

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
26 May 2006 (26.05.2006)

PCT

(10) International Publication Number
WO 2006/055840 A2

(51) International Patent Classification:

A63F 9/24 (2006.01)

(21) International Application Number:

PCT/US2005/041965

(22) International Filing Date:

18 November 2005 (18.11.2005)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

60/629,226 18 November 2004 (18.11.2004) US

(71) Applicant (for all designated States except US): **TURBO SQUID, INC.** [US/US]; 400 Poydras Street, Suite 1500, New Orleans, LA 70130 (US).

(72) Inventors: **SALAS, Daniel, Joseph**; 1014 Jib Drive, Apt. 102, Orlando, FL 32825 (US). **WISDOM, Matthew, Morgan**; 1115 Henry Clay Ave., New Orleans, LA 70118 (US). **SOSLOW, Timothy, Lowell**; 624 Webster Street, New Orleans, LA 70118 (US). **REILY, James, Harrington**; 219 Ridgewood Drive, Metairie, LA 70005 (US).

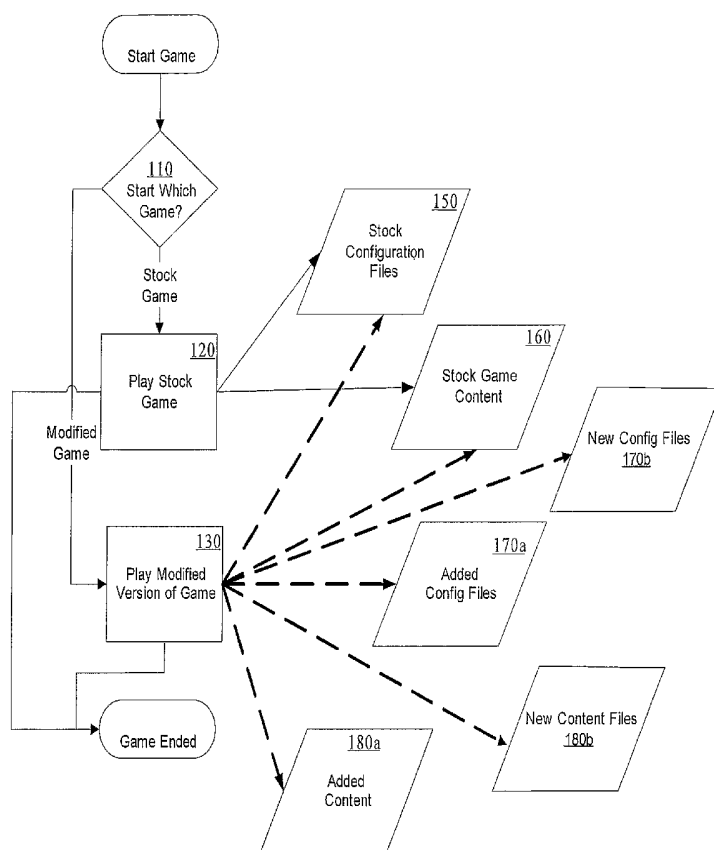
PHILLIPS, Christopher, Patrick; 919 Jefferson Avenue, New Orleans, LA 70115 (US). **LEE, Michael, S.**; 4536 Ross Lanier Lane, Kissimmee, FL 34758 (US). **WISDOM, Andrew, Bell**; 1919 State Street, New Orleans, LA 70118 (US). **GAFFNEY, Brian, John**; 341 Cherry Hills Court, Thousand Oaks, CA 91320 (US). **HALES, Matthew, Wayne**; 138 Paraiso Drive, Danville, CA 94526 (US).

(74) Agent: **MARCOU, George, T.**; Kilpatrick Stockton LLP, 607 14th Street, N.w., Suite 900, Washington, DC 20005 (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, KN, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, LY, 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.

[Continued on next page]

(54) Title: METHOD OF DYNAMICALLY MODIFYING A COMPUTER GAME



(57) Abstract: An application is described for facilitating the modification of an underlying computer game by players and developers; the development and testing of game modifications via a publishing infrastructure; the sharing and playing of game modifications and the marketing of available game modifications. Users subscribe to the application or are charge a fee to access individual content and partake in these game modification processes across a network infrastructure.

WO 2006/055840 A2



(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, LV, MC, NL, PL, PT, RO, 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.

METHOD OF DYNAMICALLY MODIFYING A COMPUTER GAME

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] The present application claims the benefit of similarly titled United States provisional patent application no. 60/629,226 filed November 18, 2004, which is incorporated herein by reference in its entirety.

BACKGROUND OF THE INVENTION

Field of the Invention

[0002] The invention is directed generally to computer game modifications. More particularly, the invention describes a system and method for allowing developers and/or retailers to develop and offer game modifications to consumers and for consumers to play, create and share game modifications through a network infrastructure.

Description of the Related Art

[0003] Referred to as game modification, computer and video games have historically been modified to allow for different play of the stock game. For example, a computer game originally coded for play with World War II weaponry, could be changed, i.e., files added, in order to allow the user to utilize modern weaponry, e.g., M16 machine gun, in the World War II setting and situations.

[0004] Creating a game modification requires technical knowledge of the inner workings of the game being modified, and this skill has typically been beyond the skill of the ordinary user, e.g., consumers, who play such games.

[0005] Accordingly, there is a need in the art for a method and system which allows the consumer, i.e., buyers and players of video games, to create (and thus become modifiers), use and share game modifications. Further, there is a need in the art for a game modification network, through which modifications may be developed, sold, and shared by modifiers, retailers, and users.

SUMMARY OF THE INVENTION

[0006] In a first exemplary embodiment of the present invention, a method for modifying the content of a game capable of being played on a processor is described. The method includes: providing access to a modification application to implement a modification to a game; providing the ability for a user to identify a game to modify to the modification application; providing access to content files for use in modifying the identified game; accepting user selection of at least one content file for use in modifying the content of the selected game; and creating at least one modification configuration file for implementing the data in the at least one content file during play of the selected game.

[0007] In a second exemplary embodiment of the invention, a method for sharing a modified version of an underlying game is described. The method includes: accepting at a host an invitation from a first player to play a modified version of the underlying game

for forwarding to at least a second player; inviting a second player to play the modified version of the underlying game; receiving an acceptance from the second player to play the modified version of the underlying game; determining if the second player has access to a modification application for sharing the modified version; if the second player does have access to the modification application, prompting the second player to log-in to modification application; if the second player does not have access to the modification application, prompting the second player to request access to the modification application, granting access to the modification application if requirements for access are met and prompting the second player to log-in to the modification application; determining if the second player has the underlying version of the game installed on the second player's processor; if the second player does have the underlying game, installing the modified version of the underlying game on the second player's processor; if the second player does not have the underlying game, inviting the second player to obtain the underlying game, determining when the underlying game has been obtained and installing the modified version of the underlying game on the second player's processor.

