A franchise game mechanic for use in an online social game that promotes interaction between users is disclosed. An exemplary method comprises: permitting, by one or more computing devices, a first player to access a virtual game environment of a second player, the virtual game environment corresponding to a computer implemented game; responsive to inputs of the first player, locating, by the one or more computing devices; a franchise object within the virtual game environment of the second player; executing, by the one or more computing devices, a harvest mechanic cycle in connection with the franchise object of the first player; and distributing an in-game award to the first and second player upon completion of the harvest mechanic cycle.
Franchise Mechanic For Interactive Social Games

TECHNICAL FIELD

[0001] The present disclosure generally relates to games and applications in general and, in particular embodiments, to computer-implemented, online social games.

BACKGROUND

[0002] In many games, there is a virtual world or some other imagined playing space where a player/user of the game controls one or more player characters (herein "character", "player character," or "PC"). Player characters can be considered in-game representations of the controlling player. As used herein, the terms "player," "user," "entity," and "friend" may refer to the in-game player character controlled by that player, user, entity, or friend, unless context suggests otherwise. The game display can display a representation of the player character. A game engine accepts inputs from the player, determines player character actions, decides outcomes of events and presents the player with a game display illuminating what happened. In some games, there are multiple players, wherein each player controls one or more player characters.

[0003] In many computer games, there are various types of in-game assets (aka "rewards" or "loot") that a player character can obtain within the game. For example, a player character may acquire game points, gold coins, experience points, character levels, character attributes, virtual cash, game keys, or other in-game items of value. In many computer games, there are also various types of in-game obstacles that a player must overcome to advance within the game. In-game obstacles can include tasks, puzzles, opponents, levels, gates, actions, etc. In some games, a goal of the game may be to acquire certain in-game assets, which can then be used to complete sub-game tasks or to overcome certain in-game obstacles. For example, a player may be able to acquire a virtual key (i.e., the in-game asset) that can then be used to open a virtual door (i.e., the in-game obstacle).

[0004] An electronic social networking system typically operates with one or more social networking servers providing interaction between users such that a user can specify other users of the social networking system as "friends." A collection of users and the "friend" connections
between users can form a social graph that can be traversed to find second, third and more remote connections between users, much like a graph of nodes connected by edges can be traversed.

[0095] Many online computer games are operated on an online social network. Such a network allows both users and other parties to interact with the computer games directly, whether to play the games or to retrieve game-related information. Internet users may maintain one or more accounts with various service providers, including, for example, online game networking systems and online social networking systems. Online systems can typically be accessed using browser clients: (e.g., Firefox, Chrome, Internet Explorer).

[0006] In many computer games, there are various types of in-game actions that a player character can make within the game. For example, a player character in an online role-playing game may be able to interact with other player characters, build a virtual house, attack enemies, go on a quest, go to a virtual store to buy/sell virtual items, etc. A player character in an online poker game may be able to play at specific tables, place bets of virtual currency for certain amounts, play or fold certain hands, play in an online poker tournament, etc.

**DESCRIPTION OF EXAMPLE EMBODIMENTS**

[0016] FIG. 1 illustrates an example of a system for implementing various disclosed embodiments. FIG. 2 illustrates an example social network.

[0007] FIG. 3 illustrates an example of a game interface for an online game.

[0009] FIG. 4 illustrates an example of a game interface for an online game.

[0110] FIG. 3 is a flow chart illustrating an example process flow.

[0129] FIG. 6 illustrates an example of a game interface to an online game.

[0130] FIG. 7 illustrates an example data flow in a system.

[0131] FIG. 8 illustrates an example network environment.

[0135] FIG. 9 illustrates an example computer system architecture.
system 120a, game networking system 120b, client system 130, and network 160. The components of system 100 can be connected to each other in any suitable configuration, using any suitable type of connection. The components may be connected directly or over a network 160, which may be any suitable network. For example, one or more portions of network 160 may be an ad hoc network, an intranet, an extranet, a virtual private network (VPN), a local area network (LAN), a wireless LAN (WLAN), a wide area network (WAN), a wireless WAN (WWAN), a metropolitan area network (MAN), a portion of the Internet, a portion of the Public Switched Telephone Network (PSTN), a cellular telephone network, another type of network, or a combination of two or more such networks.

[0017] Social network system 120a is a network-addressable computing system that can host one or more social graphs: Social networking system 120a can generate, store, receive, and transmit social networking data. Social network system 120a can be accessed by the other components of system 100 either directly or via network 160. Game networking system 120b is a network-addressable computing system that can host one or more online games. Game networking system 120b can generate, store, receive, and transmit game-related data, such as, for example, game account data, game input, game state data, and game displays. Game networking system 120b can be accessed by the other components of system 100 either directly or via network 160. Player 101 may use client system 130 to access, send data to, and receive data from social network system 120a and game networking system 120b. Client system 130 can access social networking system 120 or game networking system 120b directly, via network 160, or via a third-party system. As an example and not by way of limitation, client system 130 may access game networking system 120b via social networking system 120a. Client system 130 can be any suitable computing device, such as a personal computer, laptop, cellular phone, smart phone, computing tablet, etc.

[0018] Although FIG. 1 illustrates a particular number of players 101, social network systems 120a, game networking systems 120b, client systems 130, and networks 160, this disclosure contemplates any suitable number of players 101, social network systems 120a, game networking systems 120b, client systems 130, and networks 160. As an example and not by way of limitation, system 100 may include one or more game networking systems 120b and no social networking systems 120a. As another example and not by way of limitation, system 100 may
include a system that comprises both social networking system 120a and game networking system 120b. Moreover, although Fig. 1 illustrates a particular arrangement of player 101, social networking system 120a, gaming networking system 120b, client system 130, and network 160, this disclosure contemplates any suitable arrangement of player 101, social networking system 120a, gaming networking system 120b, client system 130, and network 160.

The components of system 100 may be connected to each other using any suitable connections 110. For example, suitable connections 110 include wireline (such as, for example, Digital Subscriber Line (DSL) or Data Over Cable Service Interface Specification (DOCSIS)), wireless (such as, for example, Wi-Fi or Worldwide Interoperability for Microwave Access (WiMAX)) or optical (such as, for example, Synchronous Optical Network (SONET) or Synchronous Digital Hierarchy (SDH)) connections. In particular embodiments, one or more connections 110 each include an ad-hoc network, an intranet, an extranet, a VPN, a LAN, a WLAN, a WAN, a WWAN, a MAN, a portion of the internet, a portion of the PSTN, a cellular telephone network, or another type of connection, or a combination of two or more such connections. Connections 110 need not necessarily be the same throughout system 100. One or more first connections 110 may differ in one or more respects from one or more second connections 110. Although Fig. 1 illustrates particular connections between player 101, social networking system 120a, gaming networking system 120b, client system 130, and network 160, this disclosure contemplates any suitable connections between player 101, social networking system 120a, gaming networking system 120b, client system 130, and network 160. As an example and not by way of limitation, in particular embodiments, client system 130 may have a direct connection to social networking system 120a or gaming networking system 120b, bypassing network 160.

Online Games and Game Systems

Game Networking Systems

In an online computer game, a game engine manages the game state of the game. Game state comprises all game play parameters, including player character state, non-player character (NPC) state, in-game object state, game world state (e.g., internal game clocks, game environment), and other game play parameters. Each player 101 controls one or more player characters (PCs). The game engine controls all other aspects of the game, including non-player...
characters (NPCs), and in-game objects. The game engine also manages game state, including player-character state for currently active (online) and inactive (offline) players.

An online game can be hosted by a game networking system 120b, which can be accessed using any suitable connection with a suitable client system 130. A player may have a game account on game networking system 120b, wherein the game account can contain a variety of information associated with the player (e.g., the player's personal information, financial information, purchase history, player character state, game state). In some embodiments, a player may play multiple games on game networking system 120b, which may maintain a single game account for the player with respect to all the games, or multiple individual game accounts for each game with respect to the player. In some embodiments, game networking system 120b can assign a unique identifier to each player 101 of an online game hosted on game networking system 120b. Game networking system 120b can determine that a player 101 is accessing the online game by reading the user's cookies, which may be appended to HTTP requests transmitted by client system 130, and/or by the player 101 logging onto the online game.

