METHODS, COMPUTER-PROGRAM PRODUCTS, AND SYSTEMS FOR IMPLEMENTING A COLONIAL RELATIONSHIP IN A MULTIPLAYER ONLINE GAME

CROSS-REFERENCE TO RELATED APPLICATIONS

This application claims priority to U.S. Provisional Patent Application Serial No. 61/298,425, filed January 26, 2010. The entire contents of this application are hereby incorporated by reference herein.

BACKGROUND

The ubiquity of the Internet and increasing bandwidth and computing power allow for increasingly collaborative online gaming experiences. For example, massively multiplayer online games (MMOGs) allow hundreds or thousands of users to interact in a persistent world.

Although various games allow users to wage virtual battles against each other, a persistent challenge exists in maintaining the loser of the battle's interest in the game. For example, if the loser's character is killed, the user may need to restart as a new player and may instead explore competing games.

Accordingly, there is a need for new methods, systems, and computer-readable media capable of maintaining a user's interest after losing a battle.

SUMMARY OF THE INVENTION

One aspect of the invention provides a method for implementing a colonial relationship in a multiplayer online game. The method includes: receiving instructions from a first entity to attack a second entity; determining whether the first entity's attack is successful; and if the first entity's attack is successful, establishing a colonial relationship between the second entity and the first entity, wherein a portion of production of the second entity is transferred to the first entity.

This aspect of the invention can have a variety of embodiments. The method can include receiving instructions from the second entity to institute an uprising; and determining whether the second entity's uprising is successful; and if the uprising is successful, terminating the colonial relationship.

The method can include: receiving instructions from the first entity to terminate the colonial relationship and terminating the colonial relationship.

The method can include: receiving instructions from a third entity to attack the second entity; determining whether the third entity's attack is successful; and if the third entity's attack is successful, establishing a colonial relationship between the second entity and the third entity, wherein a portion of production of the second entity is transferred to the third entity.

The method can include: receiving instructions from a fourth entity to attack the first entity; determining whether the fourth entity's attack is successful; and if the fourth entity's attack is successful: establishing a colonial relationship between the fourth entity and the first entity, wherein a portion of production of the first entity is transferred to the fourth entity and terminating the colonial relationship between the first entity and the second entity.

The method can include: receiving instructions from the second entity to offer a reward for assistance; transmitting the offer to a plurality of entities; receiving instructions from a fifth entity to assist the second entity in an uprising; determining whether the second entity's uprising is successful; and if the uprising is successful: terminating the colonial relationship and transferring the reward from the second entity to the fifth entity.

The method can include determining whether the first entity is eligible to enter a colonial relationship. The determining step can include assessing whether the first entity holds a required credential. The required credential can be a title. The determining step can include calculating whether the number of colonial relationships held by the first entity equals or exceeds a defined threshold. The determining step can include assessing whether the first entity is a beginner, on vacation mode, or in a truce.

The method can include determining whether the second entity is eligible to enter a colonial relationship. The determining step can include assessing whether the second entity is a beginner, on vacation mode, an ally of the first entity, or a historic entity.

The method can include determining whether the first entity and the second entity are commonly-owned.

The first entity can designate the portion of production transferred from the second entity.

The method can be a computer-implemented method.

Another aspect of the invention provides a computer program product including computer-usable medium having control logic stored therein for causing a computer to implement a method for implementing a colonial relationship in a multiplayer online game. The control logic includes: first computer-readable program code for causing the computer to receive instructions from a first entity to attack a second entity; second computer-readable program code for causing the computer to determine whether the first entity's attack is successful; and third computer-readable program code for causing the computer to, if the first entity's attack is successful, establish a colonial relationship between the second entity and the first entity, wherein a portion of production of the second entity is transferred to the first entity.

This aspect of the invention can have a variety of embodiments. The control logic can further include: fourth computer-readable program code for causing the computer to receive instructions from the second entity to institute an uprising; fifth computer-readable program code for causing the computer to determine whether the second entity's uprising is successful; and sixth computer-readable program code for causing the computer to, if the uprising is successful, terminate the colonial relationship.

