIO
GAME CODE
PLATFORM CODE
GAME API
14
TO LOWER LEVELS OF SPLIT-SOFTWARE

FIG. 3

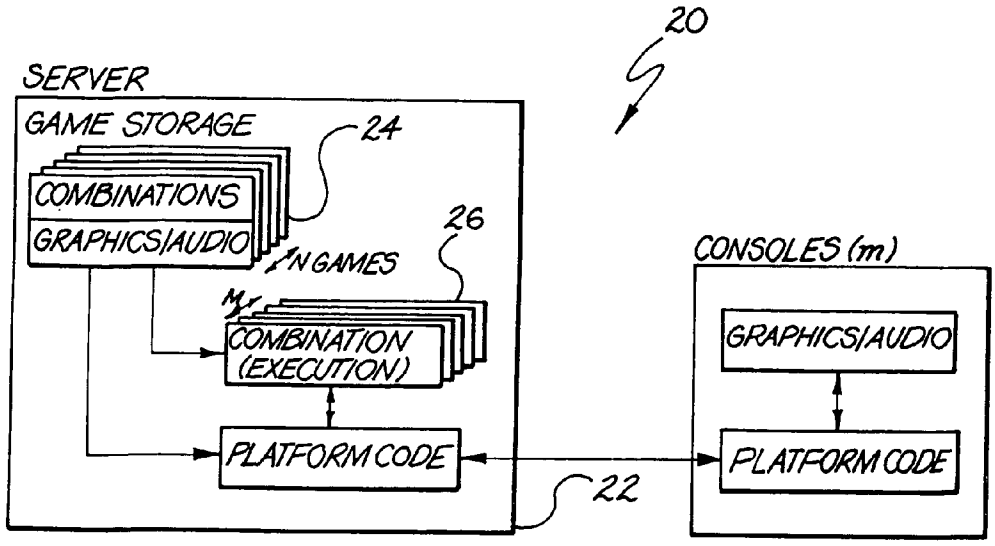

20
SERVER
GAME STORAGE
24
COMBINATIONS
GRAPHICS/AUDIO
N GAMES
26
M
COMBINATION (EXECUTION)
PLATFORM CODE
22
CONSOLES (m)
GRAPHICS/AUDIO
PLATFORM CODE

FIG. 4

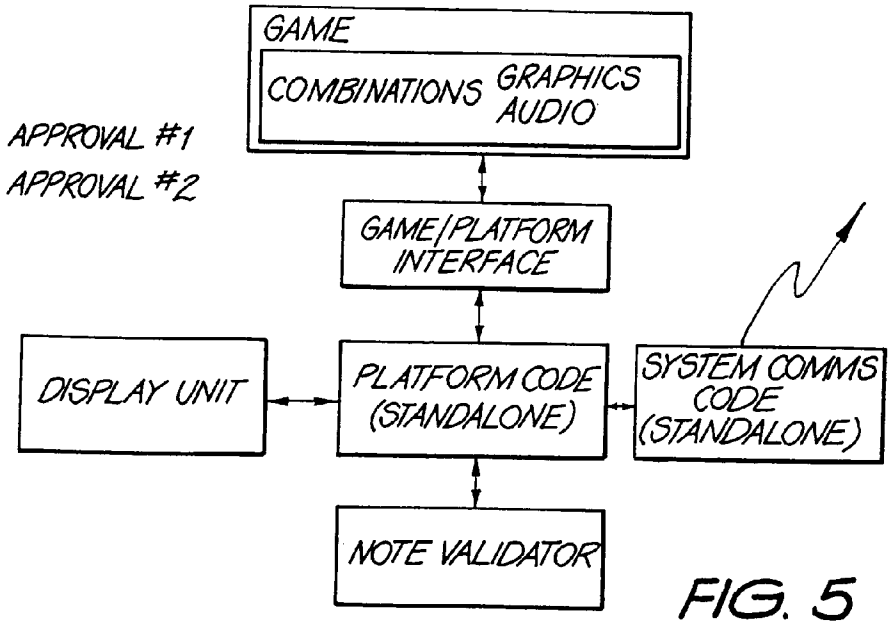

GAME
COMBINATIONS
GRAPHICS
AUDIO
APPROVAL #1
APPROVAL #2
GAME/PLATFORM INTERFACE
DISPLAY UNIT
PLATFORM CODE (STANDALONE)
SYSTEM COMMS CODE (STANDALONE)
NOTE VALIDATOR