In particular embodiments, player 101 may access an online game and control the game's progress via client system 130 (e.g., by inputting commands to the game at the client device). Client system 130 can display the game interface, receive inputs from player 101, transmitting user inputs or other events to the game engine, and receive instructions from the game engine. The game engine can be executed on any suitable system (such as, for example, client system 130, social networking system 120a, or game networking system 120b). As an example and not by way of limitation, client system 130 can download client components of an online game, which are executed locally, while a remote game server, such as game networking system 120b, provides backend support for the client components and may be responsible for maintaining application data of the game, processing the inputs torn the player, updating and/or synchronising the game state based on the game logic and each input from the player, and transmitting instructions to client system 130. As another example and not by way of limitation, each time player 101 provides an input to the game through the client system 130 (such as, for example, by typing on the keyboard or clicking the mouse of client system 130), the client-components of the game may transmit the player's input to game networking system 120b.
In an online multiplayer game, players may control player characters (PCs), a game engine controls non-player characters (NPCs) and game features, and the game engine also manages player character state and game state and tracks the state for currently active (i.e., online) players and currently inactive (i.e., offline) players. A player character can have a set of attributes and a set of friends associated with the player character. As used herein, the term “player character state” can refer to any in-game characteristic of a player character, such as location, assets, levels, condition, health, status, inventory, skill set, name, orientation, affiliation, specialty, and so on. Player characters may be displayed as graphical avatars within a user interface of the game. In other implementations, no avatar or other graphical representation of the player character is displayed. Game state encompasses the notion of player character state and refers to any parameter value that characterizes the state of an in-game element, such as a non-player character, a virtual object (such as a wall or castle), etc. The game engine may use player character state to determine the outcome of game events, sometimes also considering set or random variables. Generally, a player character’s probability of having a more favorable outcome is greater when the player character has a better state. For example, a healthier player character is less likely to die in a particular encounter relative to a weaker player character or non-player character. In some embodiments, the game engine can assign a unique client identifier to each player.

In particular embodiments, player 101 may access particular game instances of an online game. A game instance is copy of a specific game play area that is created during runtime. In particular embodiments, a game instance is a discrete game play area where one or more players 101 can interact in synchronous or asynchronous play. A game instance may be, for example, a level, zone, area, region, location, virtual space, or other suitable play area. A game instance may be populated by one or more in-game objects. Each object may be defined within the game instance by one or more variables, such as, for example, position, height/width, depth, direction, time, duration, speed, color, and other suitable variables. A game instance may be exehisive (i.e., accessible by specific players) or non-exclusive (i.e., accessible by any player). In particular embodiments, a game instance is populated by one or more player characters controlled by one or more players 101 and one or more in-game objects controlled by the game
engine. When accessing an online game, the game engine may allow player 101 to select a particular game instance to play from a plurality of game instances. Alternatively, the game engine may automatically select the game instance that player 101 will access. In particular embodiments, an online game comprises only one game instance that all players 101 of the online game can access.

In particular embodiments, a specific game instance may be associated with one or more specific players. A game instance is associated with a specific player when one or more game parameters of the game instance are associated with the specific player. As an example and not by way of limitation, a game instance associated with a first player may be named "First Player's Play Area." This game instance may be populated with the first player's PC and one or more in-game objects associated with the first player. In particular embodiments, a game instance associated with a specific player may only be accessible by that specific player. As an example and not by way of limitation, a first player may access a first game instance when playing an online game, and this first game instance may be inaccessible to all other players. In other embodiments, a game instance associated with a specific player may be accessible by one or more other players, either synchronously or asynchronously with the specific player's game play. As an example and not by way of limitation, a first player may be associated with a first game instance, but the first game instance may be accessed by all first-degree friends in the first player's social network. In particular embodiments, the game engine may create a specific game instance for a specific player when that player accesses the game. As an example and not by way of limitation, the game engine may create a new game instance each time a first player accesses an online game, wherein each game instance may be created randomly or selected from a set of predetermined game instances. In particular embodiments, the set of in-game actions available to a specific player may be different in a game instance that is associated with that player compared to a game instance that is not associated with that player. The set of in-game actions available to a specific player in a game instance associated with that player may be a subset, superset, or independent of the set of in-game actions available to that player in a game instance that is not associated with him. As an
example and not by way of limitation, a first player may be associated with Sla\k\ae re Farm in an online farming game. The first player may be able to plant crops on Black\aere Farm. If the first player accesses game instance associated with another player, such as White\aere Farm, the game engine may not allow the first player to plant crops in that game instance. However, other in-game actions may be available to the first player, such as watering or fertilizing crops on White\aere Farm.

[0026] In particular embodiments, a game engine can interlace with a social graph. Social graphs are models of connections between entities (e.g., individuals, users, contacts, friends, players, player characters, non-player characters, businesses, groups, associations, concepts, etc.). These entities are considered "users" of the social graph; as such, the terms "entity" and "user" may be used interchangeably when referring to social graphs herein. A social graph can have a node for each entity and edges to represent relationships between entities. A node in a social graph can represent any entity. In particular embodiments, a unique client identifier can be assigned to each user in the social graph. This disclosure assumes that at least one entity of a social graph is a player or player character in an online multiplayer game, though this disclosure any suitable social graph users.

[0027] The minimum number of edges required to connect a player (or player character) to another user is considered the degree of separation between diem. For example, where the player and the user are directly connected (one edge), they are deemed to be separated by one degree of separation. The user would be a so-called "first-degree friend" of the player. Where the player and the user are connected through one other user (two edges), they are deemed to be separated by two degrees of separation. This user would be a so-called "second-degree friend" of the player. Where the player and the user are connected through N edges (or N-1 other users), they are deemed to be separated by N degrees of separation. This user would be a so-called "Nth-degree friend." As used herein, the term "friend" means only first-degree friends, unless context suggests otherwise.

[0028] Within the social graph, each player (or player character) has a social network. A player’s social network includes all users in the social graph within N\textsuperscript{max} degrees of the player, where N\textsuperscript{max} is the maximum degree of separation allowed by the system managing the social graph (such as, for example, social networking system 120a or game networking system 120b).
In one embodiment, \( N_{\text{max}} \) equals 1, such that the player’s social network includes only first-degree friends. In another embodiment, \( N_{\text{max}} \) is unlimited and the player’s social network is coextensive with the social graph.

In particular embodiments, the social graph is managed by a game networking system 120b, which is managed by the game operator, in other embodiments, the social graph is part of a social networking system 120a managed by a third-party (e.g., Facebook, Friendster, Myspace). In yet other embodiments, player 101 has a social network on both game networking system 120b and social networking system 120a, wherein player 101 can have a social network on the game networking system 120b that is a subset, superset, or independent of the player’s social network on social networking system 120a. In such combined systems, game networking system 120b can maintain social graph information with edge type attributes that indicate whether a given friend is an “in-game friend”; an “out-of-game friend,” or both. The various embodiments disclosed herein are operable when the social graph is managed by soesai networking system 120a, game networking system 120b, or both.

FIG. 2 shows an example of a social network within a social graph. As shown, Player 201 can be associated, connected or linked to various other users, or “friends,” within the social network 250. These associations, connections or links can track relationships between users within the social network 250 and are commonly referred to as online “friends” or “friendships” between users. Each friend or friendship within a particular user’s social network within a social graph is commonly referred to as a “node.” For purposes of illustration and not by way of limitation, the details of social network 250 will be described in relation to Player 201. As used herein, the terms “player” and “user” can be used interchangeably and can refer to any user or character in an online multiplayer game system or social networking system. As used herein, the term “friend” can mean any node within a player’s social network.