[0008] In a third exemplary embodiment of the present invention, a method for facilitating the purchase of game modifications is described. The method includes: detecting when a user has accessed a modification application, the modification application containing at least one subapplication for each underlying game for modifying the underlying game; scanning the processor of the user to determine if each underlying game for which the modification application has at least one subapplication are located on the user's processor; if the user's processor contains all underlying games

for which the modification application has at least one subapplication, determining if the user has current modifications to the underlying games included in the at least one subapplication for each underlying game; if the user's processor does not have current modifications for each of the underlying games, offering the user access to the at least one subapplication for each of the underlying games containing current modifications; and if the user's processor does not contain all underlying games for which the modification application has at least one subapplication, determining which underlying games the user is missing, offering the user access to the underlying games and subsequently offering the user access to the at least one subapplication containing current modifications for each of the underlying games.

[0009] In a fourth exemplary embodiment of the present invention, a method for facilitating the design and publication of modified game content is described. The method includes: providing at least one processor including at least one database for collecting and storing data related to the development of modified game content for a previously released underlying game; providing a search interface to the at least one processor including at least one database wherein a user can enter search terms to identify personnel and modification projects; storing files related to the development of a game content modification; publishing a developed game content modification; and providing a platform for facilitating at least one test of the developed game content modification.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] The present invention may be better understood, and its numerous features and advantages made apparent to those skilled in the art by referencing the accompanying drawings.

[0011] **Figure 1** is a flowchart of the basic process of modifying a computer game;

[0012] **Figure 2** is a flowchart of the process employed by the invention when used by a consumer;

[0013] **Figure 3** is a flowchart of the process of sharing game modifications with other users;

[0014] **Figure 4** is a flowchart of the process of auto-detecting installed games on the client processor and using this for marketing purposes; and

[0015] **Figure 5** is a flowchart of the process used by the publishing infrastructure.

DETAILED DESCRIPTION OF THE INVENTION

[0016] As one skilled in the art will realize from a reading of the embodiments described herein, the methods described below, either in their entirety or individual steps and groupings of steps thereof, may be practiced on any number or various types of processors, processing systems, servers and/or networks, including open, closed, proprietary, public networks and the like. This includes stand-alone processors, i.e., not connected to the Internet or other open network, and networked processors connected to the Internet or other open network, through hard-wire or wireless capabilities. The processors may be any device capable of running a video game, including, but not limited to, a personal computer ("PC") having hard drive or other form of data storage such as

flash RAM and/or ROM or game console, e.g., Microsoft's Xbox®, Nintendo GameCube®, Sony PlayStation®, etc.

[0017] **Figure 1** is a flowchart that details the normal process of game playing versus the process of playing a modified game. When the user goes to start the game, the user must choose whether to play the stock game, or to play a modified version of the game. The user typically makes this choice **110** either prior to launching the game by adding special commands to the initial execution of the game, or after the game has started through a menu of recognized modifications offered by the game.

[0018] In the case where the user is playing the stock game **120**, the game will load the stock content files **160** and the stock configuration files **150** to play the game as originally sold by the publisher. Examples of content files **160** and **180a** and **180b** are 3d models, textures, sounds, and image files. Examples of configuration files **150** and **170a** and **170b** are .ini and text files.

[0019] In the case where the user would like to play a modified version of the game, the game will typically load not only the stock content files **160** and the stock configuration files **150**, but also any modified configuration and/or content files **170a**, **170b**, **180a**, **180b**. These modified files could be added content files **180a** and added configuration files **170a** which are in addition to the stock content and stock configuration files **150**, **160**. Alternatively, the stock content files **160** and the stock configuration files **150** may be superseded by new configuration files **170b** and/or new content files **180b**.

[0020] **Figure 2** is a flowchart of the process employed by the invention when used by a consumer for creating a new modification. The modification application (hereafter “MA”) may reside on the consumer’s processor, e.g., PC, phone, PDA, or console, (hereafter “user processor”) may reside partially on the consumer’s processor and partially on at least one host server or, in a preferred embodiment, the MA resides on at least one host server. In order to access the MA on the host server, the consumer logs in through, for example, a Web interface accessible via the Internet or a proprietary interface accessible through a console connected through proprietary or public lines, e.g., phone lines. In a preferred embodiment, the consumer must be a subscriber in order to obtain a valid login identifier (“ID”) and password. Alternatively, access may be free and no login ID and password required or free with login ID and password required. Further still, the consumer may access for free, but be required to provide payment information in order to access and use individual game content items and/or subapplications as described below.

[0021] The MA may be accessed independently of game-play **190** or during game-play. In the latter instance, the game may be paused and a menu provided, wherein one of the menu choices is “Create Modification?” **192** If the consumer accesses the MA outside of game play, the MA queries the consumer to identify for which game(s) the consumer wishes to create a modification **194**. Once the user has launched the MA, the user is asked to select the content **200** to add to the game modification that he or she is creating. The MA maintains a library of content available for each game. Some examples of game

content are a game map and level, gameplay styles for a map or level, a tank, a race car, a spaceship, a speedboat, a fighter jet, a handgun, a laser rifle, swords, grenades, crossbows, fireballs, a humvee vehicle, an animal, human and non-human characters such as gangsters, aliens, cowboys, zombies, marines, dwarves, werewolves, business people, and an infantry soldier. The user is then asked to make any desired changes **205** to the various values and settings for this content. In the case of a tank, this could include the speed of movement of the tank in miles per hour, the cost to construct in dollars, and whether the tank uses an artillery canon or a flamethrower as its main weapon. One skilled in the art recognizes that the types and amount of content, values and settings is unlimited. The examples provided herein are not intended to be limiting. The user may make changes to multiple content items **210**.

[0022] Once the user is satisfied with the content and the settings of that content that have been selected for the game modification, the user has the option of making changes **215** to the stock content files. Stock game content is content that came with the game at the time of purchase. For example, a game about World War II might come with a U-Boat as one of the submarines that a user might pilot. The user can change **220** the settings that govern the U-Boat within the game, for instance, to allow it to go into deeper water than the designers of the original game allowed for. In the example of the U-Boat as game content, the stock content files might include a 3d model of the U-Boat, a rusty texture file that is painted on the model, and the sound file of a propeller underwater. The present invention implements the content changes from steps **200-220** with, referring

to **Figure 1**, additional content files **180a** which are in addition to the stock content files **160** or via new content files **180b** which supersede the stock content files **160**.

[0023] When the user no longer desires to change the configuration settings relating to the stock content **160**, the user then may press the 'play modification' button **225** in the MA. This action triggers a series of processes. Firstly, the server will interpret **230** the user selections and create modified (mod) configuration files **170a** or **170b** that will cause the game to include the desired content changes of the user.