FIG. 5

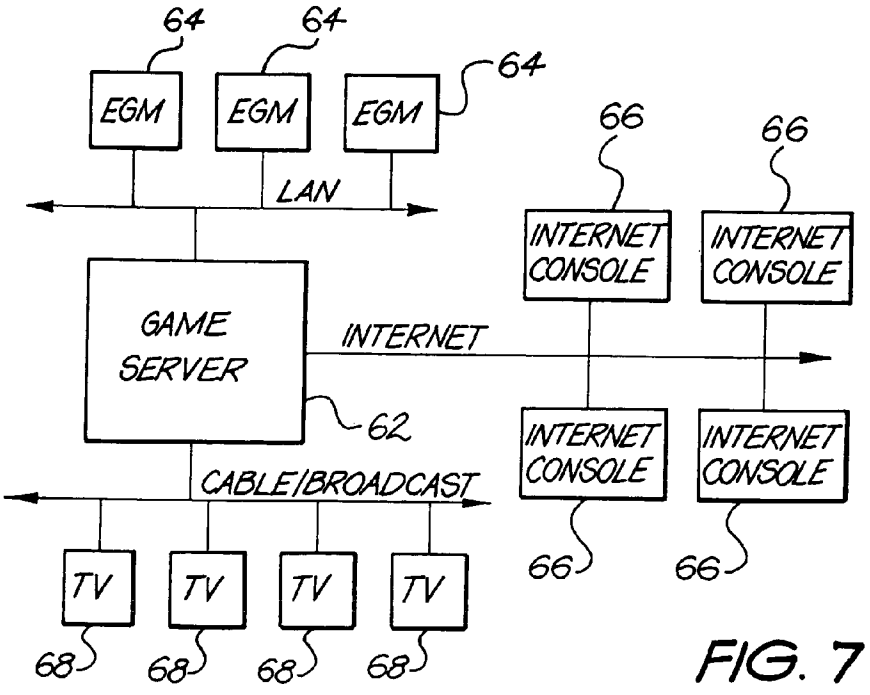

64
64
64
EGM
EGM
EGM
LAN
66
66
INTERNET CONSOLE
INTERNET CONSOLE
GAME SERVER
INTERNET
62
INTERNET CONSOLE
INTERNET CONSOLE
CABLE/BROADCAST
66
66
TV
TV
TV
TV
68
68
68
68