As shown in FIG. 2, Player 201 has direct connections with several friends. When Player 201 has a direct connection with another individual, that connection is referred to as a first-degree friend. In social network 250, Player 201 has two first-degree friends. That is, Player 201 is directly connected to Friend 1 211 and Friend 2 221 in a social graph, it is possible for individuals to be connected to other individuals through their first-degree friends (i.e., friends of friends). As described above, each edge required to connect a player to another user is
considered the degree of separation. For example, FIG. 2 shows that Player 201 has three second-degree friends to which he is connected via his connection to his first-degree friends. Second-degree Friend 12 212 and Friend 222 are connected to Player 201 via his first-degree Friend 211. The limit on the depth of friend connections, or the number of degrees of separation for associations, that Player 201 is allowed is typically dictated by the restrictions and policies implemented by social networking system 210.

[0032] In various embodiments, Player 201 can have Nth-degree friends connected to him through a chain of intermediary-degree friends as indicated in FIG. 2. For example, Nth-degree Friend 11 219 is connected to Player 201 via second-degree Friend 3 232 and one or more other higher-degree friends. Various embodiments may take advantage of and utilize the distinction between the various degrees of friendship relative to Player 201.

[0033] In particular embodiments, a player (or player character) can have a social graph within an online multiplayer game that is maintained by the game engine and another social graph maintained by a separate social networking system, FIG. 2 depicts an example of In-game social network 260 and out-of-game social network 250, in this example. Player 201 has out-of-game connections 255 to a plurality of friends, forming om-of-game social network 250. Here, Friend 12 211 and Friend 222 are first-degree friends with Player 201 in his out-of-game social network 250, Player 201 also has in-game connections 265 to a plurality of players, forming in-game social network 260. Here, Friend 221, Friend 323, and Friend 4 241 are first-degree friends with Player 201 in his in-game social network 260. In some embodiments, it is possible for a friend to be in both the out-of-game social network 250 and the in-game social network 260. Here, Friend 221 has both an out-of-game connection 255 and an in-game connection 265 with Player 201, such that Friend 221 is in both Player 201’s in-game social network 260 and Player 201’s out-of-game social network 250.

[0034] As with other social networks, Player 201 can have second-degree and higher-degree friends in both his in-game and out of game social networks, in some embodiments, it is possible for Player 201 to have a friend connected to him both in his in-game and out-of-game social networks, where the friend is at different degrees of separation in each network. For example, if Friend 2222 had a direct in-game connection with Player 201, Friend 2222 would be a second-degree friend in Player 201’s out-of-game social network, but a first-degree friend in
Player 201’s in-game social network. In particular embodiments, a game engine can access in-game social network 260, out-of-game social network 250, or both.

[0035] In particular embodiments, the connections in a player’s in-game social network can be formed both explicitly (e.g., users roost “friend” each other) and implicitly (e.g., system observes user behaviors and “friends” users to each other). Unless otherwise indicated, reference to a friend connection between two or more players can be interpreted to cover both explicit and implicit connections, using one or more social graphs and other factors to infer friend connections. The friend connections can be unidirectional or bidirectional. It is also not a limitation of this description that two players who are deemed “friends” for the purposes of this disclosure—are not friends in real life (i.e., in disintermediated interactions or the like), but that could be the case.

Game Systems

[0036] A game event may be an outcome of an engagement, a provision of access, rights and/or benefits, or the obtaining of some assets (e.g., health, money, strength, inventory, land, etc.). A game engine determines the outcome of a game event according to a variety of factors, such as the game rules, a player character’s in-game actions, player character state, game state, interactions of other player characters, and random calculations. Engagements can include simple tasks (e.g., plant a crop, clean stove), complex tasks (e.g., build a farm or business, run a cafe), or other events.

[0037] An ongoing game can be hosted by a game networking system 320, which can be accessed over any suitable network with an appropriate client system 330. A player may have a game system account on game system 320, wherein the game system account cars contain a variety of information about the player (e.g., the player’s personal information, player character state, game state, etc.). In various embodiments, an online game can be embedded into a third-party website. The game can be hosted by the networking system of the third-party website, or it can be hosted on game system 320 and merely accessed via the third-party website. The embedded online game can be hosted solely on a server of game system 320 or using a third-party vendor server. In addition, any combination of the functions of the present disclosure can be hosted on or provided from any number of distributed network resources. For example, one
or more executable code objects that implement all or a portion of the game can be downloaded to a client system for execution.

Game Interfaces

[0038] Fig. 3 illustrates an example of a webpage-based game interface for an online game accessed by a browser client 110 (e.g., Firefox, Chrome, Internet Explorer, etc.). In various embodiments, a user of a client system 130 can use a browser client to access the online game over the internet (or other suitable network). The game interface 370 illustrated in Fig. 3 may be automatically generated and presented to the user in response to the user visiting or accessing the game operator's website or a third-party's website from client system 130 with a browser client. Game system 120b can transmit data to client system 130 allowing it to display game interface 370, which is typically some type of graphic user interface. For example, the webpage downloaded to client system 130 may include an embedded call that causes client system 130 to download an executable object, such as a Flash .SWF object, which executes on client system 130 and renders the game within the context of the webpage. Other interface types are possible, such as server-side rendering and the like. Game interface 370 is configured to receive signals from the user via client system 130. For example, the user can click on game interface 370, or enter commands from a keyboard or other suitable input device. The game engine can respond to these signals to allow game play. The display of game Interface 370 can change based on the output of the game engine, the input of the player, and other signals from game system 120b and client system 130.

[0039] The game interface 370 can display various game components, such as the game environment, options available to the player (e.g., in-game actions, preferences, settings, etc.), game results, etc. Some components of the game interface may be static, while others may be dynamic (e.g., changing with game play). The user may be able to interact with some components (e.g., player character, NPCs, virtual objects, etc.) and not interact with other components (e.g., the background of the virtual world, such as the virtual street or sidewalk). The user can engage in specific in-game actions or activities by providing input to game interface 370.

[0040] In the example online game illustrated in Fig. 3, the user controls a virtual city implemented in an online social game. Game interface 370 shows the virtual city and -various
components of the city. The player can interact with various elements of the ecosphere, such as:
the virtual business objects 320 and virtual housing objects 330. The user can (or otherwise
activate) various aspects of the game interface to provide instructions to the game engine. For
example, by clicking on a virtual business object 320, the user can initiate a harvest mechanic
cycle as described in more detail below.

The user can also click on various icons in game interface 370 to activate various
game options. For example, if the user clicks on one of the icons in option bar 360, the game
engine will alter the game interface 370 to present the user with options for buying and selling
virtual items for use in the virtual city. For example, the player could buy or sell virtual furni-
ture, appliances, decor, windows, goods, etc. Similarly, the user can click on other icons in option bar
360 to access other game options.

One skilled in the art would appreciate that Fig. 3 is presented merely as an example
of an embodiment of one type of online game and that the present disclosure is intended to
encompass a variety of game types, including gambling games, role-playing games, puzzle
games, etc.

Virtual Currency

At various embodiments, players within the game earn acquire virtual currency. In
such games, the virtual currency might be represented by virtual coins, virtual cash, or by a
number—w value stored by the server for that player's benefit. Such virtual currency represents
units of value for use in the online game system, and is analogous to legal currency. Virtual
currency can be purchased in one or more actual cash or credit transactions by a player, where
the legal currency is transferred using a credit/debit/charge card transaction conveyed over a
financial network. In some embodiments, a player may earn virtual currency by taking action in
the game. For example, a player may be rewarded with one or more units of virtual currency
after completing a task, quest, challenge, or mission within the game. For example, a farming
game might reward 10 gold coins each time a virtual crop is harvested.

In some embodiments, virtual currency can be used to purchase one or more in-game
assets or other benefits. For example, a player may be able to exchange virtual currency for a
desired level, access, right, or item in an online game. In one embodiment, legal currency can be
used to directly purchase an in-game asset or other benefit. The player can select the desired in-
game asset or other benefit. Once the necessary selections are made, the player can place the order to purchase the in-game asset or other benefit. This order is received by the game system 320, which can then process the order. If the order is processed successfully, an appropriate financial account associated with the player can be debited by the amount of virtual currency or legal currency needed to buy the selected in-game asset or other benefit.