The control logic can further include: seventh computer-readable program code for causing the computer to receive instructions from the first entity to terminate the colonial relationship; and eighth computer-readable program code for causing the computer to terminate the colonial relationship.

The control logic can further include: ninth computer-readable program code for causing the computer to receive instructions from a third entity to attack the second entity; tenth computer-readable program code for causing the computer to determine whether the third entity's attack is successful; and eleventh computer-readable program code for causing the computer to, if the third entity's attack is successful, establish a colonial relationship between the second entity and the third entity, wherein a portion of production of the second entity is transferred to the third entity.

The control logic can further include: twelfth computer-readable program code for causing the computer to receive instructions from a fourth entity to attack the first entity; thirteenth computer-readable program code for causing the computer to determine whether the fourth entity's attack is successful; and fourteenth computer-readable program code for causing the computer to, if the fourth entity's attack is

successful: establish a colonial relationship between the fourth entity and the first entity, wherein a portion of production of the first entity is transferred to the fourth entity and terminate the colonial relationship between the first entity and the second entity.

The control logic can further include: fifteenth computer-readable program code for causing the computer to receive instructions from the second entity to offer a reward for assistance; sixteenth computer-readable program code for causing the computer to transmit the offer to a plurality of entities; seventeenth computer-readable program code for causing the computer to receive instructions from a fifth entity to assist the second entity in an uprising; eighteenth computer-readable program code for causing the computer to determine whether the second entity's uprising is successful; and nineteenth computer-readable program code for causing the computer to, if the uprising is successful: terminate the colonial relationship and transfer the reward from the second entity to the fifth entity.

The computer-usable medium can be non-transitory and tangible.

Another aspect of the invention provides a system for implementing a colonial relationship in a multiplayer online game. The system includes a server configured to: receive instructions from a first entity to attack a second entity; determine whether the first entity's attack is successful; and if the first entity's attack is successful, establishing a colonial relationship between the second entity and the first entity, wherein a portion of production of the second entity is transferred to the first entity.

FIGURES

For a fuller understanding of the nature and desired objects of the present invention, reference is made to the following detailed description taken in conjunction with the figure wherein:

FIG. 1 depicts an exemplary game architecture suitable for a massively multiplayer online game; and

FIGS. 2A-2E depict a method of implementing a colonial relationship in a multiplayer online game according to embodiments of the invention.

DEFINITIONS

The instant invention is most clearly understood with reference to the following definitions:

As used in the specification and claims, the singular form "a," "an," and "the" include plural references unless the context clearly dictates otherwise.

A computer-usable medium shall be understood to mean any article of manufacture that contains data that can be read by a computer or a carrier wave signal carrying data that can be read by a computer. Such computer-usable media includes, but is not limited to, magnetic media, such as a floppy disk, a flexible disk, a hard disk, reel-to-reel tape, cartridge tape, cassette tape or cards; optical media such as CD-ROM and writeable compact disc; magneto-optical media in disc, tape or card form; paper media, such as punched cards and paper tape; or on carrier wave signal received through a network, wireless network or modem, including radio-frequency signals and infrared signals. Such computer-readable medium can be non-transitory and/or tangible.

DESCRIPTION OF THE INVENTION

Game Architecture

Aspects of the invention can be applied to a variety of games, particularly online games such as MMOGs. An exemplary game architecture 100 suitable for an MMOG is depicted in FIG. 1.

At the highest level, architecture 100 includes clients 102 and servers 104 connected via a network 106 such as the Internet.

Clients 102 can include laptop computers 102a, desktop computers 102b, handheld devices 102c (*e.g.*, a tablet computer, personal digital assistant, cellular telephone, smart phone), and the like now known and later developed. Clients 102 can include display(s) appreciated by those of ordinary skill in the pertinent art. The displays can include any of a number of devices known to those skilled in the art for displaying images responsive to outputs signals from the clients 102. Such devices include, but are not limited to, cathode ray tubes (CRTs), liquid crystal displays (LCDs), plasma screens and the like. Although a simplified diagram is illustrated in FIG. 1, such illustration shall not be construed as limiting the present invention to the illustrated embodiment. It should be recognized that the signals output from the clients 104 can originate from any of a number of devices including PCI or AGP

video boards or cards mounted within the housing of the clients 102 that are operably coupled to the microprocessors and the displays thereof.

