Title: SYSTEMS AND METHODS FOR A GAMING PLATFORM

Abstract: Methods and systems, according to various embodiments, are provided for a gaming platform that includes one or more various features including a sector navigator, a noise channel, and/or pods. The gaming platform can optionally be implemented as an online game that is accessible through a traditional browser or other application capable of accessing content on the Internet. Additionally, methods and systems for smart loading the gaming sectors and pods are provided such that a user of the game is subjected to minimal gameplay delays waiting for content to load.
Published:

— with international search report (Art. 21(3))
SYSTEMS AND METHODS FOR A GAMING PLATFORM

CROSS REFERENCE TO RELATED APPLICATIONS

This application claims priority to U.S. Patent Application No. 12/037,062, entitled SYSTEMS AND METHODS FOR A GAMING PLATFORM, filed February 25, 2008, which is hereby incorporated by reference in its entirety. This application is related to U.S. Patent Application No. 11/844,884, entitled SYSTEMS AND METHODS FOR MULTI-PLATFORM TRADING CARD GAME, filed August 24, 2007, the disclosure of which is hereby incorporated by reference in its entirety.

BACKGROUND

1. Field of the Invention

The present invention relates to a gaming platform. More specifically, the present invention relates to unique features within a gaming platform and/or loading of the gaming platform.

2. Discussion of the Related Art

Games that are played on personal computers (PC), such as Microsoft Windows based computers, Apple computers, Linux based computers, mobile devices, and handheld devices, have been popular for many years. More recently, such games have integrated online capabilities to facilitate new features, such as user interaction with other players. With online capabilities, users can play with or against other players, can chat with other players, and can download updates to their game. Most games require the installation of local software to operate. Most if not all of the software files and executables of the game reside on the user's PC to be executed when the user desires to play the game. Because of the many different types of PCs
now available to users, developers are required to create and distribute different software versions of the same game. For instance, different software is required for a user who operates a Windows based PC than for the user who operates a Linux based PC. Due to the costs of creating different software versions of the same game, developers limit the number of versions they create, leaving those who use less common PCs without the ability to purchase and enjoy the game. In addition, games that are implemented using local software typically save user progress in one or more files that are stored in the local memory of the user's PC. As such, any progress a user makes in the game may not be accessible from a remote PC. That is, a user cannot play the game from a remote PC and continue where he/she left off when playing last at their local PC.

Developers have tried to solve some of the problems discussed above by implementing web-based games. Web-based games, unfortunately, can suffer from their own problems. The quality of a web-based game is highly dependent on a user's bandwidth. Users with low bandwidth, for instance, speeds slower than 3Mbit/sec, can experience choppy gameplay and long load times between portions of games. Developers try to minimize these problems by producing less visually pleasant games, games with less complex scenes and/or scenarios, and games with less interactivity between the environment and the user. These solutions and implementations detract from the overall quality of a game and in turn the user's experience.

Additionally, developers have also failed to leverage games with online capabilities into viable marketing tools. Marketing and advertising is severely limited in current games. In most cases, marketing is limited to "product placement" within games. That is, a character in a game might drink a Coca-Cola product. Further, some web-based games that are played within a web browser feature advertising banners that appear on a website, but outside of the gaming window. These banners are not physically contained within the display area of the gaming environment, are not integrated into the gameplay, are not dynamic, and are not specifically targeted to a specific user and their interaction with the game.
SUMMARY

The present embodiments relate to a gaming platform. The gaming platform can include one or more various features including a sector navigator, a noise channel, and/or pods. Additionally, in some embodiments the gaming platform is a browser based gaming platform that can be downloaded over the Internet or may be installed using an installation disk.

One embodiment can be characterized as a gaming platform comprising a main gaming area; a plurality of sectors, each of which can be displayed within the main gaming area; and a sector navigator for moving gameplay from a first sector to a second sector. Optionally, the gaming platform includes a plurality of links, wherein each link represents a sector within the game; and wherein the sector navigator is a visual representation of each of the plurality of sectors.

Another embodiment can be characterized as a method of providing navigation for a game comprising displaying a first sector of the game within a main gaming area; receiving an input at a sector navigator to move to another sector; and displaying a second sector of the game within the main gaming area in response to receiving the input. Optionally, the sector navigator includes a plurality of links, wherein each link represents a sector within the game; and wherein the sector navigator is a visual representation of a plurality of sectors of the online game.

Yet another embodiment includes a gaming platform comprising a main gaming area; and a noise channel displayed within a portion of the main gaming area, the noise channel providing content to a user that is related to a game being played within the main gaming area. Optionally, the noise channel provides video clips, commercials, advertisements or news from the game.

Still yet another embodiment can be characterized as a method of providing content to a user of a game comprising providing gameplay within a main gaming area; providing a noise channel within a portion of the main gaming area; and providing content through the noise channel, the content corresponding to the gameplay within the main gaming area. The method optionally further comprises providing video clips, commercials, advertisements or news from the game through the noise channel or providing an alert to a user through an alert window of the noise channel. As another option, the method can further include receiving an input for
selecting content from a menu of the noise channel; and providing the selected content in the noise channel in response to the received input.

Another embodiment includes a method of smart loading an online game comprising loading a first sector, wherein gameplay starts in the first sectors of the game; determining a second sector to load based upon a likelihood that a user will navigating to the second sector instead of a third sector upon leaving the first sector; and loading the second sector of the game before loading the third sector of the game. Optionally, the method further includes loading a plurality of pods in sequential order based upon a determination of the most likely pods to be first needed by the user.