[0048] In some embodiments, multiple types of virtual currency may be available for purchase from the game system operator. For example, an online game may have virtual gold coins and virtual cash. The different types of virtual currency may have different exchange rates with respect to legal currency and each other. For example, a player may be able to exchange $1 in legal currency for either 100 virtual gold coins or $2 in virtual cash, but virtual gold coins may not be exchanged for virtual cash. Similarly, where in-game assets and other benefits can be purchased with virtual currency, they may have different exchange rates with respect to the different types of virtual currency. For example, a player may be able to buy a virtual business object for $10 in virtual cash, but may not purchase the virtual business object for virtual gold coins alone. In some embodiments, certain types of virtual currency can be acquired by engaging in various in-game actions while other types of virtual currency can only be acquired by exchanging legal currency. For example, a player may be able to acquire virtual gold coins by selling virtual goods in a business, but can only acquire virtual cash by exchanging legs! currency. In some implementations, virtual cash may also be awarded for leveling up in the game.

Harvest Mechanic

[0046] In various embodiments, an online game can include a series of user-initiated in-game actions that comprise a harvest mechanic. The harvest mechanic generally has two components; a set of initiating actions and a set of collecting (harvesting) actions. In an initiating action, a player may "click" or otherwise interact with an element of the online game to initiate the harvest mechanic. In some games, multiple clicks or other actions may be necessary to complete the initiating action. Furthermore, in some implementations, the initiating actions may require the player to expend resources, such as virtual currency, energy, and the like. For example, a player may have to click: on a stove to clean it and click again to purchase and prepare ingredients and initiate cooking virtual food; in another example, a player may have to click on a
segment of land to plow it and click again to purchase and plant seeds and initiate: growing virtual crops.

[0047] Once the initiating action is complete, the online game may begin a processing action, wherein the game state of the element is modified by the game system during some waiting time period. The time period can range from seconds to days, depending on the game system. For example, the game system may require an hour for virtual food to get cooked, or it may require three-days for a virtual crop to grow. During the processing action, the player may be able to cancel the processing action, thereby resetting the entire harvest mechanic sequence. Dining the processing action, the player may also be able to interact with the element. For example, the player could add spice to his virtual food while it is cooking, or the player could add fertilizer to his virtual crops while they are growing. These interactions may or may not have an in-game effect on the element. In some embodiments, there is no processing action or waiting time period, and the collecting action is accessible immediately after the initiating action is complete. In other embodiments, the processing action is instantaneous.

[0048] After the processing action is complete, the game system can alter the game state of the in-game asset. At this point, the collecting action may be available. In a collecting action, a player may click or otherwise interact with the element to complete the harvest mechanic cycle. For example, a player may have to click on the fully-cooked virtual food to serve it. In another example, a player may have to click on fully-grown crops to harvest them. In some games, multiple clicks or other actions may be necessary to complete the collecting action. Completion of the harvest mechanic typically results in a reward, such as virtual currency, in-game assets or other loot. Some harvest mechanics may also include an explicit penalty for not initiating the collection action within a threshold period of time. For example, crops may wither if not harvested within X hours of completing the processing action, where X may vary depending on crop type or other considerations. In addition, completion of selected or all phases of a harvest mechanic may result in awards of experience points to the player.

[0049] Once the collecting action is complete, the harvest mechanic may be reset with respect to that game element. In some embodiments, the player may have to click or otherwise interact with the in-game asset to reset the harvest mechanic. In other embodiments, the harvest
mechanic may reset automatically once the collecting action is complete. Once the harvest mechanic is reset, the initiating action may again be available for that game element.

A detailed example of the harvest mechanic follows in connection with an in-game asset representing a business, NPCs may visit the business and make purchases of items. The business may be a retail store, such as a restaurant or toy shop. The business may be visually represented as an in-game object located within the virtual game instance associated with the player. Figure 3 illustrates an example business object 320 located in a game instance of the player. In some implementations, the game logic may require the player to purchase the business and/or perform a series of operations to build the business, which such actions may require the expenditure of energy or some in-game credit or currency. In this manner, a player may establish a plurality of businesses of the same or different type within the game instance of the player.

In one implementation, the harvest mechanic associated with the business object 320 may involve three phases. An initiating phase may involve the user supplying the business with units of virtual goods or supplies. In one implementation, the minimum number of units required to initiate the harvest mechanic may vary depending on the type of business. The player may be required to purchase goods units (or replenish an existing supply of goods units) using virtual currency or other in-game assets or credits. The wait or processing phase may involve one or more NPCs (or player characters separately) visiting the business to consume the goods of the business. The rate at which goods are consumed can be based on the number or population of NPCs in the game instance of the player. Game logic may apply a function that considers a variety of attributes to control the population of NPCs, such as the number and size of the housing structures within the game instance of a player. The processing phase ends and the collection (harvesting) phase begins, when all goods of the business have been consumed. For example, the business object may include a visual identifier that indicates that collection is available. A user may click on the business object to collect virtual currency (or other loot) resulting from completion of the harvest mechanic cycle. In some implementations, there is no wither or penalty component. In other implementations, a penalty can be assessed if a player fails to initiate a collection operation within a threshold period of time. For example, a business may be "robbed" or employees may steal from use cash registers.
Franchise Mechanic

[0052] As discussed in more detail below, the game application may support a franchise mechanic that allows players to locate in-game assets within the virtual game instances of other players and/or non-player characters. The franchise mechanic described herein may increase social interaction between users, thereby increasing user retention, user-activity and virality.

[0053] As Figure 3 illustrates, the user interface of the game application may include a panel 360 including avatars 365 corresponding to the in-game contacts of the player. A player may select an icon or avatar 365 corresponding to an in-game contact of the player (or a NPC), which causes the game application to display the game instance of the selected contact (or NPC). In some implementations, the game application may present the player with an opportunity to place an in-game asset within the game instance of the enlaet--referred to in this description for convenience as a franchise object. For example, in implementations involving business objects, a player may select a business object of the same type as exists in the game instance of the player and locate and/or build a franchise business object of the selected type within the game instance of the contact. Figure 4 illustrates a graphical interface including the game instance of a contact of the player. The player may repeat this process for additional in-game contacts limited by the number of in-game contacts of the player, the availability of space within the virtual instances of the respective contacts, the resources required to establish the franchise business object, and/or rules embedded in-game logic.

[0054] In some implementations, the consent of the contact of the player to establish a franchise object is required. In such an implementation, a workflow can be initiated by which the contact is presented with a notification of the players request to establish a franchise object. The contact may consent by, for example, clicking on an “OK” button or the like, in other implementations the contact can designate a location in his or her game instance and then invite his or her in game contacts (or a selected subset of in-game contacts) to establish a franchise object on the designated location. The invitation can be sent using any suitable messaging infrastructure, such as email, chat, invitations on a social network, wall posts, and the like. The invitation may also include one or more other attributes of a franchisor-franchisee relationship, such as the revenue -share split between the franchisor and franchisee player. A franchisor player may accept the offered terms implicitly by establishing a franchise business object on the
designate location. Furthermore, the designated locations may be reserved for particular in-game contacts.