In some embodiments, clients 102 include one or more video game consoles now known or later developed. Such consoles can include Ethernet port to connect to server 104 via network 106, video output ports for displaying images on a display device such as a television, input devices such as compact disc (CD) or digital video disc (DVD) drives to load games, local memory to save data, and input ports to receive input from controllers or keyboards. Examples of suitable video game consoles include the NINTENDO® WII® console available from Nintendo of America Inc. of Redmond, Washington; the SONY® PLAYSTATION® console available from Kabushiki Kaisha Sony Corporation of Tokyo, Japan; the MICROSOFT® XBOX® console available from Microsoft Corporation of Redmond, Washington; and the like.

Clients 102 can access server 104 through proprietary or commercially-available software. For example, clients 102 can utilize an Internet browser to access a Hypertext Transfer Protocol (HTTP) interface provided by server 104. Suitable Internet browsers include the INTERNET EXPLORER® browser available from Microsoft Corporation of Redmond, Washington; the FIREFOX® browser available from the Mozilla Foundation of Mountain View, California; the OPERA® browser available from Opera Software AS of Oslo, Norway; or the CHROME™ browser available from Google Inc. of Menlo Park, California. Alternatively, clients 102 can interact with system 104 via specially-programmed software such as an application installed on clients 102. In still another embodiment, the client 102 and server 104 can communicate through a multimedia platform such as ADOBE® FLASH®, available from Adobe Systems Incorporated of San Jose, California.

In order to enable scaling and reduce lag, MMOGs often utilize multiple servers 104 as depicted in FIG. 1. Each server 104 can implement one or more instances of the MMOG, which are known as "shards." Thus, a first client 102a and a second client 102b can both play on the same shard implemented on a first server 104a. Clients 102a and 102b can interact with each other, but will have limited, if any, interaction with a third client 102c playing on another shard implemented on the second server 104b. Advantageously, first server 104a does not need to be concerned with the activities of the third client 104b.

System 100 can include a load balancer module 108 to route traffic between clients 102 and servers 104. Various schemes can be used as are familiar to those of skill in the art. For example, when new clients 102 access system 100, load balancer 108 can direct client 102 to a newly opened server 104c until the server 104c reaches capacity. At that point, load balancer 108 can direct traffic to yet another server 104. Requests from existing clients 102 can be routed to the server 104 that the client 102 previously accessed.

All traffic need not flow through load balancer 108. For example, traffic can flow directly between clients 102 and servers 104 (via network 106) once initial contact is made between clients 102 and servers 104 as mediated by the load balancer 108.

Server 104 can communicate with a database 110 through a database management system (DBMS) 112. A DBMS 112 is imposed upon the data in database 110 to form a logical and structured organization of the data. A DBMS 110 lies between the physical storage of data and the users and handles the interaction between the two. Examples of DBMSes include DB2® and INFORMIX® DBMSes both available from IBM Corp. of Armonk, New York; MICROSOFT JET® and MICROSOFT SQL SERVER® DBMSes both available from the Microsoft Corporation of Redmond, Washington; MYSQL® DBMS available from the MySQL Ltd. Co. of Stockholm, Sweden; ORACLE® DBMS available from Oracle Int'l Corp of Redwood City, California; and SYBASE® DBMS available from Sybase, Inc. of Dublin, California.

In some games, a plurality of users each control one or more entities, which may vary depending on the genre of the game. For example, in the EVONY® MMOG, available from Evony, LLC of Wilmington, Delaware, users control cities. Other exemplary entities include nations, tribes, towns, villages, and the like. Users can utilize their entities to produce resources such as currency, wood, metal, food, and the like. Users can also interact with other entities to wage war and peace. For example, entities can marshal armies and obtain weapons that are used in virtual battles.

Method for Implementing a Colonial Relationship

Referring now to FIGS. 2A-2E, a method 200 of implementing a colonial relationship is provided. The method 200 can be applied to a variety of games including multiplayer online games and massively multiplayer online games.