Yet another embodiment can be characterized as a gaming platform comprising a main gaming area; and a first pod displayed within a portion of the main gaming area, wherein the pod is a user movable object that appears above the main gaming area. Optionally, the gaming platform further comprises a second pod displayed within a portion of the main gaming area; wherein the first pod is a parent pod and the second pod is a child pod.

**BRIEF DESCRIPTION OF THE DRAWINGS**

The above and other aspects, features and advantages of the present invention will be more apparent from the following more particular description thereof, presented in conjunction with the following drawings, wherein:

**FIG.** 1 is a diagram illustrating a gaming platform, according to one embodiment, displayed within a browser application window and including a sector navigator, a noise channel, and a login pod;

**FIG.** 2 is a diagram illustrating a gaming platform, according to an embodiment where, the gaming platform is implemented in a full screen mode;

**FIG.** 3 is a diagram illustrating the layout of the sectors of a game along with their associated functionality according to one embodiment;

**FIG.** 4 illustrates several different embodiments of a sector navigator;

**FIG.** 5 illustrates a pod according to one embodiment;

**FIG.** 6 depicts the interaction between the gaming platform and a mobile phone according to one embodiment;

**FIG.** 7 illustrates parent and child pods and various user interactions with the pods according to one embodiment;

**FIG.** 8 illustrates a noise channel according to an embodiment;
FIG. 9 illustrates the noise channel including a welcome pod according to one embodiment; and

FIG. 10 depicts a noise channel with an adjacent alert according to another embodiment.

Corresponding reference characters indicate corresponding components throughout the several views of the drawings. Skilled artisans will appreciate that elements in the figures are illustrated for simplicity and clarity and have not necessarily been drawn to scale. For example, the dimensions, sizing, and/or relative placement of some of the elements in the figures may be exaggerated relative to other elements to help to improve understanding of various embodiments of the present invention. Also, common but well-understood elements that are useful or necessary in a commercially feasible embodiment are often not depicted in order to facilitate a less obstructed view of these various embodiments of the present invention. It will also be understood that the terms and expressions used herein have the ordinary meaning as is usually accorded to such terms and expressions by those skilled in the corresponding respective areas of inquiry and study except where other specific meanings have otherwise been set forth herein.

DETAILED DESCRIPTION OF THE DRAWINGS

The following description is not to be taken in a limiting sense, but is made merely for the purpose of describing the general principles of the invention. The scope of the invention should be determined with reference to the claims. The present embodiments address the problems described in the background while also addressing other additional problems as will be seen from the following detailed description.

Referring first to FIG. 1, shown is a diagram illustrating a gaming platform 100 including a sector navigator 102, a pod 104, a noise channel 106, and a main display area 108. Also shown is a browser application 110.

As shown in the present example, the gaming platform 100 inhabits a defined area within the browser application 110 where a game is displayed. The gaming platform 100 occupies substantially all of a viewable area within the browser. However, in various other embodiments, the gaming platform will occupy less than the viewable area or more than the viewable area such that scroll bars are needed to view the entire gaming platform. In one embodiment, the gaming platform 100 is
implemented as an Adobe Flash object. Where the gaming platform 100 is implemented using Flash, the gaming platform 100, in one embodiment, is embedded in an HTML page and executed by a Flash compatible plug-in for the browser application 110. The Flash object communicates with a remote gaming server. Gaming content is transmitted from the remote gaming server to the Flash object, while control instructions, game saves, and messages are sent from the Flash object to the remote server to control the progress of the game. Those skilled in the art will recognize that the gaming platform 100 can be implemented using technologies other than Flash, such as Scalable Vector Graphics (SVG), Synchronized Multimedia Integration Language (SMIL), Java, Asynchronous Javascript and XML (AJAX), and Microsoft Silverlight. Such technologies also permit the gaming platform 100 to operate in conjunction with a standard browser application 110. In other embodiments, the gaming platform 100 is executed by a standalone player external from the browser application 110 or other specialized program used to access the web.

In some embodiments, the gaming platform 100 uses Flash SOL files to maintain user preferences and current activity data; in other embodiments, standard cookies or other methods are used. Using the aforementioned techniques, the gaming platform 100 can be configured to load portions of the game, such as specific sectors (discussed in more detail below) before a user is even authenticated. Alternatively, after a user is authenticated, the gaming platform 100 can be configured to access and load remote objects that maintain the user's preferences and current activity data. Once the remote objects are loaded the user can resume his or her game from any device with a standard web browser, even if the user has never used the device before. The access and loading of remote objects can be implemented using, for example, Windows Servers running .NET 2.0 with SQL SP2 Databases and an Adobe Flex front end.

The gaming platform 100 does not have to be visibly bounded by the browser application 110. Thus, in one embodiment, as illustrated in FIG. 2, the gaming platform 100 is used in a "full screen" mode, where only the gaming platform 100 is visible to a user on his or her PC screen 202. The gaming platform 100 is operable in "full screen" mode regardless of whether it is executed within the browser application 110 or by a standalone player.