[0055] Once located within the game instance of the contact, a modified version of the harvest mechanic may be executed with respect to the franchise business object. For example, in an example modified version of the harvest mechanic, the player (in this instance, a "franchisor") has the option to initiate (or contribute to initiation of) the harvest mechanic by supplying the franchise business object with all (or a subset) of the virtual supply units required to initiate the cycle. In some implementations, it may be less expensive for the franchisor to supply the goods for initiation of the harvest mechanic (relative to virtual currency, energy consumption and/or another in-game currency, credit or asset) that the same or similar supply operations associated with the franchisor players own (non-franchise) business objects. This may be a farther incentive for the player to supply the franchise object, thereby interacting on some level with the contact. In some implementations, the contact also may initiate the harvest mechanic as well. In some implementations, the contact of the player (here, a "franchisee player"), when playing in connection with his or her own game instance, may click on the franchise business object to acknowledge the supply contribution of the franchisor player and optionally receive a bonus determined by game logic. The processing phase may proceed similarly to a regular business object, as discussed above. In some implementations, however, one or more attributes of the processing phase may differ. For example, the consumption rate of the goods of the franchise business object may be greater than that of regular business objects. At the collection phase, the game logic may distribute the awards between the franchisor player and the franchisee player. A variety of award distribution implementations are possible. For example, in one implementation, the virtual currency normally awarded for completion of the harvest mechanic cycle in connection with a regular (non-franchise) business object may be shared between the franchisor player and the franchisee player. In some implementations, the game logic may award the normal amount of virtual currency to the franchisee player and a bonus amount to the franchisor player. An additional aspect of the modified harvest mechanic may involve the game logic granting an award to the franchisor player, if the franchisor player re-initiates the harvest mechanic by re-supplying the franchise business object.
To support the franchise harvest mechanic discussed above, the record, data object or other data structure that corresponds to each in-game object that can be a franchise object may include additional attributes, such as a franchisor field for storing an identifier for the franchisor player, a held or hit indicating which of the franchisor or franchisee initiated the harvest mechanic cycle. More generally, the data attributes associated with the game instance of a player may also include identifiers corresponding to one or more franchise objects of the player.

Figure 5 illustrates an example process flow directed to implementing an aspect of the franchise mechanic. Upon completion of the collection phase of the harvest mechanic, game logic may access the business object to determine whether it is a franchise object (502). If not, the game logic computes an award and allocates the award to the player (504), typically by accessing one or more data attributes of a player or game state object and incrementing a virtual currency or other credit or award value. If the business object is a franchise object, the game logic computes awards for the franchisor player and the franchisee player (506), respectively.

In some implementations, the game application may also allow the player to locate and/or build a franchise headquarters object within the game instance of the player after the player has established a threshold number (1 or more) of franchise business objects within the game instances of the player's contacts. Figure 6 illustrates a franchise headquarters object 602 according to one possible Implementation located within the game Instance of a player. The data attributes of the franchise headquarters object may include a franchise or business type identifier, pointers to one or more associated franchise objects, and the like. In one implementation, the game Instance of a player may include a single, global franchise headquarters object for the franchise objects across all business types, franchise headquarters object for each business type.

In one implementation, the game application may modulate the size of the franchise headquarters object 602 as a function of the number of associated franchise objects. The game application may set the number of stories (or other height unit) of the franchise headquarters object based on the number of associated franchise objects. In some implementations, there is a one-to-one mapping between the number of height units to associated franchise objects. In other implementations, a decay function can be employed such that more franchise objects are required
to grow the size of the franchise: headquarters object. In some implementations, the size of the franchise headquarters object can also be influenced by the aggregate ratings of the associated franchise objects. Other attributes in addition to, or in lieu of size, can be modulated, such as additional building features, cleanliness or other overall appearance, and the like.

[0060] Beyond the visual effect of the franchise headquarters object, the number and/or size of the franchise headquarters objects may affect how neighboring in-game assets function within the game instance of the player. In one implementation, each franchise headquarters object may have a beneficial halo effect on neighboring and/or nearby in-game assets. For example, one or more parameters that influence the operation of a harvest or other game mechanic relative to one or more in-game assets may be biased more favorably to the player. For example, any game related outcomes that are based at least in part on a chance-based event or condition can be biased to increase the player's chance of achieving a favorable outcome. In other implementations, parameters that affect the rate at which an in-game goal completion of a harvest mechanic cycle, or quest, is achieved can be increased. For example, the consumption rate of goods units in the harvest mechanic described above can be increased according to a function that considers the proximity (possibly measured in arbitrary grid units of the game instance) of an in-game asset to one or more franchise headquarters objects, and the respective sizes of (or some other metric associated with) the franchise headquarters objects. In other examples, the rem received from a housing object can be increased based on the foregoing factors. In yet other examples, the experience points awarded to a player for completing a harvest mechanic cycle or achieving another game goal can also be increased based on the foregoing factors.

[0861] The game logic may also include interfaces that facilitate the management of a player's franchise objects. For example, a player may click on a franchise headquarters object, causing the game application to launch a franchise management interface. The franchise management interface may display the franchise business objects, the contacts in whose game instances the franchise business objects, and an indicator of the relative health or prosperity of the franchise business object. For example, the game application may maintain a star or other rating system for each of the franchise business objects. The rating system may assess the number and/or frequency with which harvest mechanic cycles connected to the franchise
business object have been completed. In addition (or alternatively), the rating system may assess
the amount and/or frequency of the awards received by the franchisor player on account of
operation of the franchise business object. From the management interface, the player may
select a franchise business object, causing the game application to display the game instance of
the corresponding contact or franchisee player.

From the game instance of the contact or the management interface, the player may
also perform one or more management operations. For example, the player may elect to
"bulldoze" the franchise business object (essentially deleting it from the game instance of the
contact). In some implementations, the franchisor player may serve a virtual notice of intent to
terminate the franchise, causing the game application to initiate a termination workflow. Rules
implemented by the game application may allow the franchisee player to cure the default within
a defined period of time, in some implementations, rules implemented by game logic may
prevent a franchisor player from deleting a franchise game object; if its rating or other
performance metric exceeds a threshold. In some implementations, the resources of the
"bulldozed" franchise game object can be re-utilized to place the franchise game object with the
game instance of another contact. In addition, as part of a termination-workflow or otherwise,
the franchisor player may modify (or propose a modification to) any revenue splits or allocations
that are associated with the franchise relationship.

Data Flow

[0063] FIG. 7 illustrates an example data flow between the components of system 500. In
particular embodiments, system 500 can include client system 530, social networking system
520a, and game networking system 520b. The components of system 500 can be connected to
each other in any suitable configuration, using any suitable type of connection. The components
may be connected directly or over any suitable network. Client system 530, social networking
system 520a, and game networking system 520b can each have one or more corresponding data
stores such as local data store 535, social data store 545, and game data store 565, respectively.
Social networking system 520a and game networking system 520b can also have one or more
servers that can communicate with client system 530 over an appropriate network. Social
networking system 520a and game networking system 520b can have, for example, one or more
internet servers for communicating with client system 530 via the Internet. Similarly, social
networking system 520a and game networking system 520b can have one or more mobile servers for communicating with client system 530 via a mobile network (e.g., GSM, PCS, Wi-Fi, WPAN, etc.). In some embodiments, one server may be able to communicate with client system 530 over both the internet and a mobile network. In other embodiments, separate servers can be used.

Client system 530 can receive and transmit data 523 to and from game networking system 520b. This data can include, for example, webpages, messages, game inputs, game displays, HTTP packets, data requests, transaction information, updates, and other suitable data. At some other time, or at the same time, game networking system 520b can communicate data 543, 545 (e.g., game state information, game system account information, page into, messages, data requests, updates, etc.) with other networking systems, such as social networking system 520a (e.g., Facebook, Myspace, etc). Client system 530 can also receive and transmit data 523 to and from social networking system 520a. This data can include, for example, webpages, messages, social graph information, social network displays, HTTP packets, data requests, transaction information, updates, and other suitable data.

Communication between client system 530, social networking system 520a, and game networking system 520b can occur over any appropriate electronic communication medium or network using any suitable communications protocols. For example, client system 530, as well as various servers of the systems described herein, may include Transport Control Protocol/Internet Protocol (TCP/IP) networking stacks to provide for datagram and transport auctions. Of course, any other suitable network and transport layer protocols can be utilized.