[0024] An alternate embodiment for creating the modified configuration files **170a** and **170b** is for the server to create a list of changes to the stock configuration files **150** that are stored within the game on the user's processor. The MA would then create the set of additional configuration files **170a**, **170b** by processing the change list on the stock configuration files **150** that were installed with the game.

[0025] The MA will download **235** the configuration files **150** and **170a** or **170b** from the server. Following this, the MA will check **240** with the server if any additional content files **180a** are required in order for the game modification to function. In the case where additional content files **180a** are required on the client processor, the MA will download **245** all such additional content files.

[0026] The preferred embodiment for the game modification is to use a system where all or as many as possible of the modified content files **180a**, **180b** are installed on the user

processor at the same time as either the MA is installed and/or accessed and subsequently updated each time the user accesses the MA to modify and/or play the game. Installation could be accomplished on-line over the network connection between the user processor and the host server or via CD-ROM off-line at the user processor. This dramatically reduces the download time required to create each subsequent modification of the game by the user. Alternatively, the content files are installed on the user processor as selected by the user.

[0027] Once all of the appropriate content files **180a** or **180b** and configuration files **170a** or **170b** are on the user processor, the MA installs **250** those files into the appropriate user file folders in whatever format is required by the selected game. Alternatively, the install may be accomplished via a web browser plug-in. The MA then launches **255** the game in the appropriate way such that the user can play the modified version of the game that he or she created.

[0028] The user at any time will then make the decision **260** to either make a new modification to the game or to exit the MA, or to use some other functionality included in the MA.

[0029] An alternate embodiment of this invention is to remove the client/server aspect and to create the whole invention as a self-sustained application on the user's processor wherein no Internet or other networking connectivity is required. In this embodiment, the

user would be limited in the types of modifications that may be created as the MA would be static.

[0030] **Figure 3** is a flowchart of the process of sharing game modifications with other users. At some time after starting the MA, the user may invite another user **300**, typically a friend, to play any game modification that the user has created or played himself/herself. The user provides the MA with one or more email addresses **305** and a written note to the intended invitees. The written note could be a standard note supplied by the MA, including proposed date and time for playing modification.

[0031] The MA conveys this list to the host server, and the host server sends **310** an invitation to each of the friends. This invitation can be sent to the friend's processor, mobile phone, personal digital assistant ("PDA"), console or other device. The message to play the game modification selected by the user **300** is sent via email, sms, or other electronic messaging protocol. When the friend receives the invitation **315**, the friend can either ignore the invitation **320**, or can begin the process of setting up his or her processor **325** to play the game modification. If the friend already has the MA **325**, the friend can immediately access the MA **335**. However, if not, the friend can purchase or sign-up to use the MA **330** so that he/she may continue the process to use the game modification **300**.

[0032] When the friend starts the MA **335**, it will perform a check **337** to verify that the user has the appropriate underlying computer game required to play the game

modification **300**. If the friend does not have the appropriate computer game, then he/she will have to purchase/access **339** the required game. The friend could purchase the game in any of a variety of ways, including purchasing off-line from a brick and mortar vendor which would delay the game play or, the MA could provide an on-line vendor cite where the friend could purchase and download in real-time. Alternatively still, the MA could act as a real-time vendor of the underlying game.

[0033] At this point, the friend will have both the required access to the MA and the required computer game. The friend can then accept the invitation **340** to play the game modification. The friend's MA will download **345** the game modification from the invitation. If the user and the friend do not intend to play the game modification together on an online platform, e.g., server **347**, then the friend finishes this process and plays the game modification alone **349**. However, if they do intend to play the game modification together, the server will prepare a game server **350** to play the game modification online automatically in real-time or near real-time. The processors of both the user and the friend will indicate **355** that the server is ready to play. Both the user and friend will execute whatever commands are appropriate to play the game modification online **360** for that particular game. This will complete the process of inviting a friend to play a game modification.

[0034] **Figure 4** is a flowchart of the process of auto-detecting installed games on the user processor and using this information for marketing purposes, i.e., in order to offer game modifications or the ability to modify games via the MA. This embodiment

assumes that the user has access rights to the MA. When the user launches the MA, the MA automatically checks 400 for all supported games to see if they are installed on the client processor. If the MA detects that supported games are installed 402, the MA offers 405 the user to purchase and/or access and download the appropriate subapplication of the MA for each installed game that is supported but that is not already installed. The subapplication contains the library of modified content for that particular game.

[0035] For example, if there is a subapplication of the MA that supports a hypothetical game called “Bungee Jump”, then the MA will check 400 to see if the user has the Bungee Jump game installed on his/her processor. If the user does, in fact, have that game installed, then the client will offer the user 405 to purchase access to the subapplication of the MA that supports the game Bungee Jump. In the case where the user does want to buy access to the subapplication of the MA 410 (like for Bungee Jump), the MA (or a website) will process the transaction 415 and the MA will download and install 420 the newly purchased subapplication.

[0036] If there are any games that are supported by the MA 422, but not installed on the client’s processor, the client will offer the user to purchase each of those games 425. So for example, say the user does not have the game Bungee Jump installed but the MA does support that game. In this case the user will be offered to purchase 425 Bungee Jump. Once Bungee Jump is installed, the client will detect it 402 and offer to the user to purchase the subapplication containing modifications to Bungee Jump 405 as detailed in the beginning of this process. If there are no such games that are supported by the

invention but are not installed, then the user will exit this process and continue with normal client activity. However, if the user does want to purchase the game 430, the user will be able to purchase the game 435 within either the MA or through a website provided by the MA. Alternatively, the user may purchase the game from a brick and mortar vendor. After this step, the user will exit this process and continue with normal client activity.

[0037] **Figure 5** details a process according to an embodiment of the invention for utilizing a publishing infrastructure (hereafter “infrastructure”) in order to create, test and market game modifications. The publishing infrastructure is intended to facilitate and simplify the process of producing game modifications for the game modification developer. The preferred embodiment of the method surrounding the publishing infrastructure is to allow free usage of the publishing infrastructure in return for exclusive distribution rights of the products created therein. For example, if a person is a developer and needs the use of a collaboration and publishing infrastructure for his or her team to create a game modification with 5 Army tanks, the person can use the publishing infrastructure with no hard cost associated with it. Once the game modification with the army tanks is created, however, it may only be distributed through the MA described herein with respect to **Figures 1-4**. Alternate embodiments are to charge a flat or monthly fee for the use of the publishing infrastructure.

[0038] The infrastructure is comprised of applications for facilitating the creation, testing and distribution of game modifications. One skilled in the art recognizes that such an

infrastructure may be implemented through any number of client/server and networking configurations and architectures. In a preferred embodiment, the developer accesses the infrastructure from a web browser or client application. Typically, at the point of starting use of the system the developer will create an online profile **500** of him/herself that details contact information, skills, known software applications, and any other required personal information. This can be viewed online by other developers, particularly when the developer is requesting to join a project team **516**, or when a developer has created his or her own project **504** and is searching **508** for other developers to join the team.