Referring back to FIG. 1, the game implemented by the gaming platform 100 includes one or more sectors which are generally displayed in the main
display area 108. Generally, a single sector will be displayed within the main display area 108. Each sector represents a different virtual location within the universe of the game. For instance, consider a virtual village that has a town square, a courthouse, a market, a church, and a bank that are all connected by a street. Each location, as well as the street, is represented as a different sector within the game implemented by the gaming platform 100. Collectively, the sectors make up a spatial configuration that represents the virtual village. Sectors allow a user to have direct access to multiple features of the game at the same time, without being limited to the physical demands of a monitor. Unlike the virtual village discussed above, sectors need not be areas typically thought of as logically or physically contiguous. In one embodiment, for example, the sectors are a community, a web forum, a gameplay background story area, a real-time chat area, a welcome screen, a rankings area, a market area, a card management area, a gameplay tips area, and a battle area. Each of these sectors provides different functionality for playing the online game. As different games are developed, these sectors will change. These sectors will be described below with reference to FIG. 3.

In order to facilitate smooth gameplay over the web, in some embodiments, the gaming platform 100 permits a user to view at least a portion of a sector within the main display area 108 even if the sector has not been completely loaded. This is accomplished, in some embodiments, by building sectors using sub-components. Each sub-component implements a specific functionality within the game, such as displaying a user's current score. Once a sub-component is loaded by the gaming platform 100 its functionality is immediately available to the user, even if the sector the sub-component is a part of is not completely loaded. In some embodiments, a single sub-component can be utilized by more than one sector or to implement the gaming platform 100 itself. Thus, once a sub-component is loaded for one sector, it can nearly instantaneously be used to implement other sectors across the gaming platform 100 without having to be reloaded.

In one embodiment, the gaming platform 100 intelligently loads sectors of the game over the Internet (or other network) before they are desired to be accessed by the user during gameplay. Intelligent loading of sectors is accomplished in several different ways. In some embodiments, where bandwidth and computational capabilities are permitting, all the sectors of the game are loaded prior to, or at the start of, game play. In other embodiments, the sectors of the game are loaded in a
default order during startup of the game such that there is minimal to no delay in the
game when a user moves from one sector to another. Once a user's behavior within
the gameplay is learned, specific sectors are dynamically loaded using intelligent
loading rules based on previous user behavior and/or by anticipating the user's next
course of conduct based on their current activity. Such dynamic loading makes game
play as smooth and seamless as possible. For instance, if a user has a history of
consistently visiting the User Rankings sector at least once during a gameplay session,
the User Rankings sector will be loaded during the game's startup so that the User
Rankings sector will be immediately available when desired to be viewed by the user.

In another example, if a user is currently viewing the User Rankings sector, the
system can anticipate that the user will next view either the ranking of the user with
the highest ranking or the user's own ranking and load such rankings before loading
other user's rankings. Additionally, the next sector a user will likely visit after
viewing the User Ranking sector can be anticipated and loaded from a server
accordingly. Intelligent loading is not limited to loading only the next sector a user is
likely to visit; sectors can be loaded as many steps ahead of the user as connection
speeds and current transfer requests allow.

In some embodiments, such intelligent loading rules are implemented
using standard artificial intelligence and/or data mining techniques, the
implementation of which are well known to those skilled in the art. In other
embodiments, the intelligent loading rules are specified by weights given to each
possible option presented to the user. This can be based upon a specific user's
previous actions and/or the actions of everyone who plays the game. A weighted
calculation can be made that anticipates the one or more most likely decisions a user
will next make. The higher the weight, the more likely the option and its resulting
screen, pod, or sector will be preloaded. In some embodiments, the intelligent loading
rules are continuously reevaluated to remain accurate and effective. This preloading
helps to provide users with completely transparent game loading and removes
undesirable delays when playing the game due to certain features not being available
as they have yet to be downloaded. The computations leading to the intelligent
loading rules discussed above are also flexible. In some embodiments, the
computations are carried out locally on the user's local PC. In other embodiments,
the computations are done remotely at the gaming server and, optionally, the final
intelligent loading rules are transmitted to the local user. As an alternative, the user's
local PC and the remote server can dynamically allocate the computation of the intelligent loading rules between the two machines based on signal response time, computation loads, or other factors. The allocation of computation of the intelligent loading rules is managed by the gaming platform 100 in conjunction with the gaming server using standard techniques well known to those skilled in the art.

Intelligent loading has been discussed above with reference to sectors, though persons of ordinary skill in the art will recognize that the techniques and methods discussed can be used to load the sub-components of a sector or other components that make up the gaming platform 100. In addition, while the intelligent loading procedures have been described above with reference to a completely downloadable version of the gaming platform, in other embodiments of the invention, the gaming platform may be loaded (fully or partially) through the use alternative storage media, such as a load disk, CD-ROM, CD, DVD, memory stick, or other memory storage device.

In some embodiments, the gaming platform 100 may include sectors or other components that are simply too large to efficiently download over the Internet during the course of game play. Examples of such components may include expansive 3D models of creatures, high resolution movie clips, game sounds, etc. In these embodiments, alternative storage media can be used to provide the source and content for the gaming platform 100. Users can obtain the requisite content through traditional outlets, such as a brick and mortar stores, or through a separate online download. Alternatively, if such alternative storage media devices are not properly connected or logically available to the user, substitute content can be provided by the gaming server or the gaming platform 100 will display a "request to insert media."