In addition, hosts or end-systems described herein may use a variety of higher layer communications protocols, including client-server (or request-response) protocols, such as the HyperText Transfer Protocol (HTTP) and other communications protocols, such as HTTP-S, FTP, SNMP, TELNET, and a number of other protocols, may be used. In addition, a server in one interaction context may be a client in another interaction context. In particular embodiments, the information transmitted between hosts may be formatted as HyperText Markup Language (HTML) documents. Other structured document languages or formats can be used, such as XML, and the like, Executable code objects, such as JavaScript and ActionScript, can also be embedded in the structured documents.
In some client-server protocols, such as the use of HTML over HTTP, a server generally transmits a response to a request from a client. The response may comprise one or more data objects. For example, the response may comprise a first data object, followed by subsequently transmitted data objects. In particular embodiments, a client request may cause a server to respond with a first data object, such as an HTML page, which itself refers to other data objects. A client application, such as a browser, will request these additional data objects as it parses or otherwise processes the first data object.

In particular embodiments, an instance of an online game can be stored as a set of game stats parameters that characterize the state of various in-game objects, such as, for example, player character state parameters, non-player character parameters, and virtual item parameters. In particular embodiments, game state is maintained in a database as a serialized, unstructured string of text data as a so-called Binary Large Object (BLOB). When a player accesses an online game on a game networking system 520b, the BLOB containing the game state for the instance corresponding to the player can be transmitted to client system 530 for use by a client-side executed object to process. In particular embodiments, the client-side executable may be a FLASH-based game, which can deserialize the game state data in the BLOB. As a player plays the game, the game logic implemented at client system 530 maintains and modifies the various game state parameters locally. The client-side game logic may also batch game events, such as mouse clicks, and transmit these events to game networking system 520b. Game networking system 520b may itself operate by retrieving a copy of the BLOB from a database or an intermediate memory cache (memcache) layer. Game networking system 520b can also deserialize the BLOB to resolve the game state parameters and execute its own game logic based on the events in the batch file of events transmitted by the client to synchronize the game state on the server side. Game networking system 520b may then re-serialize the game state, now modified, into a BLOB and pass this to a memory cache layer for lazy updates to a persistent database.

With a client-server environment in which fee online games may run, one server system, such as game networking system 520b, may support multiple client systems 530. At any given time, there may be multiple players at multiple client systems 530 all playing the same online game. In practice, the number of players playing the same game at the same time may be
very large. As the game progresses with each player, multiple players may provide different inputs to the online game at their respective client systems 530, and multiple client systems 536 may transmit multiple player inputs and/or game events to game networking system 520b for further processing. In addition, multiple client systems 530 may transmit other types of application data to game networking system 520b.

In particular embodiments, a computer-implemented game may be a text-based or text-based game implemented as a series of web pages that are generated after a player selects one or more actions to perform. The web pages may be displayed in a browser client executed on a client system 530. As an example, and not by way of limitation, a client application downloaded to a client system 530 may operate to serve a set of webpages to a player. As another example and not by way of limitation, a computer-implemented game may be an animated or rendered game executable as a stand-alone application or within the context of a webpage or other structured document. In particular embodiments, the computer-implemented game may be implemented using Adobe Flash-based technologies. As an example and not by way of limitation, a game may be fully or partially implemented as a SWF object that is embedded in a web page and executable by a Flash media player plug-in. In particular embodiments, one or more described webpages may be associated with or accessed by social networking system 520a, tin's disclosure contemplates using any suitable application for the retrieval and rendering of structured documents hosted by any suitable network-addressable resource or website.

Application event data of a game is any data relevant to the game (e.g., player inputs). In particular embodiments, each application datum may have a name and a value, and the value of the application datum may change (be, be updated) at any time. When an update to an application datum occurs at client system 530, either caused by an action of a player or by the game logic itself, client system 530 may need to inform game networking system 520b of the update. For example, if the game is a farming game with a harvest mechanic (such as Zyrrga FarmVille), an event can correspond to a player clicking on a parcel of land to harvest a crop. In such an instance, the application event data may identify an event or action (e.g., harvest) and an object in the game to which the event or action applies. For illustration purposes and not by way of limitation, system 500 is discussed in reference to updating a multi-player online game hosted on a network-addressable system (such as, for example, social networking system 520a or game
networking system 520b), where: an instance of the online game is executed remotely on a client system 530, which flics transmits application event data to the hosting system such that the remote game server synchronizes; game state associated with the instance executed by the client system 530.

[0072] In particular embodiment, one or more objects of a game may be represented as an Adobe Flash object. Flash may manipulate vector and raster graphics, and supports bidirectional streaming of audio and video. “Flash” may mean the authoring environment, the player, or the application files. In particular embodiments, client system 530 may include a Flash client. The Flash client may be configured to receive and run Flash application or game object code from any suitable networking system (such as, for example, social networking system 520a or game networking system 520b). In particular embodiments, the Flash client may be run in a browser: client executed on client system 530. A player can interact with Flash objects using client system 530 and the Flash client. The Flash objects can represent a variety of in-game objects. Thus, the player may perform various in-game actions on various in-game objects by make various changes and updates to the associated Flash objects. In particular embodiments, in-game actions can be initiated by clicking or similarly interacting with a Flash object that represents a particular in-game object. For example, a player can interact with a Flash object to use, move, rotate, delete, attack, shoot, or harvest an in-game object. This disclosure contemplates performing any suitable in-game action by interacting with any suitable Flash object. In particular embodiments, when the player makes a change to a Flash object representing an in-game object, the client-executed game logic may update one or more game state parameters associated with the in-game object. To ensure synchronization between the Flash object shown to the player at client system 539, the Flash client may send the events that caused the game state changes to the in-game object to game networking system 520b. However, to expedite the processing and hence the speed of the overall gaming experience, the Flash client may collect a batch of some number of events or updates into a batch file. The number of events or updates may be determined by the Flash client dynamically or determined by game networking system 520b based on server loads or other factors. For example, client system 530 may send a batch file to game networking system 520b whenever 50 updates have been collected or after a threshold period of time, such as every minute,
[0073] As used herein, the term “application event data” may refer to any data relevant to a computer-implemented game application that may affect one or more game state parameters, including, for example and without limitation, changes to player data or metadata, changes to player social connections or contacts, player inputs to the game, and events generated by the game logic. In particular embodiments, each application datum may have a name and a value. The value of an application datum may change at any time in response to the game play of a player or in response to the game engine (e.g., based on the game logic). In particular embodiments, an application data update occurs when the value of a specific application datum is changed. In particular embodiments, each application event datum may include an action or event name and a value (such as an object identifier). Thus, each application datum may be represented as a name-value pair in the batch file. The batch file may include a collection of name-value pairs representing the application data that have been updated at client system 530. In particular embodiments, the batch file may be a text file and the name-value pairs may be in string format.

[0074] In particular embodiments, when a player plays an online game on client system 530, game networking system 520b may serialize all the game-related data, including, for example and without limitation, game states, game events, user inputs, for this particular user and this particular game into a BLOB and stores the BLOB in a database. The BLOB may be associated with an identifier that indicates that the BLOB contains the serialized game-related data for a particular player and a particular online game. In particular embodiments, while a player is not playing the online game, the corresponding BLOB may be stored in the database. This enables a player to stop playing the game at any time without losing the current state of the game the player is in. When a player resumes playing the game next time, game networking system 520b may retrieve the corresponding BLOB from the database to determine the most-recent values of the game-related data. In particular embodiments, while a player is playing the online game, game networking system 520b may also load the corresponding BLOB into a memory cache so that the game system may have faster access to the BLOB and the game-related data contained therein.
Systems and Methods

[0075] In particular embodiments, one or more described webpages may be associated with a networking system or networking service. However, alternate embodiments may have application to the retrieval and tendering of structured documents hosted by any type of network addressable resource or web site. Additionally, as used herein, a user may be an individual, a group, or an entity (such as a business or third party application).