In step S202 depicted in FIG. 2A, server 104 receives instructions from a first entity to attack a second entity. Such instructions can be transmitted in a variety of methods. For example, the user controlling the first entity can utilize a graphical user interface to designate the second entity and declare war.

In step S204, server 104 determines whether the first entity is eligible to eligible to enter a colonial relationship. For example, the first entity can be required to hold a defined credential (*e.g.*, a title) in order to enter a colonial relationship. Additionally or alternatively, the first entity can be limited to a defined number of colonial relationships. In still another embodiment, the first entity can be prohibited from entering a colonial relationship if the first entity is a beginner, on vacation mode, in a truce, or the like.

In step S206, server 104 determines whether the first entity is eligible to eligible to enter a colonial relationship. For example, the second entity can be required to hold a defined credential (*e.g.*, a title) in order to enter a colonial relationship. In another embodiment, the second entity can be prohibited from entering a colonial relationship if the first entity is a beginner, on vacation model, an ally of the first entity, a historic entity, or the like. Moreover, a colonial relationship can be prohibited in the first entity and the second entity are commonly owned.

In step S208, server 104 determines whether the first entity's attack is successful. This determination can be made in accordance with the rules of particular games. For example, the determination can reflect the strength, abilities, and/or quantity of troops deployed by the first entity to attack and/or the strength, abilities, and/or quantity of troops deployed by the second entity to defend. Additionally, an element of chance can be factored into the decision (*e.g.*, by dice roll or other random number).

In step S210, if the first entity's attack is successful and both the first and the second entities are eligible to enter a colonial relationship, a colonial relationship is established between the second entity and the first entity such that a portion of production of the second entity is transferred to the first entity in step S212. In such a colonial relationship, the occupying party receiving tributes can be referred to as the master or the suzerain and the occupied party paying the tribute can be referred to as the slave or protectorate.

Otherwise, the method can be terminated in step S214.

Tributes can be pre-defined by the game or can be set by the first entity. Tributes can be one-time tributes or can be assessed on a regular basis (*e.g.*, every turn, every hour, every day, every week, and the like).

Colonial relationships can be modified in various ways as discussed below.

Referring now to FIG. 2B, in step S216, the server 104 can receive instructions from the second entity to institute an uprising. In step S218, the server 104 determines if the uprising is successful. This determination can be made in the same or similar manner as in step S208.

There can be several reasons why an uprising is successful even though the first entity's attack was recently successful. In some situations, the first entity may sustain significant losses during the attack. In other situations, the first entity may withdraw troops from the second entity after the colonial relationship is established. Furthermore, the second entity can continue develop resources necessary for a successful uprising during the colonial relationship. Additionally, the rules concerning attacks and uprisings can differ in significant ways. For example, ordinary citizens of the second entity can, in some embodiments, be available for use in an uprising, but may not be available for defense against an attack.

If the uprising is unsuccessful, the colonial relationship is maintained and tributes continue to be transferred from the second entity to the first entity in step S212 as discussed herein. If the uprising is successful, the colonial relationship is terminated in step S220.

Still referring to FIG. 2B, server 104 can receive instructions from the first entity in terminate the colonial relationship in step S222 and terminate the colonial relationship in step S220. The first entity can desire to terminate the colonial relationship for a variety of reasons. For example, the first entity may have entered the maximum number of colonial relationships allowed and may seek to cull colonial relationships with lower performing entities in order to pursue colonial relationships with higher performing entities.

Referring now to FIG. 2C, server 104 can receive instructions from a third entity to attack the second entity in step S224. Server 104 determines in step S226 whether the attack is successful in the same or similar manner as discussed herein. In some embodiments, troops of the first entity stationed in the second entity can be used to defend the second entity. If the attack is successful, a colonial relationship is established between the third entity and the second entity in step S228 and the

colonial relationship between the first entity and the second entity is terminated in step S220. If the attack is unsuccessful, the colonial relationship is maintained in step S212.