The features of the sector navigator 102 will next be discussed. The sector navigator 102 is used to move from one sector of the game to another. In one embodiment, as shown, the sector navigator 102 includes nine links taking the form of squares. Each link represents a sector within the game. The sector navigator not only provides links, but is also a visual representation of the entire playing area of the game. The embodiment shown includes nine sectors that are configured to be positioned next to each other such as illustrated by the sector navigator. When a link in the sector navigator 102 is clicked, the link's corresponding sector is navigated to and displayed within the main display area 108 of the gaming platform 100. Only one sector is typically shown at a time within the main display area 108. In some
embodiments, the transition from sector to sector is intuitive. For example, if a user is currently viewing the sector associated with the bottom right square of sector navigator 102, and the user clicks on the top left square of sector navigator 102, the sector corresponding to the top left square will be displayed in the main display area 108. Furthermore, in some embodiments, the transition between the two sectors is dynamic and the main display area 108 will display in real time the actual movement from the bottom right portion of the virtual world to the top left portion of the world, including movement over the middle sector. In this manner, the user will see the actual navigation from one sector to another in a dynamic fashion.

As described, the sector navigator 102 of the present embodiment is shown as a one dimensional box that includes nine links shaped as squares. Generally, the sector navigator 102 will include a link for each sector contained in the game. Additionally, the sector navigator provides a visual representation of how the sectors in the game are located with respect to each other within the gaming platform 100. In other embodiments, the sector navigator 102 includes more or less links that correspond to more or less sectors of a game. Optionally, only some links and their corresponding sectors are accessible to a given user. Thus, for example, an unregistered game user may only have access to some of the sectors of the game, in which case only some of the sector links will function as normal.

Referring next to FIG. 3, shown is a representation of the layout of the various sectors of a game. The sectors shown are only exemplary and will be dependent upon the game that is being played. Other games will have different sectors, more or less sectors, or, for example, can have a three-dimensional relationship between the sectors.

The nine sectors of the game shown in FIG. 3 correspond to the sector navigator 102, which, as shown in FIG. 1, also includes nine links that correspond to the nine sectors. In order to move from one sector to another in the game, the user clicks on the corresponding link in the sector navigator 102. Thus, for example, if a user wanted to travel to the community sector 332, a user would select the link in the upper left hand corner of the sector navigator causing the game to navigate to the community sector 332. In this manner, the sector navigator is a visual representation of the layout of the various sectors in the game and can be used for easy navigation within the game.
The community sector 332 allows a user to open and make changes to a profile and a buddy list, manage account settings, manage an avatar, manage various devices (e.g., a cell phone), manage alerts and notifications, manage assets (e.g., points), use an instant messenger, or give away game items to a friend.

The market sector 334 can be used to trade cards, view available cards, view, search and filter the market area, post trades, view trade history, accept trades, get trade card value help, and set notifications.

The card management sector 336 is used to upload cards, build and manage decks of cards, align and manage armies, manage and sort a user's portfolio, and manage assets and draccoins (e.g., potions, downloads, skins, ringtones).

The web forum sector 326 is used as a message board where users can post messages, give away assets to a friend, and provides moderator controls for those who have moderator access.

The welcome sector 328 is used for registration and login, parent controls, retail products, game information, television show listings, news, events, newsletter, and legal disclaimers.

The ranking sector 330 is used for displaying league standings, challenges, and setting up tournaments.

The background sector 320 is used to provide a history of the game including a story about the various creatures and locations and details on the television show that is related to the game.

The gameplay tips sector 322 is used to provide quick start instructions, rules, tips and tricks, ask questions, provide answers to FAQ's, information on retail locations and a deck doctor.

The battle sector 324 is where the game is actually played and where battles take place. Here a user will have a system defined IM presence and will be able to manage play options.

Referring now to FIG. 4, the sector navigator and its links can take many forms depending upon the number of sectors implemented in a game and the relationship between the sectors. In some embodiments the sector navigator 302 appears to be three dimensional, and has links defined around, for example, a pyramid or a sphere. This will correspond to a game that has sectors corresponding to a pyramid or sphere, respectively. Still alternatively, the sector navigator 304 has links defined at different levels or depths. In this case, the sector navigator 304
corresponds to a multi-dimensional game and thus allows a user to navigate to sectors at different dimensions, i.e., sectors that are not in contiguous virtual planes within the game.

In some embodiments, the sector navigator 306 is also dynamic. For instance, the sector navigator's 306 layout or links can change depending on the location of the user within the game, the action occurring in the game, or based on programming provided by the user. A new link may appear within the sector navigator 306 when a user discovers a new sector in the game. In cases where sectors move with relation to each other, the links within the sector navigator 306 will also move accordingly. In some embodiments, the links in the sector navigator 308 are also dynamic and provide a visual indication of what is taking place within the sector. Thus, the links can change in size or appearance to correspond to the action or the status of their corresponding sectors. For example, when a sector is being loaded, its corresponding link may graphically fade into view. If there is an explosion in the gameplay of one sector, its corresponding link may shake or turn red. The links can change form depending on the preferences of the game developer, the user, or the circumstances of the game.

In some embodiments, the sector navigator 310 also includes a hierarchical menu 312. The menu 312 provides easy access to user preferences, specific pods and many other features. The menu 312 changes based on the specific sector currently being viewed and will display sector specific controls for the given sector and/or its pods. In addition, in some embodiments, the links of the sector navigator 310 also include roll over menu details 314 that can be clicked for more information to be displayed as the developer sees fit.