[0039] Once the developer profile is created **500**, if the developer desires **502** the developer can create a project **504** on the system. Such a project contains a profile that describes the intended output of the project, a file storage system **520**, task assignment system **524**, bug tracking system **526**, messaging system **527**, and other functionality that can be used to aid in the production of the project.

[0040] In some cases, a developer who has created a project may want to add team members **506** to the project as additional labor. The developer will search **508** for team members based on the skill sets, experience levels, and other criteria contained within the developer profile. The developer will select appropriate candidates for joining his or her project and invite them **510** to join the team. The invitee may accept or decline this invitation.

[0041] To work on a project, a developer must either create his or her own project **502**, or join a project created by another developer. The process of joining another developer's project is firstly to search **514** for desired projects based on the criteria in the project profile. For any projects that the developer deems worthy of joining, the developer will request to join the project **516**. The creator or otherwise authorized team member of such projects will accept or decline this request. If the request is accepted, the developer will become a team member for that project and can work **518** on that project.

[0042] The process of working on a project **518** includes using many collaboration and work-saving features that operate over the Internet. The first such feature is a file system **520**. This is a filesystem that can be accessed by team members through a web browser, ftp (file transfer protocol) client, or similar client application. Team members from a single location or widespread geographic areas are all able to write, read, move, delete, and otherwise manipulate files and folders in a shared environment. It also allows for file versioning, a process whereby the servers that host the publishing infrastructure maintain all past versions of a file in case the developer would like to view an older version of any given file.

[0043] Another feature supporting project work is a task assignment system **524**. This allows developers to create tasks for themselves, or other team members working on projects that the user is a member of. The tasks are stored in the database, and can be linked to files within the system.

[0044] Another feature supporting project work is the defect tracking system 526. This system allows a developer to create database entries that detail any “bugs” or defects in the project. These defect entries may be assigned to the user, or to other team members within the system.

[0045] Another feature supporting project work is a messaging system 527. This is an interface that allows team members to communicate in writing by creating emails, instant messages, or notes that are given to other team members. These communications can be linked to other files within the system.

[0046] The collaboration system also features a shared file pool 529 for use with other teams. This system allows various teams to both use from and contribute to a central library of files for a project. For example, one team might create and complete 528 a 3d model of a tank that another team can use. Any team that uses a file contributed to the shared pool must compensate the team by paying a share of royalties from revenue generated by the project that uses the shared file. An alternate embodiment of the file sharing pool compensation is for teams that contribute to the shared pool to receive ownership shares (similar to stock shares) that receive a percentage of the total revenue generated by the invention. Another alternate embodiment is non-monetary system whereby a team that wishes to use a file must contribute a file of equal quality to the shared pool for others to use.

[0047] During the normal course of work 518, the user may complete one or more aspects of the project 528 that are worthy of being published 532 as part of a game modification. For example, the user may complete a single army tank and wish to publish it for use in the MA as a game modification. While the project might be to create 5 such tanks, this one tank is ready to publish 532.

[0048] Alternatively, the developer or team may complete an entire game modification 530 that does not use the MA for delivery to the end user. This is a standalone game modification and typically includes a software installer to make use easier for the game-playing user.

[0049] The process of publishing content 532, such that users of the MA can play it as a game modification, is facilitated by a special publishing interface. This publishing interface allows the developer to publish his or her content for use as a game modification without requiring underlying knowledge of the manipulation of game configuration files. For example, the developer would publish a tank by selecting the appropriate 3d model, texture, sound and other files that comprise the tank, and select any values such as cost by entering a number into a text input field. At the appropriate time, the system will generate the required game configuration files such that the published tank is included within the game.

[0050] Further to the processes of publishing content 530, 532, the infrastructure uses a compatibility layer which allows modified content created independently by numerous

developers or development teams for a particular game or type of game to be interoperable and combined together to form new game elements. The compilation of modified content to form a game modification can occur at any point, all the way up to the point of playing the game, without any specification of what the content is or will be. For example, according to steps 500-528 of **Figure 5**, a first developer might create a soldier character for an American Revolution war game, and second developer might create a musket-style gun with a bayonet, and a third developer creates a game "map" that is modeled on the Battle of Trenton which serves as the environment for the users to play within. Because this content is published according to the requirements of the compatibility layer, a user, i.e., the game player, using the MA, can add each item of content all together into one game that he/she can play. This is accomplished by using a neutral compatibility layer where the file references for the individual content items are not named according to the default scheme required by the game as per the prior art, and thus can later be changed so that the file references for the content items point at the appropriate reference locations for the new items of content.

[0051] For example, in an implementation of the compatibility layer, modified game maps may contain references to game content that is specified only to the extent that they could be used to represent content during game play, but unless selected, do not appear during game play. These references are "null" references or arbitrary strings such as "ReplaceMe" that can be located later by the MA. For instance, the Battle of Trenton game map mentioned above could contain ten (10) "null" rifle references, and ten (10) "null" sword references which are placeholders within the map. The infrastructure

allows the game playing user, as desired, to place from zero (0) to ten (10) of the various rifles and from zero (0) to ten (10) of the various swords at the pre-specified locations to create the modified game. The unused references do not show up in the game. This allows the user to dynamically add varying amounts of game content.

[0052] In another example of the implementation of the compatibility layer, in a game map which includes weapon spawn s, i.e., places within the map where weapons appear during game play, the developer places “null” weapon spawns as placeholders in the map that can be edited later by the MA to contain actual weapon spawns for in-game play.. The actual weapon spawn to be created is specified outside of the map in, e.g., a script file, that the “null” weapon spawn is directed to during the creation of the map. With the script file outside of the map it can be changed without affecting the map itself. Additionally, with “null” weapon spawn scripts, the game map is open to a wide variety of weapon play types that can be created at will by the game player using the MA, for example, a battle of many types of modern rocket propelled grenades but set at the Battle of Trenton during the American Revolutionary war.

[0053] Similarly, “null” weapons are placeholders for weapons in a proposed game modification. The “null” weapons allow additional weapons to be added to the game without the need for the game to be recompiled. The “null” weapons are multiple copies of certain classes of weapons, and they can be made to be completely unique weapons by changing a few parameters, and giving them a new model. With the “null” weapons it is

possible to create a wide range of game play, by being able to provide a large selection of many different kinds of weapons.

[0054] While the preferred embodiment of the infrastructure system and method requires users to format developed content according to the specifications of the compatibility layer, the infrastructure may include the ability to auto-detect content references that can be retro-fitted to use the compatibility layer, i.e., where content has been specified, the infrastructure is able to detect those content references and retro-fit to include a “null” reference in order to allow for configuration and inclusion within modification games in the MA.