FIG. 7

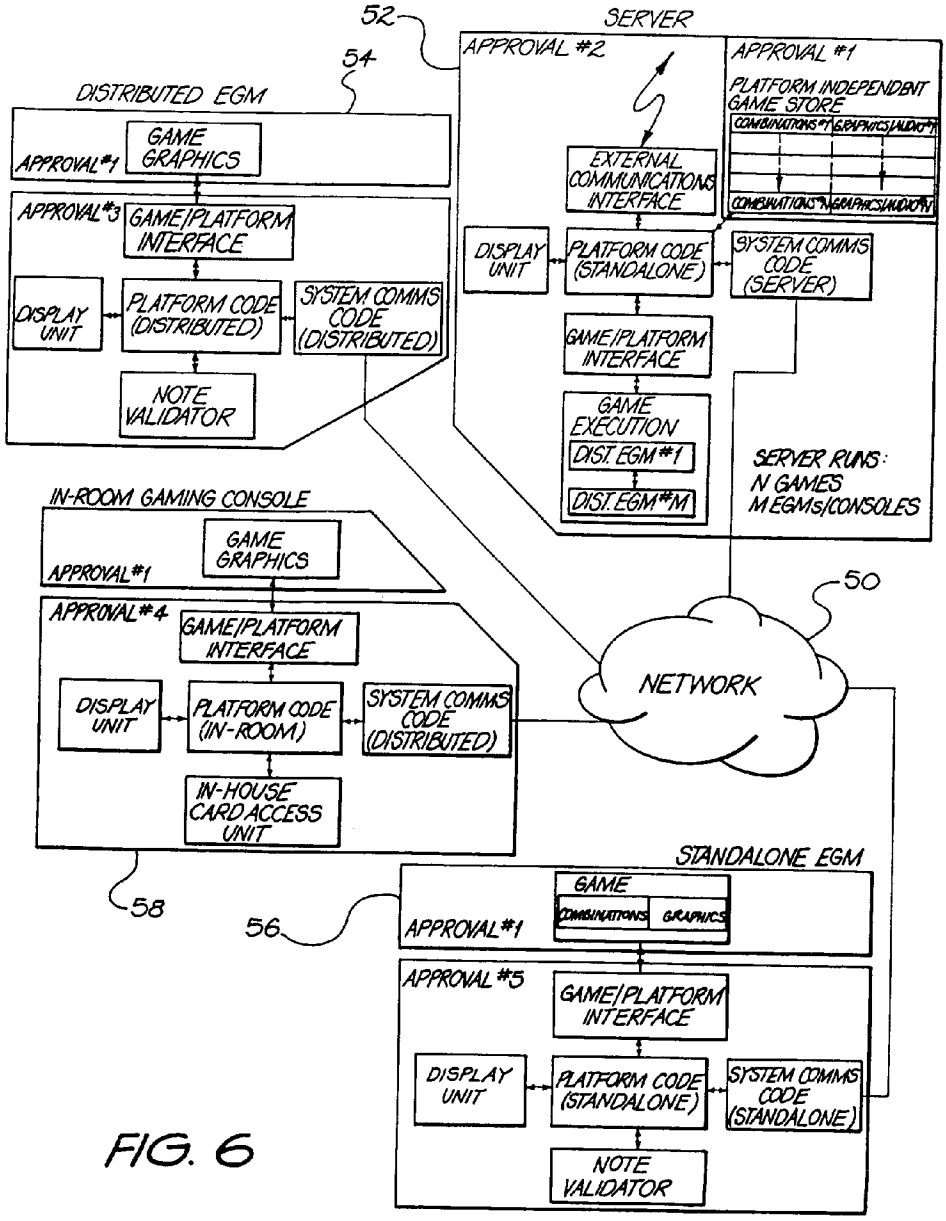

52
SERVER
54
APPROVAL #2
APPROVAL #1
PLATFORM INDEPENDENT GAME STORE
DISTRIBUTED EGM
GAME GRAPHICS
APPROVAL#1
EXTERNAL COMMUNICATIONS INTERFACE
APPROVAL#3
GAME/PLATFORM INTERFACE
DISPLAY UNIT
PLATFORM CODE (DISTRIBUTED)
SYSTEM COMMS CODE (DISTRIBUTED)
PLATFORM CODE (STANDALONE)
SYSTEM COMMS CODE (SERVER)
NOTE VALIDATOR
GAME EXECUTION
DIST. EGM #1
DIST. EGM #M
SERVER RUNS: N GAMES M EGMs/CONSOLES
IN-ROOM GAMING CONSOLE
APPROVAL#4
50
NETWORK
PLATFORM CODE (IN-ROOM)
IN-HOUSE CARD ACCESS UNIT
58
56
STANDALONE EGM
GAME
COMBINATIONS
GRAPHICS
APPROVAL#5
PLATFORM CODE (STANDALONE)
SYSTEM COMMS CODE (STANDALONE)

FIG. 6

MULTI-PLATFORM GAMING ARCHITECTURE

CROSS-REFERENCE TO RELATED APPLICATION

[0001] This is a divisional of pending U.S. application Ser. No. 09/238,535 filed Jan. 27, 1999.

INTRODUCTION

[0002] The present invention relates to gaming console architecture. The present invention seeks to provide an architecture which may be utilized on a wide variety of different platforms and which can reduce variation in games and allow them to use a wider variety of platform architectures.