Referring now to FIG. 5, illustrated is a pod according to an embodiment. As shown in FIG. 1, each sector is can include one or more pods 104. Pods 104 are objects that appear in each sector in the main display area 108, and provide nearly all the tools and functions necessary to operate (i.e. play) a game implemented by the gaming platform 100. In some embodiments, the pods appear "above" or are "windows" within the gaming area. The tools and functions of the pods are used to interact with the sectors and other pods 104 that make up the game. A pod 104 can be sector specific or persistent. A sector specific pod depends on content within a given sector, in some cases inherits information from the sector, and is generally not visible or functional outside of the given sector. A persistent pod
functions within multiple sectors and is not dependent on a given sector to operate. For instance, a Battle pod offers the same fighting techniques and instructions to be implemented within all sectors, and would not significantly change from sector to sector. A user could initiate the Battle pod in any sector and it would include the same functionality. The Battle pod would thus be a sector persistent pod. A User Login pod, on the other hand, would only appear when a user is logging into the game and is viewing the welcome sector 328. The User Login pod would thus be a sector specific pod.

The pod 402 shown in FIG. 5 includes a title 404, basic controls 406, an options area 408, content space 410, action controls 412, and a skin 414 in accordance with one embodiment. The basic controls 406 allow a user to control the pod 402 and are generally independent of the specific pod content. The basic controls 406 include the ability to close the pod 402, enlarge the pod 402, minimize the pod 402, or get more information about the pod 402. The options area 408 includes controls that allow a user to control properties of the pod 402, such as locking or unlocking the position of the pod 402, and configuring pod settings, such as, for example, transparency, user mode, the pod skin, permissions and others. The content space 410 is where the functional content for the pod 402 is displayed. The content space 410 can display images, dynamic content, play audio or video, or anything else desired by the game developers. The action controls 412 provide actions a user can take in response to content displayed in the content space 410 or the displayed sector. For instance, the user could rewind a video displayed in the pod 402, or open a new pod in response to a battle taking place in the displayed sector. All the elements of the pod 402 are configurable, and can take any shape desired by the game developers.

In some embodiments, the elements can also be configured by the user. In addition, elements can be added or subtracted, either by the user or during the course of gameplay. For instance, if a game developer wants to create a pod for playing commercials, the pod will not include basic controls or action controls, thus preventing the user from skipping or closing the commercials. In another example, if a user is participating in a battle, then the user may not be permitted to access the options area 408. Alternatively, a user may desire custom controls for the Battle pod. The user can create unique controls with specific custom functions and include those controls in the Battle pod.
Referring back to FIG. 1, in one embodiment, pods 104 are made up of one or more SWF files. Pods 104 are hierarchical and content driven, which permits the efficient use of computation and bandwidth resources of a user's computer while providing a highly interactive user experience. For instance, a child pod inherits tools and functions of its parent, and can be configured to move and be displayed in conjunction with its parent without any coding. In some embodiments, changes to the parent pod 104 are inherited by the child pod, while changes to the child pod would generally not alter the parent pod 104. The content, size, position, functions, appearance, state, and hierarchy of each pod 104 are dynamic and can be created or altered in real-time based on a user's current progress within the game. For instance, some pods 104 appear immediately when navigating to and viewing a sector. Other pods 104 appear after completing certain tasks within a sector. Some pods 104 are altered or disappear based on the state of the current sector. Pods 104 can also be viewed, modified, or destroyed by other pods 104 within a given sector. TABLE 1 and TABLE 2 depict typical attributes that pods 104 can have. Each pod only needs to have one or more of these attributes, however, is not required to have all of these attributes.
As pods 104 are content driven, the underlying code that drives the intelligence and operation of each pod 104 is also alterable based on the current state of the sectors and other pods within the gaming platform 100. In some embodiments,
the code is modified using XML calls. Because the content and state of each pod 104 and sector can be shared across the gaming platform 100, in one embodiment, each graphic, line of code, piece of text or audio must only be loaded once, greatly reducing bandwidth requirements.

In some embodiments, pods 104 are loaded according to the same intelligent loading techniques discussed above with reference to sectors. Thus, the gaming platform 100 intelligently loads pods 104 as necessary throughout the game. The pods 104 then appear nearly instantaneously once they are requested to appear during gameplay. As described above, in some embodiments, all or a portion of the gaming platform is loaded using a load disk. Thus, all or portions of pods 104 can also be loaded by downloading them over the Internet or from a load disk.

The design and implementation of pods 104 within the gaming platform 100 makes for easy extension of games onto devices such as mobile phones, PDAs, GBAs, and other third party devices. From a development standpoint, each such device has the same attributes and is interacted with the same way as a pod 104. Thus, such devices can be treated as a pod 104 for programming and development purposes, making the gaming platform 100 extremely flexible. For instance, referring to FIG. 6, a user may be in a card management sector 336 manipulating a digital playing card 502 using a pod 504. When the user drags the pod 504 with the card 502 to a corner of the main display area 108, a connection request from the user's PC 506 to a mobile phone 508 (or other electronic device) is transmitted. After communication is established between the user's PC 506 and the mobile phone 508, the digital card 502 is transmitted to the mobile phone 508, and is removed from the card management sector 336. Such operation is an intuitive and interactive way to move the card 502 from the game defined within the gaming platform 100 to a third party device such as the mobile phone 508. The gaming platform 100 thus provides the unique ability for a user to manage their card portfolio on his or her mobile device. Because "real world" playing cards can be copied and used in the "virtual" game implemented using the gaming platform 100, in some embodiments, the gaming platform 100 only permits one electronic copy of a unique card to exist at any given time in any given place within the game. Since it is likely that numerous "real world" cards will exist at any given time and be owned by different people, such restriction reduces the practical logistical problems surrounding potential duplicity of the electronic card.
In one embodiment, pods 104 have collision detection technology. Using a pod's 104 internal coordinates in conjunction with the location of the main display area 108, a pod 104 can compute when it has been moved to or past the boundary of the main display area 108. If a pod 104 is moved to such a location, the pod 104 can be configured to invoke predefined rules, such as connecting to a mobile phone, as discussed above.