Referring now to FIG. 2D, server 104 can receive instructions from a fourth entity to attack the first entity in step S230. Server 104 determines in step S232 whether the attack is successful in the same or similar manner as discussed herein. If the attack is successful, the server 104 establishes a colonial relationship between the fourth entity and the first entity in step S234 and terminates the colonial relationship between the first entity and the second entity in step S220. If the attack is unsuccessful, the colonial relationship is maintained in step S212.

Referring now to FIG. 2E, server 104 can receive instructions to offer a reward for assistance in ending a colonial relationship in step S236. For example, the second entity might offer 1,000 units of currency in exchange for 50 soldiers. In step S238, server 104 transmits this offer to other entities. The offer can be structured such that only one entity can accept or can be structured such that multiple entities can accept (*e.g.*, until such time as the requested level of assistance is provided). In step S240, server 104 receives instructions from another entity to provide assistance.

In step S242, the server 104 determines whether the uprising is successful. If the uprising is successful, the server 104 transfers the reward from the second entity to the assisting entity or entities in step S244 and terminates the colonial relationship between the first entity and the second entity in step S220. If the attack is unsuccessful, the colonial relationship is maintained in step S212.

Implementation in Hardware and/or Software

The systems and methods herein can be implemented on general-purpose or specially-programmed hardware or software. For example, the methods can be implemented by a computer-usable medium. The computer-usable medium can be non-transitory and/or tangible. For example, the computer-usable medium can be volatile memory (*e.g.*, random access memory and the like) or non-volatile memory (*e.g.*, read-only memory, hard disks, floppy discs, magnetic tape, optical discs, paper table, punch cards, and the like).

INCORPORATION BY REFERENCE

All patents, published patent applications, and other references disclosed herein are hereby expressly incorporated by reference in their entireties by reference.

EQUIVALENTS

The functions of several elements may, in alternative embodiments, be carried out by fewer elements, or a single element. Similarly, in some embodiments, any functional element may perform fewer, or different, operations than those described with respect to the illustrated embodiment. Also, functional elements (*e.g.*, modules, databases, computers, clients, servers and the like) shown as distinct for purposes of illustration may be incorporated within other functional elements, separated in different hardware or distributed in a particular implementation.

While certain embodiments according to the invention have been described, the invention is not limited to just the described embodiments. Various changes and/or modifications can be made to any of the described embodiments without departing from the spirit or scope of the invention. Also, various combinations of elements, steps, features, and/or aspects of the described embodiments are possible and contemplated even if such combinations are not expressly identified herein.

CLAIMS

1. A method for implementing a colonial relationship in a multiplayer online game, the method comprising:

receiving instructions from a first entity to attack a second entity;

determining whether the first entity's attack is successful; and

if the first entity's attack is successful, establishing a colonial relationship between the second entity and the first entity, wherein a portion of production of the second entity is transferred to the first entity.

2. The method of claim 1, further comprising:

receiving instructions from the second entity to institute an uprising; and

determining whether the second entity's uprising is successful; and

if the uprising is successful, terminating the colonial relationship.

3. The method of claim 1, further comprising:

receiving instructions from the first entity to terminate the colonial relationship; and

terminating the colonial relationship.

4. The method of claim 1, further comprising:

receiving instructions from a third entity to attack the second entity;

determining whether the third entity's attack is successful; and

if the third entity's attack is successful, establishing a colonial relationship between the second entity and the third entity, wherein a portion of production of the second entity is transferred to the third entity.

5. The method of claim 1, further comprising:

receiving instructions from a fourth entity to attack the first entity;

determining whether the fourth entity's attack is successful; and

if the fourth entity's attack is successful:

establishing a colonial relationship between the fourth entity and the first entity, wherein a portion of production of the first entity is transferred to the fourth entity; and

terminating the colonial relationship between the first entity and the second entity.

6. The method of claim 1, further comprising:

receiving instructions from the second entity to offer a reward for assistance;

transmitting the offer to a plurality of entities;

receiving instructions from a fifth entity to assist the second entity in an uprising;

determining whether the second entity's uprising is successful; and

if the uprising is successful:

terminating the colonial relationship; and

transferring the reward from the second entity to the fifth entity.