BACKGROUND OF THE INVENTION

[0003] The "combinations" of a game describe the mathematical structure of the game and define all possible games, including the winning patterns and the payouts associated with each. From the combinations, the game statistics are determined, including the theoretical return to the player.

[0004] As the terms are used in this document, "combinations" and "graphics and audio" may contain both data and code.

[0005] Currently a number of gaming architectures exist and are suitable for implementing games on a wide variety of platforms.

[0006] 1. Standalone Electronic Gaming Machine (EGM). A standalone gaming machine contains all its functions within a secure environment. EGM's are commonly networked, primarily for data collection, bonusing and simple control.

[0007] 2. Distributed gaming, such as Internet or In-room gaming (Hotel), separates the user interface (console) from the secure gaming server. High level communications reduces the need to send high bandwidth graphics data. A single server controls multiple consoles. Graphic and audio data necessary for game display are stored on the server and downloaded to the console as required for game display. To maintain security all functions that may effect game outcomes and accounting are performed on the server. A variation on this architecture is the distributed gaming accelerator, in which the gamble outcome decision logic is implemented at the console in a smartcard. Both server and console run combinations, generate/verify game outcomes and perform accounting.

[0008] 3. TV allows the use of a television to play games. It separates the user interface (TV) from the rest of the gaming machine in a similar manner to the distributed gaming architecture, yet it could also be considered that a traditional EGM generates a display which is viewed via a remote TV screen. It requires more bandwidth than the distributed architecture as all picture information is generated at the server and sent in a low level, high bandwidth form to the user interface. A low bandwidth communication channel is required to pass user input back to the server.

[0009] 4. Hand held architecture. The secure functions of the EGM are implemented in a smartcard and all other functions in the unsecured console. The hand held architecture implements the secure functions within a smartcard. Limited storage capacity on the smartcard requires storage of graphics and combinations on the console for maximum flexibility. Variations on the storage location depend on the requirements jurisdictional and otherwise). The smartcard may implement fixed combination(s) or download secured (encrypted or digitally signed) combinations from the console. When smartcard storage capacity is sufficiently large the smartcard may store graphics and audio for download to the console. Table 1 shows the variety of platforms that may be implemented with these four fundamental gaming architectures.

TABLE 1

	Gaming applications and architectures			
	Architecture			
Platform	Traditional	Distributed	TV	Hand held
EGM	•	•		•
Hotel In-room		•		•
In-flight		•		•
On Ship	•	•	•	•
Internet		•		•
Cable TV		•	•	
Hand held				•

[0010] Split-Software Architecture

[0011] Referring to **FIGS. 1 and 2**, Split-software architecture has been previously proposed to allow games to be run independently of changes in the hardware platform. Further, the use of interpreted code allows games to be run on different microprocessors.

[0012] Split software architecture divides the gaming machine software into its two major elements. The game software contains all that software that is different between games, while the platform software contains the software that is common. In the traditional gaming machine, game software refers to the entire gaming machine software, while in the split software architecture, game software refers to that software that is different between "games". The exact meaning of game software is therefore context dependent. Typically game code includes graphics and sound data and combinations, while platform code includes all hardware drivers, kernel, accounting, security, operator interface, communications, etc.

[0013] In principal the two separate parts of the game can be approved independently, so that:

[0014] the platform EPROM is examined and approved only when it is changed and is tested with only a small subset of the possible games it will actually be used with. Hardware changes can be made to the platform with software changes limited to the platform. Games need not be re-approved.

[0015] The game EPROM is examined and approved without considering the version of the platform it will be used with.

[0016] Platform upgrades are feasible as only a single EPROM(s) need be submitted (for each market) to allow all games to run.

[0017] The present invention seeks to provide a game architecture which may by utilized on a wide variety of different platforms, which they can reduce the amount of approvals required and simplify game creation.

SUMMARY OF THE INVENTION

[0018] The present invention provides a gaming architecture including a game platform interface and a game program, the game program including a plurality of functional modules which interact via the platform interface. By providing such architecture in the game (software), the game can run on a variety of platform architectures without modifications of the game.