Referring now to FIG. 7, a diagram is shown illustrating parent and child pods and various user interactions with the pods according to one embodiment. Shown is the sector navigator 510, a main gameplay area 512 and a pod display area 513. The pod display area 513 is used in the present example, to show some of the various relationships between parent and child pods. It by no way limits where pods can be displayed within the gameplay area 512. Pods generally will be able to be displayed anywhere within the main gameplay area 512. Additionally, the pods can have a hierarchical relationship and events in one pod can affect the content of another pod.

When a user interacts with something in the game (e.g., a menu of the sector navigator 510) this can cause a pod to appear in the game. For example, when a user interacts with the menu in the sector navigator 510, this can cause a parent pod 514 to appear in the pod display area 513. Next, depending upon an event in the game, a user interaction or an event in the parent pod 514, the parent pod 514 causes the first child pod 516 and the second child pod 518 to open. The child pods can move along with the parent pod within the main gameplay area 512 or can move independently. For example, each of the pods acts like a separate "window" within the main gameplay area 512. When a user drags the parent pod 514 to a new location, the first child pod 516 may move along with the parent pod 514 and the second child pod 518 may stay in its present location. This demonstrates one of the hierarchical relationships that can exist between a parent and child pod.

As another example, when a user interacts with the first child pod 516, this can cause a content update in one or both of the second child pod 518 and the parent pod 514. Similarly, a user interaction with the second child pod 518 can, for example, cause a content update in the first child pod 516. Some pods will only be shown as long as another pod is also shown. Thus, for example, in one embodiment, if a user closes the first child pod 516, the second child pod 518 will also close.
Reopening of the first child pod 516 and the second child pod 518 can then be initiated by the parent pod 514.

The above examples demonstrate some of the dependencies between different pods within the game. As described above, in some embodiments, these pods are all dynamically created during gameplay and can be smart loaded such that there is no delay noticed by the players during gameplay.

Referring now to FIG. 8, the noise channel 106 is a persistent pod that provides players with information from a variety of sources during their gameplay. The noise channel 106 is similar to television channel, however, in one embodiment, the noise channel 106 cannot be turned off by the player of the game. In one embodiment, the user has minimal control over the noise channel 106, and typically can only change the audio volume level and the size of the video. The noise channel 106 is used to broadcast internal banners to direct traffic to register for upcoming tournaments, push video clips, commercials, and advertisements, and highlight news from the game. The noise channel 106 can be interrupted by announcements for the user, including, but not limited to: when a buddy, friend or other opponent comes online, an acceptance of a card trade, a request for an online challenge, announcements regarding the start of a TV show, news alerts, and contests. More specifically, in some embodiments, the noise channel 106 is customized based on the user's location (IP address or ZIP code) to check local listings and notify the user of television shows being broadcast to their current location, or other activities taking place near the user's current location. As shown in FIG. 1, the noise channel 106 is located within a defined area within the main display area 108. In the embodiment shown the noise channel 106 is in the bottom corner of the main display area 108 so as to be always visible, however, to avoid getting in the way of the main game play. The noise channel, however, is not limited to this positioning and can be moved to anywhere within the main gaming area 104 depending upon the game being implemented.

The noise channel 106, in one embodiment, includes one or more of the following features: a monitor area 602, a ticker area 604, a menu 608, a menu button 609, and a welcome button 610. In other embodiments, the noise channel includes only the monitor area 602 that is used to display content to a user and any of the other features are optional. The monitor area 602 displays visual content such as streaming video, Flash movies, videos, images, and promotions. The visual content
optionally includes links to display an additional pod, navigate to a new sector, or open a new webpage if clicked. In some embodiments, when a user's mouse is hovering over the monitor area 602, new content is prevented from appearing. This allows a user to pause the content if it is of interest to the user. If a video or movie is playing, the content will continue to play to the end, and then the video or movie will repeat from the beginning until the user's mouse is removed from the monitor area 602.

The ticker area 604 displays game news, ranking information, and other general news about the game or related information. In general, text is displayed in a crawling manner from right to left across the ticker area 604. In some embodiments, the ticket area 604 is controlled by arrows that rewind or advance the ticker at a faster rate than the standard crawl speed. When a user hovers their mouse over the ticker, the ticker will optionally stop. In some embodiments, the textual content is linked to an additional content such as a pod, a sector, or a webpage. When the textual content is clicked, the linked content is displayed.

The menu 608 is displayed by clicking the menu button 609. The menu 608 is hierarchy of organized content that has been displayed in the noise channel 106. The menu 608 allows a user to easily find any content that has been displayed in the noise channel 106 and replay that content in the noise channel 106. In some embodiments, additional details about the replayed content is also displayed for the user either in the noise channel 106 or in a separate pod (not shown). Clicking the menu button 609 again causes the menu 608 to disappear.

Referring now to FIG. 9, the welcome button 610 is used to activate the display of an additional pod, the welcome pod 702. The welcome pod 702 comprises different information sections, such as a news section 704, an alerts section 706, a messages section 708, and a challenges section 710. Within each section, relevant information is displayed, such as current news topics, a list of game alerts, a list of a user's messages, or a list of the user's challenges. Images and videos can also appear within each section. The user can click on any piece of information to view more content, such as a corresponding article, a private message, or a specific alert. Clicking the welcome button 610 again causes the welcome pod 702 to disappear.