[0055] The process of providing quality control for the published content is a three-phase beta testing process. In phase one 534, the developer or team is required to test the published file within the MA as if it were generally available to all users. However; only the project team has access to this published content in phase. In the example of publishing an army tank, once the tank has been published, the developer (or another team member) would direct the MA to include the tank in a game modification. The developer would download and play the game modification to verify that there were no defects in the tank. If there are any defects, then the developer will fix the files and re-publish them. If there are no defects, the developer will certify that the tank has passed phase one.

[0056] Once the developer or other team member has tested the content he or she may certify that it functioned successfully with no defects by clicking a button within the interface. This will complete phase one 534.

[0057] Phase two 536 of the beta testing process is when the content is made available in the MA for only a select group of developers outside of the project team that test new content. The developers will test the content through the MA as if it were generally available to all users, even though only they and the project team are allowed to see the content. Once a certain number of phase two testers certify that the content is defect free and of high artistic quality, the content will move on to phase three 538. If a tester says that the content has a defect, the developer will have to correct the defect and resubmit to stage two before it can be published. Additionally, if a preponderance of stage two testers say that the content is of low artistic quality, the developer will have to improve the artistic quality and resubmit to stage two testing before becoming eligible for stage three testing.

[0058] In the example of the tank in phase two 536, once phase one 534 is completed the tank will become available to a larger group of testers that are not part of the project team. Each person that tests the tank will record the results of their testing. Say that the tank has a flaw in the turret. The tester will report the flaw. If the defect is both serious and verified by authorized users of the publishing system, this will cause the developer to get a notification that the content has failed stage two testing. The developer must repair the defect and begin the process again at phase one 534.

[0059] Phase three 538 of the beta testing process is when the owners of the publishing infrastructure system itself approve the content. There are two approvals required here. The first is a test for defects by employees of the owners of the infrastructure in much the same way of testing as phase one 534 and phase two 536. Any substantial defects found must be repaired by the developer and the process begun again 534. The second approval is for whether the quality and type of content is appropriate for the system itself. Take the example of a tank. It may successfully make it to phase three and pass the testing for defects. However, there may already be a similar tank published. In this case, the tank would be refused publishing into the modification application.

[0060] If the content is approved, it is added to the MA library, described above with reference to the processes shown in **Figures 1-4**, and made available for use by subscribers to the MA 540. The processes described herein facilitate mixing and matching of game modifications across content types, game levels (environments within an individual game), game titles (individual games such as Half Life or Madden 2006), and processors (such as a personal computer, Xbox, Playstation). A further feature of an exemplary embodiment of the present invention is the availability and display of the history of where each available game modification has been used. For example, a user may like a particular weapon modification. This user can view the history of which games and where this weapon modification has been used and play or modify those games further. The ability to view history of modification use may inspire new and different modifications across different games.

[0061] The embodiments set forth herein are intended to be exemplary and in no way limiting. One skilled in the art can appreciate the many variations to portions of these embodiments and the resulting alternative embodiments which fall within the scope of the invention as set forth herein. Further, one skilled in the art understands the many hardware and software configurations that may be used to implement the embodiments that are within the scope of this invention without undue experimentation.

Claims:

1. A method for modifying the content of a game capable of being played on a processor comprising:
 - providing access to a modification application to implement a modification to a game;
 - providing ability for a user to identify a game to modify to the modification application;
 - providing access to content files for use in modifying the identified game;
 - accepting user selection of at least one content file for use in modifying the content of the selected game; and
 - creating at least one modification configuration file for implementing the data in the at least one content file during play of the selected game.
2. The method according to claim 1, wherein the accessible content files include stock content files for the selected game and added content files selectable from a library of content files.
3. The method according to claim 2, wherein accepting user selection of at least one content file for modifying the content of the selected game includes:
 - accepting user selection of at least one content parameter, selected from a choice of content parameters.

4. The method according to claim 3, wherein the content parameters include new content and selectable changes to existing content.
5. The method according to claim 2,
wherein the at least one modification configuration file is one of either an added configuration file or a new configuration file;
further wherein added configuration files implement data in the added content files and new configuration files implement data in stock content files, including changes to stock content.
6. The method according to claim 1, further comprising installing a game modification at the processor including the at least one modification configuration file.
7. The method according to claim 1, wherein the modification application resides solely on at least one server separate from the processor.
8. The method according to claim 1, wherein providing access to the modification application comprises:
considering user identification information entered at a login screen provided on a display associated with the processor.

9. The method according to claim 1, wherein the game to modify is determined by the modification application when a user accesses the modification application during play of the game.
10. The method according to claim 1, the game to modify is determined by the modification application when the game is affirmatively selected by a user outside of play of the game.
11. The method according to claim 1, wherein the modification application is accessible via the Internet and a user is charged for use of the modification application either by (i) requiring a subscription to access the modification application or (ii) requiring a fee for use of individual content files.
12. The method according to claim 1, wherein the modification application resides partially on at least one server separate from the processor and partially on the processor.
13. The method according to claim 1, wherein the modification application resides solely on the processor.
14. The method according to claim 1, wherein the processor is selected from the group consisting of a personal computer, a game console, and a personal digital assistant.

15. A method for sharing a modified version of an underlying game comprising:
- accepting at a host an invitation from a first player to play a modified version of the underlying game for forwarding to at least a second player;
 - inviting a second player to play the modified version of the underlying game;
 - receiving an acceptance from the second player to play the modified version of the underlying game;
 - determining if the second player has access to a modification application for sharing the modified version;
 - if the second player does have access to the modification application, prompting the second player to login to the modification application;
 - if the second player does not have access to the modification application, prompting the second player to request access to the modification application, granting access to the modification application if requirements for access are met and prompting the second player to login to the modification application;
 - determining if the second player has the underlying version of the game installed on the second player's processor;
 - if the second player does have the underlying game, installing the modified version of the underlying game on the second player's processor;
 - if the second player does not have the underlying game, inviting the second player to obtain the underlying game, determining when the underlying game has been obtained and installing the modified version of the underlying game on the second player's processor.

16. The method according to claim 15, further comprising providing a platform on which the first and second player can play the modified game.

17. The method according to claim 15, wherein requirements for access to the modification application include purchasing access to the modification application through an on-line payment process.