[0019] In one preferred embodiment, the game program includes a user interface module and a combinations module and communication of combinations to be displayed, are conveyed from the combinations module to the user interface module via the platform interface.

BRIEF DESCRIPTION OF THE DRAWINGS

[0020] Embodiments of the invention will now be described, by way of example with reference to the accompanying drawings in which:

[0021] **FIG. 1** diagrammatically illustrates a Split-Software Architecture;

[0022] **FIG. 2** diagrammatically illustrates in greater detail the upper layers of the Split-software of **FIG. 1**;

[0023] **FIG. 3** diagrammatically illustrates a variation in the upper layers of a Split-software architecture to provide a Multi-platform architecture in an EGM according to an embodiment of the present invention;

[0024] **FIG. 4** diagrammatically illustrates a variation in the arrangement of **FIG. 3** which is used in a Multi-platform Distributed Architecture;

[0025] **FIG. 5** diagrammatically illustrates a variation in the arrangement of **FIG. 3** which is used in a Multi-platform Standalone EGM;

[0026] **FIG. 6** diagrammatically illustrates Multi-platform Distributed Gaining System incorporating a number of different platforms; and

[0027] **FIG. 7** schematically illustrates the interconnection of a Game Server to the various components of a Multi-platform system.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0028] In a multi-platform architecture, according to a preferred embodiment of the present invention, the game software is split into separate functions, such that the functions can be distributed to, and run on, each of the platforms for which it is required that the game support.

[0029] After removing the platform code, what remains of a traditional monolithic game is principally the combinations and the graphics/audio. By splitting the game software into separate combinations and graphics/audio code which always interact with each other via the platform, the game can be run on a wide range of platforms. As used herein, the term "user interface module" is the totality of the presentation of the game to the user, and usually comprises the graphics and sound.

[0030] A single software architecture is described below which is capable of supporting diverse platform requirements. In principal an approved game can be run on each of the platforms without modification, and approval of the game on one platform is sufficient for approval on all platforms.

[0031] The four traditional gaming architectures described differ in where each of these functions of the game is stored and executed. In a traditional EGM both functions are stored and executed within the EGM. In a distributed system the server stores the entire game, but executes only the combinations, while the console executes the downloaded graphics and audio. The distributed gaming accelerator system stores games on the secure server, executes combinations on both server and console, and graphics/audio on the user console. In the handheld system the entire game may be stored either on the console or smartcard or split between them depending on implementation, but combinations are executed on the smartcard and graphics/audio on the console. The TV system is identical in this respect to a traditional EGM, but the graphics are viewed on the remote TV.

[0032] Traditional game software is compatible only with the architecture and hence platform for which it was designed. It cannot run on multiple platforms, as the components of the game are either monolithic (as in an EGM) or separated (as in a distributed system).

[0033] The two functions of the game, combinations and graphics/audio are separable, and the combinations may be secured (through encryption or digital signature or physically) to prevent tampering. The separation between the functions requires that the game API (application program interface) layer mediate communications between the functions.

[0034] When the game is run the game code is separated, as required by the platform architecture, into the appropriate functions and downloaded (where necessary) to the appropriate part of the platform. **FIG. 3** shows how the architecture is implemented in a traditional EGM. The combinations **10** and graphics and audio **12** are separated. They communicate with each other only through the API layer/platform interface **14**.

[0035] Each of the separate game functions (e.g. combinations, graphics/audio) may need to be secured to prevent tampering. If the function may be downloaded over an unsecured communication channel then the possibility of tampering exists. The consequences of tampering with combinations are particularly severe as it allows the payout of the game to be changed. Encrypting the data or creating a digital signature provides security. Encryption hides the data, however, a digital signature is generally quicker to authenticate.

[0036] In the distributed system the entire game is initially stored on the server. When the player requests a game, it is separated and graphics/audio are downloaded to the console and combinations are kept in the server. The same game in a traditional EGM is simply stored and executed unchanged.