Referring now to FIG. 10, in some embodiments, an alert window 802 appears adjacent to the noise channel 106 when a new alert is received. As shown in steps I though III, in some embodiments, the alert window 802 "slides" from behind
the noise channel 106 to be adjacent to the noise channel 106. Referring now to step III, the alert window 802 includes a title 804, a content area 806, an alert type area 808, and a minimize button 810. The content of the alert is displayed within the content area 806, and can be text, video, images and other information. In some embodiments, the content can be clicked to cause a new sector, pod, or window to appear with more information about the alert. The alert type area 808 displays the type of alert, such as whether the alert is regarding a challenge, a message, or an accepted trade. The alert window 802 remains visible for a predetermined period of time specified by the user or by the alert sender. Alternatively, the user can click the minimize button 810 to cause the alert window 802 to slide back behind the noise channel 106. Previously received alerts can be accessed through the menu 608.

While the invention herein disclosed has been described by means of specific embodiments and applications thereof, other modifications, variations, and arrangements of the present invention may be made in accordance with the above teachings other than as specifically described to practice the invention within the spirit and scope defined by the following claims.
CLAiMS

We claim:

1. A gaming platform comprising:
   a main gaming area;
   a plurality of sectors, each of which can be displayed within the main gaming area; and
   a sector navigator for moving game play from a first sector to a second sector.

2. The gaming platform of claim 1 wherein the sector navigator includes a plurality of links, wherein each link represents a sector within the game.

3. The gaming platform of claim 2 wherein the sector navigator is a visual representation of each of the plurality of sectors.

4. The gaming platform of claim 1 wherein the sector navigator includes a non-enabled link to a sector that is unavailable to a user.

5. The gaming platform of claim 1 wherein the sector navigator provides a visual indication of what is taking place within a sector.

6. A method of providing navigation for a game comprising:
   displaying a first sector of the game within a main gaming area;
   receiving an input at a sector navigator to move to another sector; and
   displaying a second sector of the game within the main gaming area in response to receiving the input.

7. The method of claim 6 wherein the sector navigator includes a plurality of links, wherein each link represents a sector within the game.
8. The method of claim 7 wherein the step of receiving an input includes selection of one of the plurality of links.

9. The method of claim 7 wherein the sector navigator is a visual representation of a plurality of sectors of the game.

10. The method of claim 6 further comprising enabling a non-enable link within the sector navigator to provide access to a sector that was unavailable to a user.

11. The method of claim 6 providing a visual indication within the sector navigator corresponding to an event that is taking place within a sector.

12. A gaming platform comprising:
   a main gaming area; and
   a noise channel displayed within a portion of the main gaming area, the noise channel providing content to a user that is related to a game being played within the main gaming area.

13. The gaming platform of claim 12 wherein the noise channel provides video clips, commercials, advertisements or news from the game.

14. The gaming platform of claim 12 wherein the noise channel includes a monitor area for displaying the content.

15. The gaming platform of claim 14 wherein the noise channel further includes a ticker and an alert window.

16. The gaming platform of claim 12 wherein the noise channel further includes a menu which allows for selection of content that has been or can be displayed in the noise channel.
17. The gaming platform of claim 12 wherein the noise channel cannot be turned off by a user.

18. A method of providing content to a user of a game comprising:
   providing gameplay within a main gaming area;
   providing a noise channel within a portion of the main gaming area; and
   providing content through the noise channel, the content corresponding to
   the gameplay within the main gaming area.

19. The gaming platform of claim 18 further comprising providing video clips, commercials, advertisements or news from the game through the noise channel.

20. The gaming platform of claim 18 further comprising providing an alert to a user through an alert window of the noise channel.

21. The gaming platform of claim 18 further comprising:
   receiving an input for selecting content from a menu of the noise channel;
   and
   providing the selected content in the noise channel in response to the
   received input.

22. The gaming platform of claim 18 wherein the noise channel cannot be turned off by a user.

23. A method of smart loading an online game comprising:
   loading a first sector, wherein gameplay starts in the first sector of the
   game;
   determining a second sector to load based upon a likelihood that a user
   will navigating to the second sector instead of a third sector upon leaving the first sector; and
loading the second sector of the game before loading the third sector of the game.

24. The method of claim 23 wherein the determining step is based upon the behavior of the user during prior gaming sessions.

25. The method of claim 23 wherein the determining step is based upon statistical behavior of other users of the online game.

26. The method of claim 23 further comprising loading a plurality of pods in sequential order based upon a determination of the most likely pods to be first needed by the user.

27. A gaming platform comprising:
   a main gaming area; and
   a first pod displayed within a portion of the main gaming area, wherein the pod is a user movable object that appears above the main gaming area.

28. The gaming platform of claim 27 wherein the first pod is specific to a certain sector of a game.

29. The gaming platform of claim 27 wherein the first pod is persistent within at least two sectors within the gaming area.

30. The gaming platform of claim 27 further comprising:
   a second pod displayed within a portion of the main gaming area;
   wherein the first pod is a parent pod and the second pod is a child pod.