18. The method according to claim 15, wherein the format of the invitation is an electronic message.

19. A method for facilitating the purchase of game modifications comprising:

detecting when a user has accessed a modification application, the modification application containing at least one subapplication for each underlying game for modifying the underlying game;

scanning the processor of the user to determine if each underlying game for which the modification application has at least one subapplication are located on the user's processor;

if the user's processor contains all underlying games for which the modification application has at least one subapplication, determining if the user has current modifications to the underlying games included in the at least one subapplication for each underlying game;

if the user's processor does not have current modifications for each of the underlying games, offering the user access to the at least one subapplication for each of the underlying games containing current modifications; and

if the user's processor does not contain all underlying games for which the modification application has at least one subapplication, determining which underlying games the user is missing, offering the user access to the underlying games and subsequently offering the user access to the at least one subapplication containing current modifications for each of the underlying games.

20. The method according to claim 19, wherein offering the user access to the underlying games includes an on-line purchase request.

21. The method according to claim 20, wherein offering the user access to the underlying games includes directing the user to the website of an on-line vendor of the underlying games.

22. The method according to claim 19, wherein offering the user access to the at least one subapplication for each of the underlying games includes an on-line purchase request.

23. The method according to claim 22, further comprising providing access to the at least one subapplication for each of the underlying games in response to completion of the on-line purchase.

24. A method for facilitating the design and publication of modified game content comprising:

providing at least one processor including at least one database for collecting and storing data related to the development of modified game content for a previously released underlying game;

providing a search interface to the at least one processor including at least one database wherein a user can enter search terms to identify personnel and modification projects;

storing files related to the development of a game content modification; and

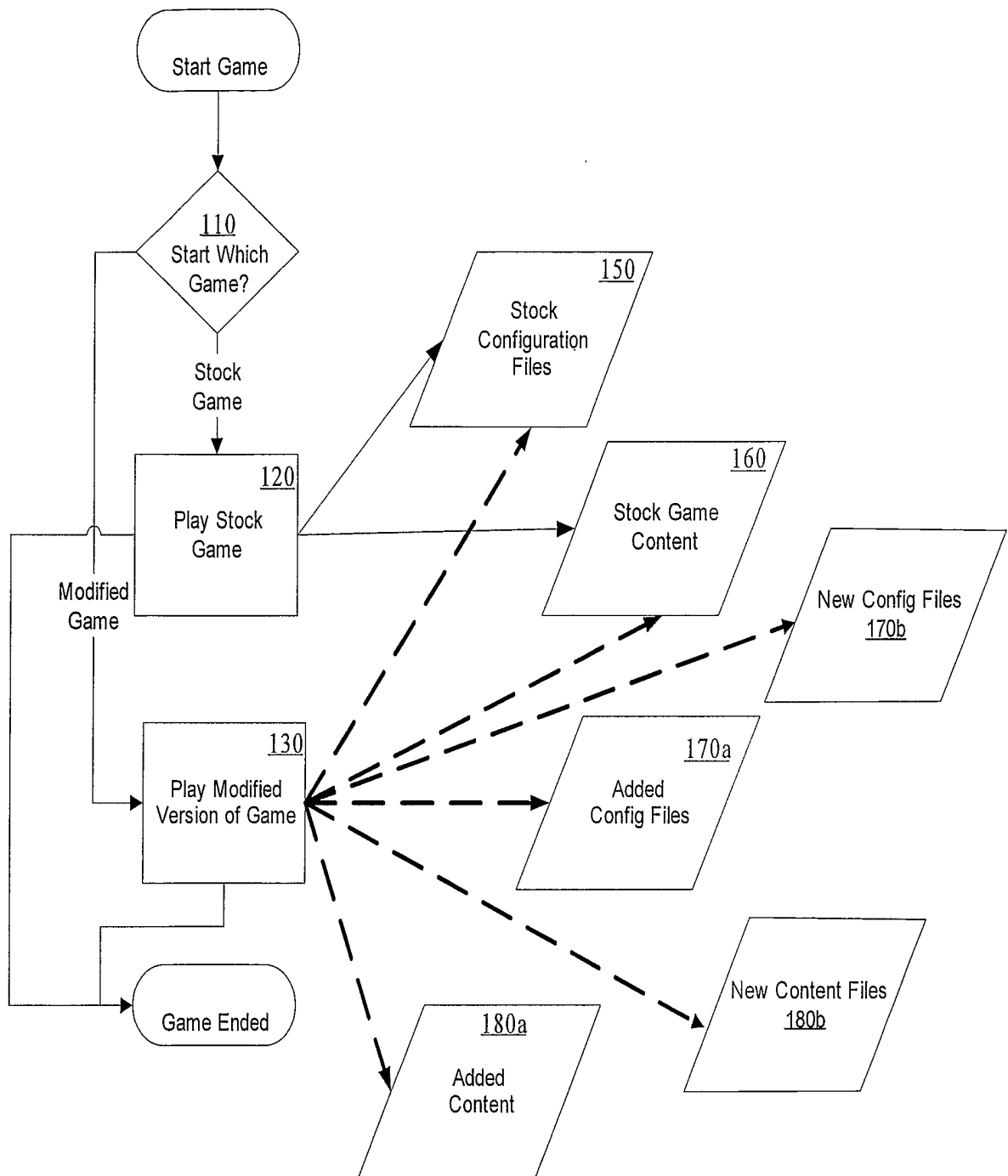
publishing a developed game content modification.

25. The method according to claim 24, wherein the data includes developer profile data, modification project data, proposed content modification data and modification testing data.

26. The method according to claim 24, wherein the files include at least one of the following group consisting of game content modification tasks, identified game content modification defects, developer messages regarding the game content modification.

27. The method according to claim 24, wherein the platform facilitates one or more tests of the developed game content.

28. The method according to claim 24, wherein the search terms include types of content, developer name, and name of underlying game.
29. The method according to claim 24, wherein the at least one processor including at least one database and the platform are accessible via the Internet.
30. The method according to claim 24, wherein access to the at least one processor including at least one database and the platform require a subscription.
31. The method according to claim 24, wherein a fee is collected prior to allowing access to individual game content modifications.
32. The method according to claim 24, wherein the developed game content modification is published in accordance with compatibility specifications to allow use of the developed game content modification across multiple games titles, game levels and processors.