[0076] Particular embodiments may operate in a wide area network environment, such as the Internet, including multiple network addressable systems. FIG. 6 illustrates an example network environment, in which various example embodiments may operate. Network cloud 660 generally represents one or more interconnected networks, over which the systems and hosts described herein can communicate. Network cloud 660 may include packet-based wide area networks (such as the Internet), private networks, wireless networks, satellite networks, cellular networks, paging networks, and the like. As FIG. 8 illustrates, particular embodiments may operate in a network environment comprising one or more networking systems, such as social networking system 620a, game networking system 620b, and one or more client systems 620. The components of social networking system 620a and game networking system 6M6b operate analogously, as such, hereinafter they may be referred to simply at networking system 620. Client systems 630 are operably connected to the network environment via a network service provider, a wireless carrier, or any other suitable means.

[0077] Networking system 620 is a network addressable system that, in various example embodiments, comprises one or more physical servers 622 and data stores 624. The one or more physical servers 622 are operably connected to computer network 660 via, by way of example, a set of routers and/or networking switches 626. In an example embodiment, the functionality hosted by the one or more physical servers 122 may include web or IOTP servers, FTP servers, as well as, without limitation, webpages and applications implemented using Common Gateway Interface (CGI) script, PHP Hyper-text Preprocessor (PHP), Active Server Pages (ASP), Hyper Text Markup Language (HTML), Extensible Markup Language (XML), Java, JavaScript, Asynchronous JavaScript and XML (AJAX), Flash, ActionScript and the like.

[0078] Physical servers 622 may host functionality directed to the operations of networking system 620. Hereinafter servers 622 may be referred to as server 622, although server 622 may
include numerous servers hosting, for example, networking system 620, as well as other content
distribution servers, data stores, and databases. Data store 6:24 may store content and data
relating to, and enabling, operation of networking system 620 as digital data objects. A data
object, in particular embodiments, is an item of digital information typically stored or embodied
in a data file, database, or record. Content objects may take many forms, including: text (e.g.,
ASCII, SGML, HTM...), images (e.g., jpeg, tif-arsd gil), graphics (vector-based or bitmap), audio,
video (e.g., mpegf or other multimedia, and combinations thereof. Content object data may also
include executable code objects (e.g., games executable within a browser window or frame),
podcasts, etc. Logically, data store 624 corresponds to one or more of a variety of separate and
integrated databases, such as relational databases and object-oriented databases, that maintain
information as an integrated collection of logically related records or files stored on one or more
physical systems. Structurally, data store 624 may generally include one or more of a large class
of data storage and management systems, in particular embodiments, data store 624 may be
implemented by any suitable physical system(s). Including components, such as one or more
database servers, mass storage media, media library systems, storage area networks, data storage
clouds, and the like. In one example embodiment, data store 624 includes one or more servers,
databases (e.g., MySQL), and/or data warehouses. Data store 624 may include data associated
with different networking system 620 users and/or client systems 630.

[0079] Client system 630 is generally a computer or computing device including
functionality for communicating (e.g., remotely) over a computer network. Client system 630
may be a desktop computer, laptop computer, personal digital assistant (FDA), in- or on-of-ear
navigation system, smart phone or other cellular or mobile phone, or mobile gaming device,
among other suitable computing devices. Client system 630 may execute one or more client
applications, such as a web browser (e.g., Microsoft Internet Explorer, Mozilla Firefox, Apple
Safari, Google Chrome, and Opera), to access and view content over a computer network. In
particular embodiments, the client applications allow a user of client system 630 to enter
addresses of specific network resources to be retrieved, such as resources hosted by networking
system 620. These addresses can be Uniform Resource Locators (URLs) and the like. In
addition, once a page or other resource has been retrieved, the client applications may provide
access to other pages or records when the user "clicks" on hyperlinks to other resources. By way
of example, such hyperlinks may be located within the webpages and provide an automated way for the user to enter the URL of another page and to retrieve that page.

[0080] A webpage or resource embedded within a webpage, which may itself include multiple embedded resources, may include data records, such as plain textual information, or more complex digitally encoded multimedia content, such as software programs or other code objects, graphics, images, audio signals, videos, and so forth. One prevalent markup language for creating webpages is the Hypertext Markup Language (HTML). Other common web browser-supported languages and technologies include the Extensible Markup Language (XML), the Extensible Hypertext Markup Language (XHTML), JavaScript, Flash, ActionScript, Cascading Style Sheet (CSS), and, frequently, Java. By way of example, HTML enables a page developer to create a structured document by denoting structural semantics for text and links, as well as images, web applications, and other objects that can be embedded within the page. Generally, a webpage may be delivered to a client as a static document; however, through the use of web elements embedded in the page, an interactive experience may be achieved with the page or a sequence of pages. During a user session at the client, the web browser interprets and displays the pages and associated resources received or retrieved from the website hosting the page, as well as, potentially, resources from other websites.

[0081] When a user at a client system 630 desires to view a particular webpage (hereinafter also referred to as target structured document) issued by networking system 620, the user’s web browser, or other document rendering engine or suitable client application, formulates and transmits a request to networking system 620. The request generally includes a URL or other document identifier as well as metadata or other information. By way of example, the request may include information identifying the user, such as a user ID, as well as information identifying or characterising: the web browser or operating system running on the user’s client computing device 639. The request may also include location information identifying a geographic location of the user’s client system or a logical network location of the user’s client system. The request may also include a timestamp identifying when the request was transmitted.

[0082] Although the example network environment described above and illustrated in FIG. 6 described with respect to social networking system 620a and game networking system 620b, this disclosure encompasses any suitable network environment using any suitable systems. As an
example and not by way of limitation, the network environment may include online media systems, online reviewing systems, online search engines, online advertising systems, or any combination of two or more such systems.

[0083] FIG. 9 illustrates an example computing system architecture, which may be used to implement a server 622 or a client system §30. In one embodiment, hardware system 700 comprises a processor 702, a cache memory 704, and one or more executable modules and drivers, stored on a tangible computer readable medium, directed to the functions described herein. Additionally, hardware system 700 may include a high performance input/output (I/O) bus 706 and a standard I/O bus 70S. A host bridge 710 may couple processor 702 to high performance I/O bus 706, whereas I/O bus bridge 712 couples the two buses 706 and 708 to each other. A system memory 714 and one or more network/communication Interfaces 716 may couple to bus 706, Hardware-system 700 may further include video memory (not shown) and a display device coupled to the video memory. Mass storage 718 and I/O ports 720 may couple to bus 708, Hardware system 700 may optionally include a keyboard, a pointing device, and a display device (not shown) coupled to bus 708. Collectively, these elements are intended to represent a broad category of computer hardware systems. Including but not limited to general purpose computer systems based on the x86-compatible processors manufactured by Intel Corporation of Santa Clara, California, and the x86-compatible processors manufactured by Advanced Micro Devices (AMD), Inc., of Sunnyvale, California, as well as any other suitable processor.

[0084] The elements of hardware system 700 are described in greater detail below. In particular, network interface 716 provides communication between hardware system 700 and any of a wide range of networks, such as an Ethernet (e.g., IEEE 802.3) network, a backplane, etc. Mass storage 718 provides permanent storage for the data and programming instructions to perform the above-described functions implemented in servers 422, whereas system memory 714 (e.g., DRAM) provides temporary storage for the data and programming instructions when executed by processor 702. I/O ports 720 arc one or more serial and/or parallel communication ports that provide communication between additional peripheral devices, which may be coupled to hardware system 700,
Hardware system 700 may include a variety of system architectures and various components of hardware system 700 may be rearranged. For example, cache 704 may be on-chip with processor 702. Alternatively, cache 704 and processor 702 may be packed together as a "processor module," with processor 702 being referred to as the "processor core." Furthermore, certain embodiments of the present disclosure may not require nor include all of the above components. For example, the peripheral devices shown coupled to standard I/O bus 708 may couple to high performance I/O bus 796. In addition, in some embodiments, only a single bus may exist, with the components of hardware system 700 being coupled to the single bus. Furthermore, hardware system 700 may include additional components, such as additional processor storage devices, or memories.