7. The method of claim 1, further comprising:

determining whether the first entity is eligible to enter a colonial relationship.

8. The method of claim 7, wherein the determining step includes assessing whether the first entity holds a required credential.

9. The method of claim 7, wherein the required credential is a title.

10. The method of claim 7, wherein the determining step includes calculating whether the number of colonial relationships held by the first entity equals or exceeds a defined threshold.

11. The method of claim 7, wherein the determining step includes assessing whether the first entity is a beginner, on vacation mode, or in a truce.

12. The method of claim 1, further comprising:

determining whether the second entity is eligible to enter a colonial relationship.

13. The method of claim 12, wherein the determining step includes assessing whether the second entity is a beginner, on vacation mode, an ally of the first entity, or a historic entity.

14. The method of claim 1, further comprising:

determining whether the first entity and the second entity are commonly-owned.

15. The method of claim 1, wherein the first entity can designate the portion of production transferred from the second entity.

16. The method of claim 1, wherein the method is a computer-implemented method.

17. A computer program product comprising computer-usable medium having control logic stored therein for causing a computer to implement a method for implementing a colonial relationship in a multiplayer online game, the control logic comprising:

first computer-readable program code for causing the computer to receive instructions from a first entity to attack a second entity;

second computer-readable program code for causing the computer to determine whether the first entity's attack is successful; and

third computer-readable program code for causing the computer to, if the first entity's attack is successful, establish a colonial relationship between the second entity and the first entity, wherein a portion of production of the second entity is transferred to the first entity.

18. The computer program product of claim 17, wherein the control logic further comprises:

fourth computer-readable program code for causing the computer to receive instructions from the second entity to institute an uprising;

fifth computer-readable program code for causing the computer to determine whether the second entity's uprising is successful; and

sixth computer-readable program code for causing the computer to, if the uprising is successful, terminate the colonial relationship.

19. The computer program product of claim 17, wherein the control logic further comprises:

seventh computer-readable program code for causing the computer to receive instructions from the first entity to terminate the colonial relationship; and

eighth computer-readable program code for causing the computer to terminate the colonial relationship.

20. The computer program product of claim 17, wherein the control logic further comprises:

ninth computer-readable program code for causing the computer to receive instructions from a third entity to attack the second entity;

tenth computer-readable program code for causing the computer to determine whether the third entity's attack is successful; and

eleventh computer-readable program code for causing the computer to, if the third entity's attack is successful, establish a colonial relationship between the second entity and the third entity, wherein a portion of production of the second entity is transferred to the third entity.

21. The computer program product of claim 17, wherein the control logic further comprises:

twelfth computer-readable program code for causing the computer to receive instructions from a fourth entity to attack the first entity;

thirteenth computer-readable program code for causing the computer to determine whether the fourth entity's attack is successful; and

fourteenth computer-readable program code for causing the computer to, if the fourth entity's attack is successful:

establish a colonial relationship between the fourth entity and the first entity, wherein a portion of production of the first entity is transferred to the fourth entity; and

terminate the colonial relationship between the first entity and the second entity.

22. The computer program product of claim 17, wherein the control logic further comprises:

fifteenth computer-readable program code for causing the computer to receive instructions from the second entity to offer a reward for assistance;

sixteenth computer-readable program code for causing the computer to transmit the offer to a plurality of entities;

seventeenth computer-readable program code for causing the computer to receive instructions from a fifth entity to assist the second entity in an uprising;

eighteenth computer-readable program code for causing the computer to determine whether the second entity's uprising is successful; and

nineteenth computer-readable program code for causing the computer to, if the uprising is successful:

terminate the colonial relationship; and

transfer the reward from the second entity to the fifth entity.

23. The computer-program product of claim 17, wherein the computer-usable medium is non-transitory and tangible.

24. A system for implementing a colonial relationship in a multiplayer online game, the system comprising:

a server configured to:

receive instructions from a first entity to attack a second entity;

determine whether the first entity's attack is successful; and

if the first entity's attack is successful, establishing a colonial relationship between the second entity and the first entity, wherein a portion of production of the second entity is transferred to the first entity.