31. The gaming platform of claim 30 wherein an event in the first pod causes a content update in the second pod.
32. The gaming platform of claim 30 wherein an event in the second pod causes a content update in the first pod.
FIG. 3

<table>
<thead>
<tr>
<th>Chaotic Community</th>
<th>Trade Cards</th>
<th>Card management</th>
</tr>
</thead>
<tbody>
<tr>
<td>Community main page; opens on my profile and buddy list.</td>
<td>View Market</td>
<td>Upload multiple cards</td>
</tr>
<tr>
<td>Manage Account Settings</td>
<td>Search / Filter Market</td>
<td>Build and Manage Decks</td>
</tr>
<tr>
<td>Manage Profile</td>
<td>Post Trades</td>
<td>Align and Manage Armies</td>
</tr>
<tr>
<td>Manage Avatar</td>
<td>Accept Trades</td>
<td>Manage and sort Portfolio</td>
</tr>
<tr>
<td>Manage Devices</td>
<td>View Trade History</td>
<td>View Statistics</td>
</tr>
<tr>
<td>Manage Notifications</td>
<td>Get Trade Card value help</td>
<td>All of Assets exchange; use</td>
</tr>
<tr>
<td>Manage Assets (points)</td>
<td>Set Notifications</td>
<td>Dracokens (and modifiers)</td>
</tr>
<tr>
<td>System defined IM presence</td>
<td></td>
<td>- potions</td>
</tr>
<tr>
<td>Give stuff away to friend presence</td>
<td></td>
<td>- downloads</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- skins</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ringtones</td>
</tr>
</tbody>
</table>
| | | One can give limited amount to another user (profile).

<table>
<thead>
<tr>
<th>Chaotic Forums</th>
<th>Welcome</th>
<th>The Arena rankings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message boards</td>
<td>Registration / Login</td>
<td>League Standing</td>
</tr>
<tr>
<td>- Give stuff away to friend presence (Members only)</td>
<td>Parent zone module</td>
<td>- Challenge (Issue / Accept)</td>
</tr>
<tr>
<td>- Chaotic Ambassador / moderator controls for those picked to be moderators. (Members only)</td>
<td>Retail products / stores</td>
<td>- Set Notifications</td>
</tr>
<tr>
<td></td>
<td>Game info / TV Show listings</td>
<td>Tournaments</td>
</tr>
<tr>
<td></td>
<td>News / Events / Newsletter</td>
<td>- Sign Up / Set Notifications</td>
</tr>
<tr>
<td></td>
<td>About the site (legal stuff)</td>
<td>- Schedule</td>
</tr>
<tr>
<td></td>
<td>This section is available to non-registered, so this is where a flash introduction to the game and site goes.</td>
<td>- Standing</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- System defined IM presence</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Portal to Perim</th>
<th>Gameplay tips</th>
<th>Arena battle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Background Story</td>
<td>Quick Start Instructions</td>
<td>Play the game</td>
</tr>
<tr>
<td>- Creatures</td>
<td>Rules / How to guide</td>
<td>- System defined IM presence</td>
</tr>
<tr>
<td>- Locations (Perim map)</td>
<td>Advanced tips and tricks</td>
<td>- Manage Play Options</td>
</tr>
<tr>
<td>TV show</td>
<td>Ask a Chaotic Star Player</td>
<td></td>
</tr>
<tr>
<td>(more detail than Welcome Sector)</td>
<td>FAQS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Deck Doctor</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Retail Locations</td>
<td></td>
</tr>
</tbody>
</table>

332 334 336
326 328 330
320 322 324
FIG. 4
User selects a menu function

Legend

User Steps (visual experience)

Parent requests child pods to open

Forced content update

Originator updating other pods

The main pod for the specific functions appears

Pod A request B and C to open

Pod B initiates content update in A and C

Pod C initiates content update in B

FIG. 7
FIG. 8
Welcome to Chaotic!

June 14th, 2006

alerts:

- Tournament March 15 is scheduled
- Tournament March 16 is scheduled
- Tournament March 17 is scheduled

You have 15 alerts

news:

- June 13, 2006: Gamespy gives chaotic online a thumbs up with a #4 rating
- June 10, 2006: The June newsletter hit email inboxes today

messages:

- New deck dude! - littleguy

You have 1 message

challenges:

- League Challenge - JoeSchmoe
- League Challenge - Uberdude
- Friendly Challenge - Equinox

You have 12 challenges

FIG. 9
INTERNATIONAL SEARCH REPORT

International application No
PCT/US2008/054997

A CLASSIFICATION OF SUBJECT MATTER
IPC(8) - A63F 9/00 (2008 04)
USPC- 463/30
According to International Patent Classification (IPC) or to both national classification and IPC.

B FIELDS SEARCHED
Minimum documentation searched (classification system followed by classification symbols)
IPC(8) - A63F 9/00 (2008 04)
USPC- 463/30, 245/651, 711/17, 118, 137

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
Patbase, Google Scholar

C DOCUMENTS CONSIDERED TO BE RELEVANT

<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>17, 22, 29-32</td>
</tr>
<tr>
<td>Y</td>
<td>US 6,155,926 A (MIYAMOTO et al) 05 December 2000 (05 12 2000) entire document</td>
<td>23-26</td>
</tr>
</tbody>
</table>

Further documents are listed in the continuation of Box C

Date of the actual completion of the international search
25 June 2008

Date of mailing of the international search report
08.11.2008

Name and mailing address of the ISA/AJS
Mail Stop PCT, Attn ISA/US, Commissioner for Patents
P.O. Box 1450, Alexandria, Virginia 22313-1450
Facsimile No 571-273-3201

Authorized officer
Blame R Copenheaver
PCT Helpdesk 571 272-6300
PCT DSP 571 272 7774