FIG. 4 shows a distributed system generally indicated by **20**, with one server **22** containing 'N' games **24** and controlling 'M' consoles **26**. Clearly more than one server may be used.

[**0037**] Platform code is that software required to support a game (or part of a game), on a particular platform. It may perform the separation and distribution of game functions to those platforms for which it is required and provides communications where needed. Platform code exists for each platform within the gaming system and is in principal approved once for all games. A traditional EGM will have platform code for the EGM, while a distributed gaming system will have distinctly different platform code for the server and console(s). In practice platform code will typically contain code to recognize player inputs (push-buttons, handle, touch-screen, etc), drive player outputs (video, stepper reels, audio, etc) and drive machine accessories (printer, hopper, note-validator, security, etc). Depending on system implementation the system communications code may also be considered part of the platform code, but has been drawn separately in the Figures to aid in understanding the systems.

[**0038**] **FIG. 5** shows an exemplary implementation of a standalone EGM using the multi-platform architecture and shows the separate game and platform approvals.

[**0039**] **FIG. 6** shows a distributed gaming system generally indicated by **50**, comprising of a server **52**, distributed EGM **54**, standalone EGM **56** and in-room gaming console **58**. The architecture of distributed and in-room gaming EGM is essentially the same, with the main differences being in payment systems and physical design. Separate approvals are required for games (#1), server (#2), distributed EGM (#3), in-room EGM (#4) and standalone (#5) platform code. The standalone EGM may be monitored by and have games downloaded from the server.

[**0040**] In a casino, standalone EGM's are often connected to networks to allow monitoring and simple control. These networks can be extended to perform similar functions over the systems described, with, for example, distributed EGM's or the server itself being connected to these traditional networks. For compatibility the distributed EGM may emulate a traditional EGM. Alternately the network monitoring system may either be upgraded to understand the server or the server may emulate the appropriate number of standalone EGM's. Clearly both options may be implemented simultaneously.

[**0041**] In an extension to the architecture, the game may contain multiple graphic/audio and combination files, only one of which is used to play a particular game. One useful application is where various target platforms have different screen resolutions. Multiple graphics allow the best possible picture to be displayed. Where the target platform has only a very simple non-graphic interface one of the graphics files may cater for this. Different graphics may also allow the player to select their choice of "game". Different combinations cater for different player preferences, for example high win rates or large win values.

[**0042**] Partial replacements of the combinations and/or graphics/audio files allow the game to be partially modified, thus decreasing the storage requirements compared to storing complete copies of each possible games variation. For example, if a game is created that may be used with 50 different currencies a single main set of game graphics can be stored together with 49 different currency symbols. The total storage is far less than if 50 entire sets of game graphics were to be stored. Even worse, if 3 different symbols were to be selected, each from 50 possible, then the total number of variations is 125000 (50×50×50). Similarly audio and combinations may be partially replaced by equivalent data.

[**0043**] The multi-platform architecture allows embodiments incorporating other additional functions, which may be required if other aspects of a game are created or if new platforms are developed which require other functions of a game to run on one platform, but not another. In this way the multi-platform architecture can support a diverse range of platforms with a single game. The multi-platform architecture may be used in conjunction with interpreted code to achieve CPU independence on all platforms.

[**0044**] The architecture enables the creation of a generic "game server". The game server **62** stores games for execution and distribution to the various platforms, as shown in **FIG. 7**. The game server may therefore be used to distribute games to traditional EGMs **64**, Internet consoles **66**, televisions **68**, etc.

[**0045**] It will be appreciated by persons skilled in the art that numerous variations and/or modifications may be made to the invention as shown in the specific embodiments without departing from the spirit or scope of the invention as broadly described. The present embodiments are, therefore, to be considered in all respects as illustrative and not restrictive.