An operating system manages and controls the operation of hardware system 700, including the input and output of data to and from software applications (not shown). The operating system provides an interface between the software applications being executed on the system and the hardware components of the system. Any suitable operating system may be used, such as the LINUX Operating System, the Apple Macintosh Operating System, available from Apple Computer Inc. of Cupertino, Calif, UNIX operating systems, Microsoft (r) Windows(r) operating systems, BSD operating systems, and the like. Of course, other embodiments are possible. For example, the functions described herein may be implemented in firmware or on an application-specific integrated circuit.

Furthermore, the above-described elements and operations can be comprised of instructions that are stored on non-transitory storage media. The instructions can be retrieved and executed by a processing system. Some examples of instructions are software, program code, and firmware. Some examples of non-transitory storage media are memory devices, tape, disks, integrated circuits, and servers. The instructions are operational when executed by the processing system to direct the processing system to operate in accord with the disclosure. The term "processing system" refers to a single processing device or a group of inter-operational processing devices. Some examples of processing devices are integrated circuits and logic circuitry. Those skilled in the art are familiar with Instructions, computers, and storage media.
Miscellaneous

One or more feature* from any embodiment may be combined with one or more features of any other embodiment without departing from the scope of the disclosure.

A recitation of "a", "an," or "the" is intended to mean "one or more" unless specifically indicated to the contrary. In addition, it is to be understood that functional operations, such as "awarding", "locating", "permitting" and the like, are executed by game application logic that accesses, and/or causes changes to, various data attribute values maintained in a database or other memory.

The present disclosure encompasses all changes, substitutions, variations, alterations, and modifications to the example embodiments herein that a person having ordinary skill in the art would comprehend. Similarly, where appropriate, the appended claims encompass all changes, substitutions, variations, alterations, and modifications to the example embodiments herein that a person having ordinary skill in the art would comprehend.

For example, the methods, game features and game mechanics described herein may be implemented using hardware components, software components, and/or any combination thereof. By way of example, while embodiments of the present disclosure have been described as operating in connection with a networking website, various embodiments of the present disclosure can be used in connection with any communications facility that supports web applications. Furthermore, in some embodiments the term "web service" and "website" may be used interchangeably and additionally may refer to a custom or generalized API on a device, such as a mobile device (e.g., cellular phone, smart phone, personal GPS, personal digital assistance, personal gaming device, etc), that makes API calls directly to a server. Still further, while the embodiments described above operate with business-related virtual objects (such as stores and restaurants), the invention can be applied to any in-game asset around which a harvest mechanic is implemented, such as a virtual stove, a plot of land, and the-like. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense. It will however, be evident that various modifications and changes may be made thereunto without departing from the broader spirit and scope of the disclosure as set forth in the claims and that the disclosure is intended to cover all modifications and equivalents within the scope of the following claims.
CLAIMS:

What is claimed is:

1. A method, comprising:
   permitting, by one or more computing devices, a first player to access a virtual game environment of a second player, the virtual game environment corresponding to a computer-implemented game;
   responsive to inputs of the first player, loesting, by the one or more computing devices, a franchise object within the virtual game environment of the second player;
   executing, by the one or more computing devices, a harvest mechanic cycle in connection with the franchise object of the first player; and
   distributing an in-game award to the first and second player upon completion of the harvest mechanic cycle.

2. The method of claim 1 wherein the computer-implemented game permits either the first or second player to initiate the harvest mechanic.

1. The method of claim 2 further comprising
   responsive to inputs of the first player, initiating a second harvest mechanic cycle in connection with the franchise object;
   allocating a second in-game award to the first player based on initiation of the second harvest mechanic.

4. The method of claim 1 wherein the computer-implemented game permits only the second player to initiate a collection phase of the harvest mechanic.

5. The method of claim 1 wherein the in-game award is an amount of virtual currency.

6. The method of claim 1 farther comprising
responsive to inputs of the first player, locating, by the one or more computing devices, a
franchise headquarters object within the virtual game environment of the first player, wherein the
franchise headquarters object is associated with the franchise object.

7. The method of claim 6 further comprising modulating one or more attributes of the franchise
headquarters object based on a number of franchise objects associated with the franchise
headquarters object.

8. The method of claim 7 further comprising modulating one or more operational parameters of
an in-game object based on a proximity of the in-game object to the irandehise headquarters
object and at least one of the one or more attributes of the franchise headquarters object.

9. The method of claim 6 wherein the irandehise headquarters object is an activatable element
operative to cause a franchise management interface to launch upon user activation.

10. The method of claim 1 wherein the first player and the second player are connected within a
social graph.

11. A non-transitory, computer readable medium comprising instructions operative, when
executed, to cause one or more processors to:

   permit a first player to access a virtual game environment of a second player, the virtual
game environment corresponding to a computer-implemented game;

   responsive to inputs of the first player, locate a franchise object within the virtual game
environment of the second player;

   execute, by the one or more computing devices, a harvest mechanic cycle in connection
with the franchise object of the first player; and

   distribute an in-game award to the first and second player upon completion of the harvest
mechanic cycle.
12. The computer readable medium of claim 11 wherein the computer-implemented game permits either the first or second player to initiate the harvest mechanic.

13. The computer readable medium of claim 12 further comprising instructions operative to cause the one or more processors to:
   - responsive to inputs of the first player, initiate a second harvest mechanic cycle in connection with the franchise object;
   - allocate a second in-game award to the first player based on initiation of the second harvest mechanic.

14. The computer readable medium of claim 11 further comprising instructions operative to cause the one or more processors to:
   - permit only the second player to initiate a collection phase of the harvest mechanic.

15. The computer readable medium of claim 11 wherein the in-game award is an amount of virtual currency.

16. The computer readable medium of claim 11 further comprising instructions operative to cause the one or more processors to:
   - responsive to inputs of the first player, locate, by the one or more computing devices, a franchise headquarters object within the virtual game environment of the first player, wherein the franchise headquarters object is associated with the franchise object.

17. The computer readable medium of claim 16 further comprising instructions operative to cause the one or more processors to:
   - modulate one or more attributes of the franchise headquarters object based on a number of franchise objects associated with the franchise headquarters object.

18. The computer readable medium of claim 17 further comprising instructions operative to cause the one or more processors to:
   - modulate one or more operational parameters of an in-game
object based on a proximity of the in-game object to the franchise headquarters object and at least one of the one or more attributes of the franchise headquarters object.

19. The computer readable medium of claim 16 wherein the franchise headquarters object is an activatable element operative to cause a franchise management interface to launch upon user activation.

20. The computer readable medium of claim 11 wherein the first player and the second player are connected within a social graph.
FIG. 2
Thanks for visiting! I've saved a great spot for your business.
FIG. 5

FIG. 7
INTERNATIONAL SEARCH REPORT

A. CLASSIFICATION OF SUBJECT MATTER

IPC(8) - A63F 9/24 (201.01)

USPC - 463/42

According to International Patent Classification (IPC) or to both national classification systems

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC(8) - A63F 9/24, 13/00; G06F 17/00, 19/00 (201.01)
USPC - 273/461; 463/42; 705/319

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

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>X</td>
<td></td>
<td>1-4, 6-9, 11-14, 16-19</td>
</tr>
<tr>
<td>Y</td>
<td>US 2009/0144148 A1 (JUNG et al) 04 June 2009 (04.06.2009) entire document</td>
<td>5, 15</td>
</tr>
</tbody>
</table>

Further documents are listed in the continuation of Box C.

- Special categories of cited documents:
  - "A" document defining the general state of the art which is not considered to be of particular relevance
  - "E" earlier application or patent but published on or after the international filing date
  - "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
  - "O" document referring to an oral disclosure, use, exhibition or other means
  - "P" document published prior to the international filing date but later than the priority date claimed
  - "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
  - "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
  - "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
  - "&" document member of the same patent family

Date of the actual completion of the international search
13 December 2011

Date of mailing of the international search report
23 DEC 2011

PCT Helpdesk: 571-272-4300
PCT OSP: 571-272-7774