**FIGURE 1**

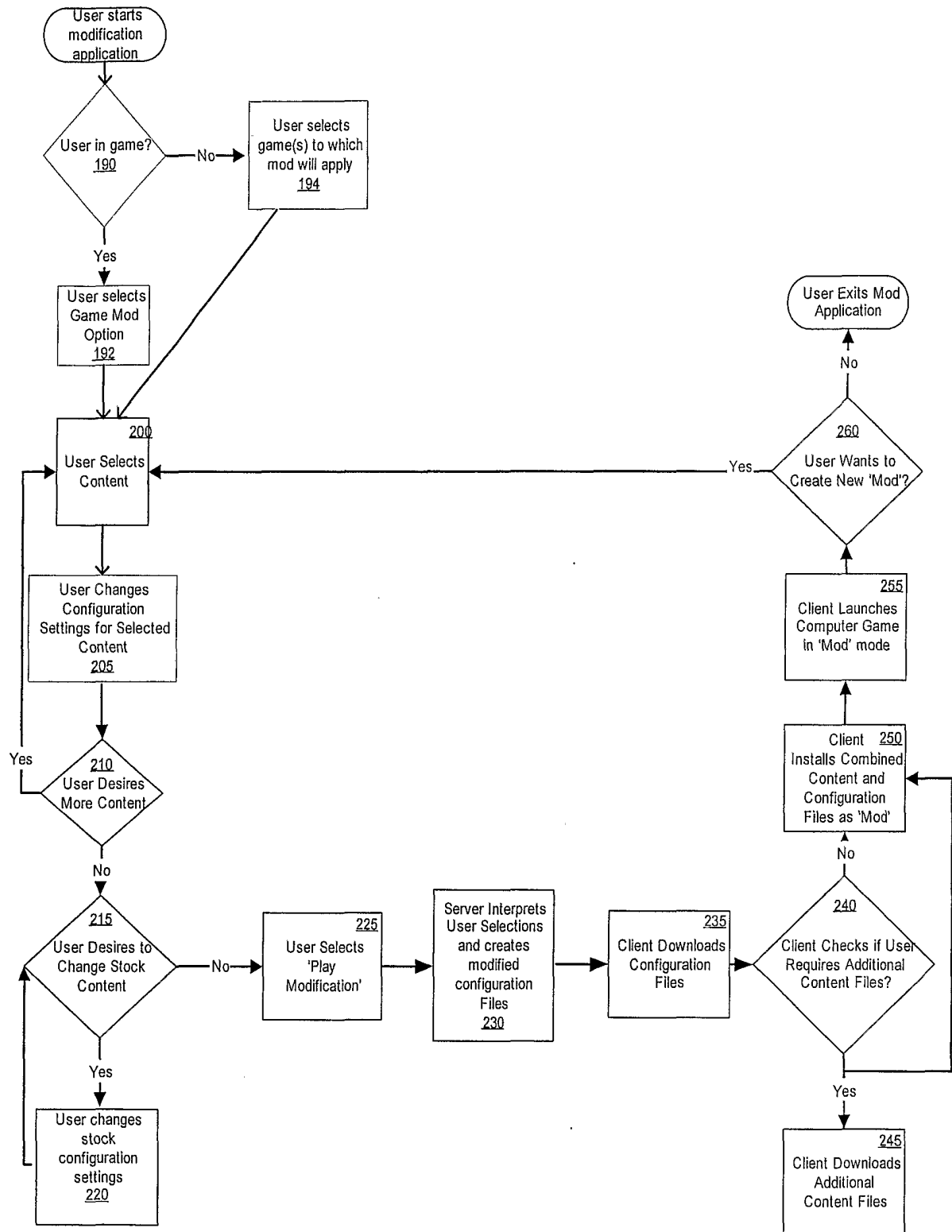


FIGURE 2

FIGURE 3

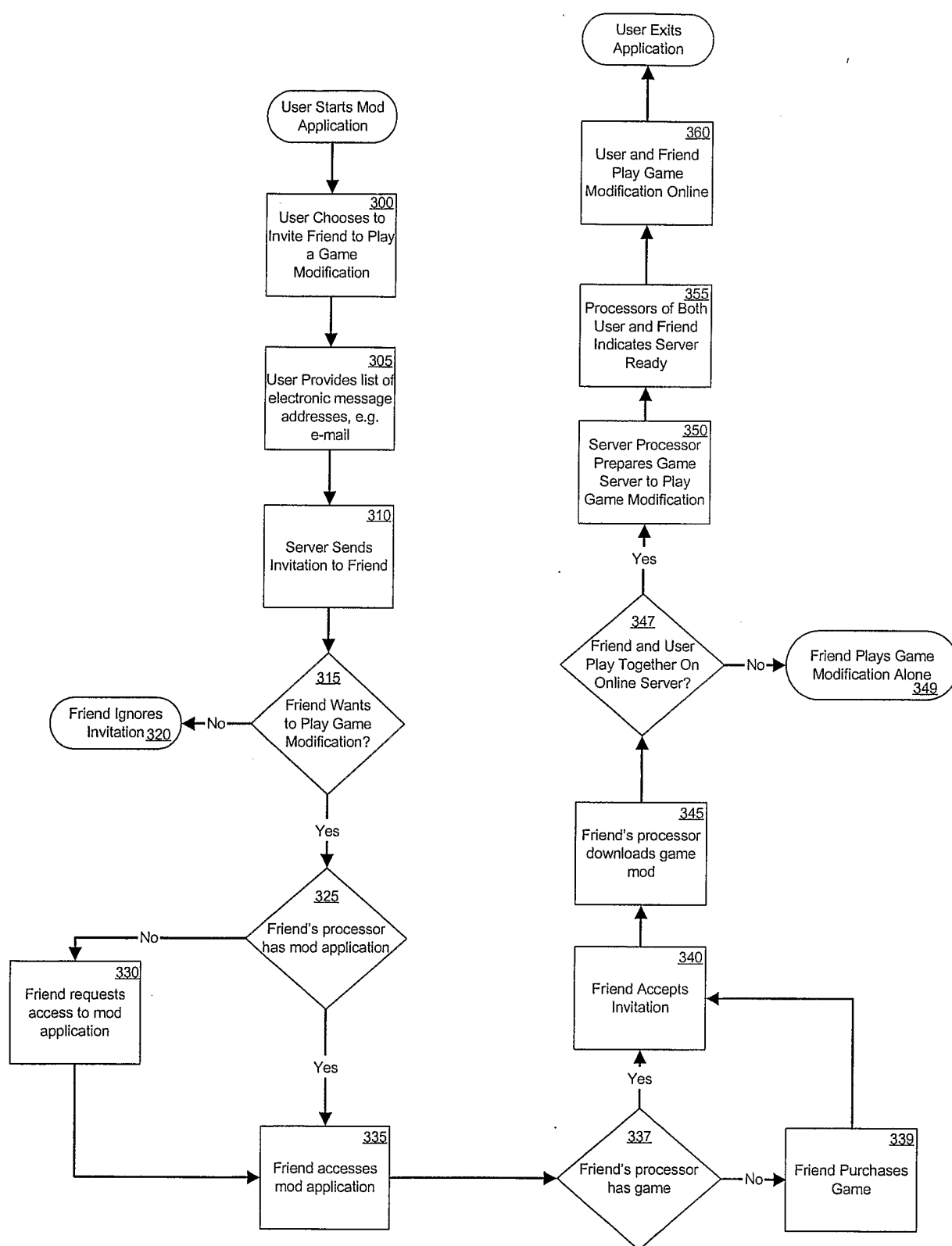
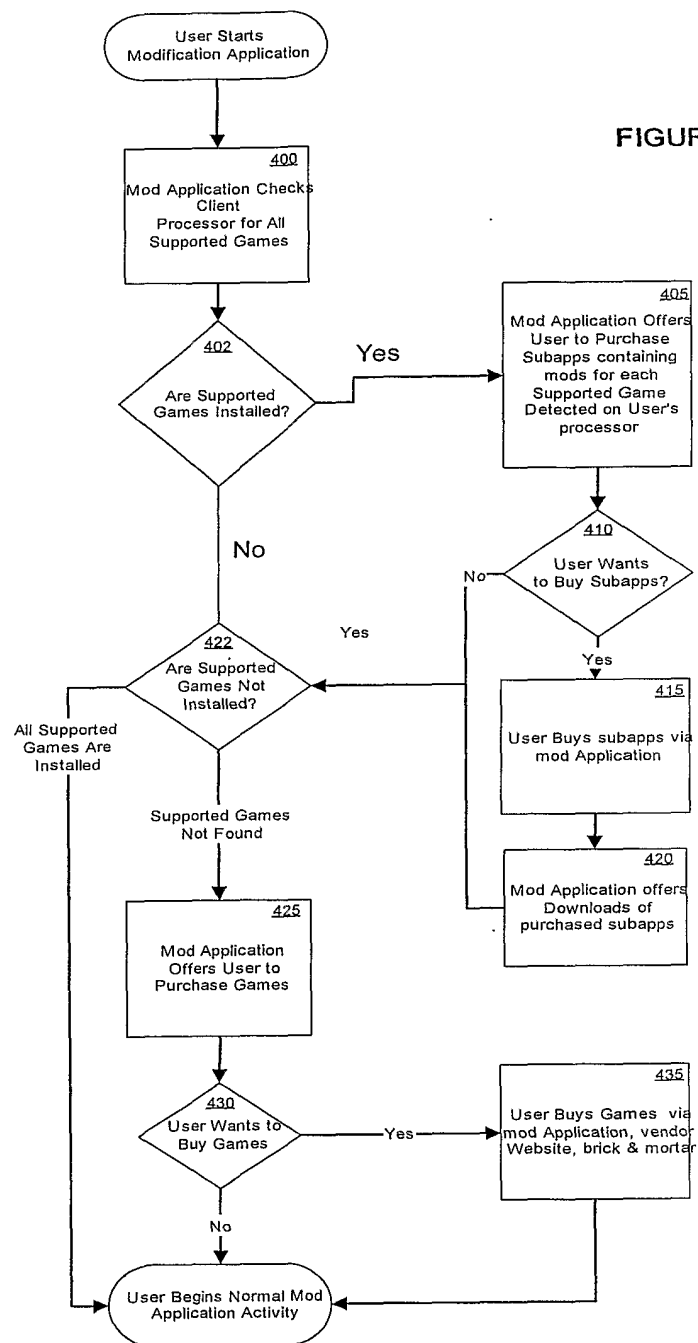