I claim:

1. A game system comprising one or more program processing components, the program processing components including a user interface console, and each of the program processing components being an instance of one of a plurality of logically different gaming program processing platforms and a set of game programs having a game program structure wherein the game program structure is capable of running on each of the plurality of different gaming program processing platforms, the program structure comprising:

one or more platform specific Gaming Application Programming Interfaces, each configured to run on a different one of the plurality of gaming program processing platforms; and

a plurality of platform independent game component modules comprising at least one game combination program module and at least one user interface program module, wherein each of the platform independent game component modules is capable of running on any one of the game program processing platforms, and communicates with the system and other platform independent game component modules in the plurality of platform independent game component modules only through the Gaming Application Programming Interface of the respective program processing component on which it is running;

wherein a game implementation, when run on the gaming system, comprises a Gaming Application Programming Interface running on each of the one or more program processing components, and a Game Set of platform independent game component modules each running on one of the program processing components, wherein

the Game Set of platform independent game component modules cooperate to provide functionality required to play a game on the system, the Game Set of platform independent game component modules including a user interface program module and a game combination program module, the user interface program module running on the user interface console to provide game progress and outcome information to the user in response to information from the combination program module; and

wherein the Gaming Application Programming Interfaces, when running on different program processing components, communicate with one another, whereby communication between the Game Set of platform independent game component modules, whether running on the same or different program processing components, communicate with one another via their respective Gaming Application Programming Interfaces to cooperatively implement the playing of a game on the system.

2. The game system of claim 1, wherein communication of game outcomes to be displayed, are conveyed from the game combination program module to the user interface program module via the respective platform specific Gaming Application Programming interface.

3. The game system of claim 1, wherein the user interface program module comprises a graphics generation program for generating game images on a user display.

4. The game system of claim 2, wherein communication between two or more of the gaming program processing platforms in the system is encrypted.

5. A gaming system as claimed in claim 3, wherein communication between two or more of the gaming program processing platforms in the system is secured by means of a digital signature.

6. The game system of claim 1, wherein communication between two or more of platform independent game component modules in the electronic gaming machine is encrypted.

7. The game system of claim 1, wherein communication between two or more of the platform independent game component modules in the electronic gaming machine is secured by means of a digital signature.

8. The game system of claim 1, wherein each game implementation comprises a plurality of files each file containing one instance of one type of platform independent game component module.

9. The game system of claim 8, wherein each game implementation comprises a plurality of user interface program module files each containing one user interface program module, each user interface program module providing a different game appearance or game style.

10. The game system of claim 8, wherein each game implementation comprises a plurality of game combination program module files each containing one game combination program module and each game combination program module providing a different set of game outcome possibilities.

11. A distributed gaming system incorporating the gaming system of claim 1, the distributed gaming system comprising:

a first program processing component acting as a server processing unit;

a server specific Gaming Application Programming Interface;

a plurality of second program processing components acting as gaming consoles

a plurality of gaming console specific Gaming Application Programming Interfaces, one console specific Gaming Application Programming Interface running on each gaming console; and

a plurality of games stored on the server processing unit, each game being implemented as a game set of platform independent game component modules, and

wherein the server specific Gaming Application Programming Interface located in the server processing unit functions to transfer at least one of the platform independent game component modules of one game set to a gaming console, the gaming console specific Gaming Application Programming Interface functions to enable execution of the at least one of the platform independent game components transferred to the gaming console, and the server specific Gaming Application Programming Interface functions to enable execution of the platform independent game components not transferred to the gaming console.

12. The distributed gaming system of claim 11, wherein communication of game outcomes to be displayed, are conveyed from the game combination program module to the user interface program module via the respective platform specific Gaming Application Programming interface.

13. The distributed gaming system of claim 11, wherein the user interface program module comprises a graphics generation program for generating game images on a user display.

14. The distributed gaming system of claim 11, wherein communication between two or more of the gaming program processing platforms in the system is encrypted.

15. The distributed gaming system of claim 11, wherein communication between two or more of the platforms in the system is secured by means of a digital signature.

16. The distributed gaming system of claim 11, wherein communication between two or more of platform independent game component modules in the electronic gaming machine is encrypted.

17. The distributed gaming system of claim 11, wherein communication between two or more of the platform independent game component modules in the electronic gaming machine is secured by means of a digital signature.

18. The distributed gaming system of claim 11, wherein each game implementation comprises a plurality of files each file containing one type of platform independent game component module.

19. The distributed gaming system of claim 18, wherein each game implementation comprises a plurality of user interface program module files each containing one user interface program module, each user interface program module providing a different game appearance or game style.

20. The distributed gaming system of claim 18, wherein each game implementation comprises a plurality of game combination program module files each containing one game combination program module and each game combination program module providing a different set of game outcome possibilities.

21. The distributed gaming system of claim 18, wherein the combinations module runs on the server processing unit to determine a game outcome, and wherein one or more platform independent game component module files including at least one user interface program module file are distributed to one or more of the gaming consoles for execution to display to a player playing a game on the respective gaming console, the game outcome determined on the game combination program module running on the server processing unit.

22. An electronic gaming machine incorporating the game system of claim 1, the electronic gaming machine comprising:

a program processing components acting as gaming console; and

a gaming console specific Gaming Application Programming Interface running on the gaming console

a game stored on the gaming console, the game being implemented as a game set of platform independent game component modules, wherein the gaming console specific Gaming Application Programming Interface functions to enable execution of all platform independent game components running on the gaming console.

23. The electronic gaming machine of claim 22, wherein communication of game outcomes to be displayed, are conveyed from the game combination program module to the user interface program module via the gaming console platform specific Gaming Application Programming interface.

24. The electronic gaming machine of claim 22, wherein the user interface program module comprises a graphics generation program for generating game images on a user.

25. The electronic gaming machine of claim 22, wherein communication between two or more of platform independent game component modules in the electronic gaming machine is encrypted.

26. The electronic gaming machine of claim 22, wherein communication between two or more of the platform independent game component modules in the electronic gaming machine is secured by means of a digital signature.

27. The electronic gaming machine of claim 22, wherein the game set of platform independent game component modules comprises a plurality of user interface program module files each containing one user interface program module, each user interface program module providing a different game appearance or game style.

28. The electronic gaming machine of claim 27, wherein the game set of platform independent game component modules comprises a plurality of game combination program module files each containing one game combination program module and each game combination program module providing a different set of game outcome possibilities.

29. A server for a distributed gaming system incorporating the game system of claim 1, the server comprising:

a first program processing component acting as a server processing unit;

a server specific Gaming Application Programming Interface;

a plurality of games stored on the server processing unit, each game being implemented as a game set of platform independent game component modules; and wherein the server communicates with a plurality of second program processing components acting as gaming consoles and running a plurality of gaming console specific Gaming Application Programming Interfaces, one console specific Gaming Application Programming Interface running on each gaming console, and wherein the server specific Gaming Application Programming Interface located in the server processing unit functions to transfer at least one of the platform independent game component modules of one game set to a gaming console when it is run, and the server specific Gaming Application Programming Interface functions to enable execution on the server of the platform independent game components not transferred to the gaming console.

30. The server of claim 29, wherein communication of game outcomes to be displayed, are conveyed from the game combination program module to the user interface program module via the respective platform specific Gaming Application Programming interface.

31. The server of claim 29 wherein the user interface program module comprises a graphics generation program for generating game images on a user display.

32. The server of claim 29 wherein communication between the server and each gaming console in the system is encrypted.

33. The server of claim 29, wherein communication between the server and each gaming console in the system is secured by means of a digital signature.

34. The server of claim 29, wherein each game implementation comprises a plurality of files each file containing one type of platform independent game component module.

35. The server of claim 34, wherein each game implementation comprises a plurality of user interface program module files each containing one user interface program module, each user interface program module providing a different game appearance or game style.

36. The server of claim 34, wherein each game implementation comprises a plurality of game combination program module files each containing one game combination program module and each game combination program module providing a different set of game outcome possibilities.

37. The server as claimed in claim 36, wherein the combinations module runs on the server processing unit to determine a game outcome, and wherein one or more platform independent game component module files including at least one user interface program module file are distributed to one or more of the gaming consoles for execution to display to a player playing a game on the respective gaming console, the game outcome being determined on the game combination program module running on the server processing unit and communicated to the respective gaming console.

* * * * *