FIGURE 4



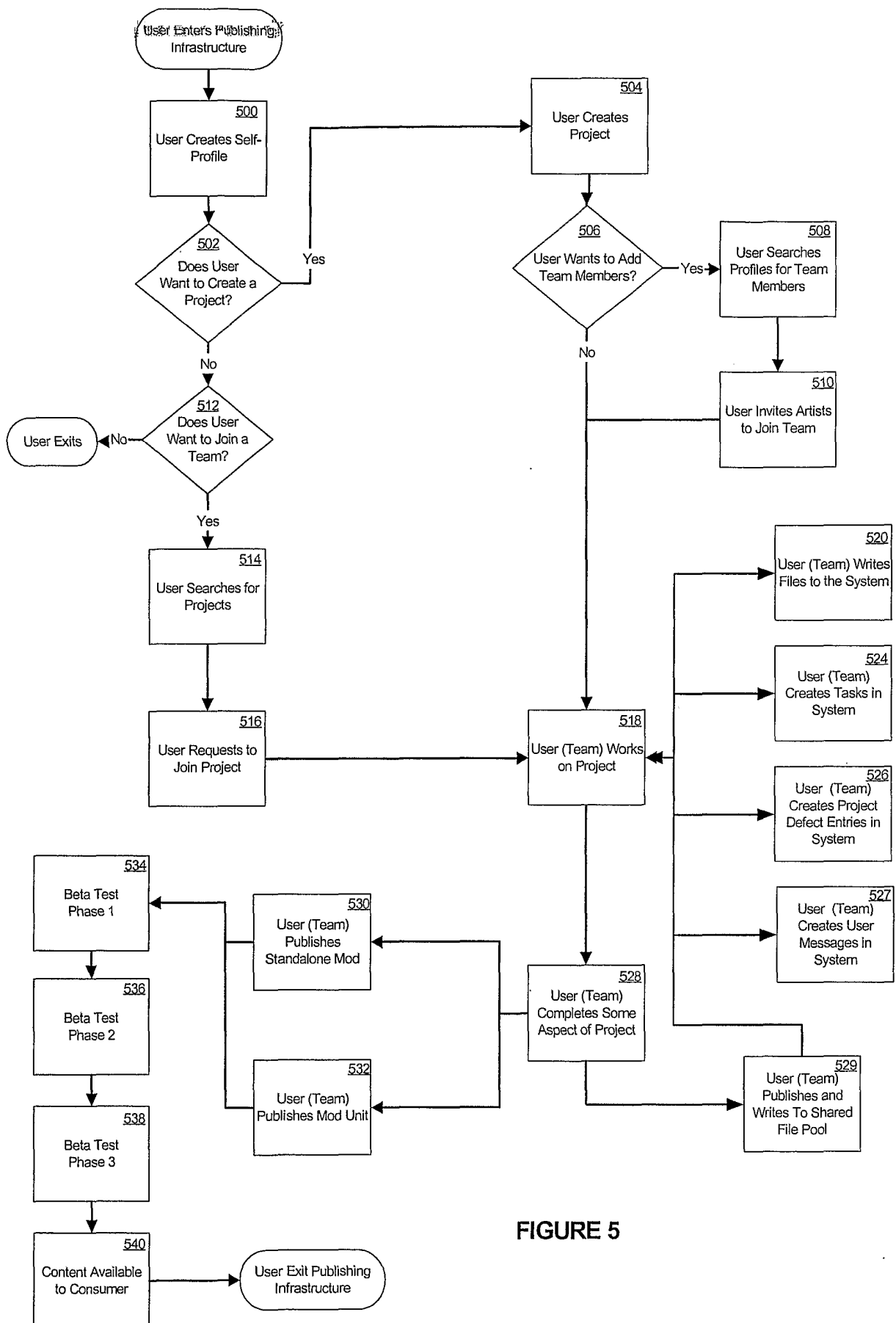


FIGURE 5