US 20090181777A1

(19) **United States**

(12) **Patent Application Publication**

Christiani et al.

(10) **Pub. No.: US 2009/0181777 A1**

(43) **Pub. Date: Jul. 16, 2009**

(54) **NETWORK COMPUTER GAME LINKED TO REAL-TIME FINANCIAL DATA**

(76) Inventors: **Michael Gerard Christiani**, McLean, VA (US); **Zachary Dale Buckner**, Charlottesville, VA (US); **John Fletcher Elder, IV**, Charlottesville, VA (US); **John Anthony Dimeo**, Culpeper, VA (US)

Correspondence Address:
W. EDWARD RAMAGE
COMMERCE CENTER SUITE 1000, 211 COMMERCE ST
NASHVILLE, TN 37201 (US)

(21) Appl. No.: **12/353,241**

(22) Filed: **Jan. 13, 2009**

Related U.S. Application Data

(60) Provisional application No. 61/020,911, filed on Jan. 14, 2008.

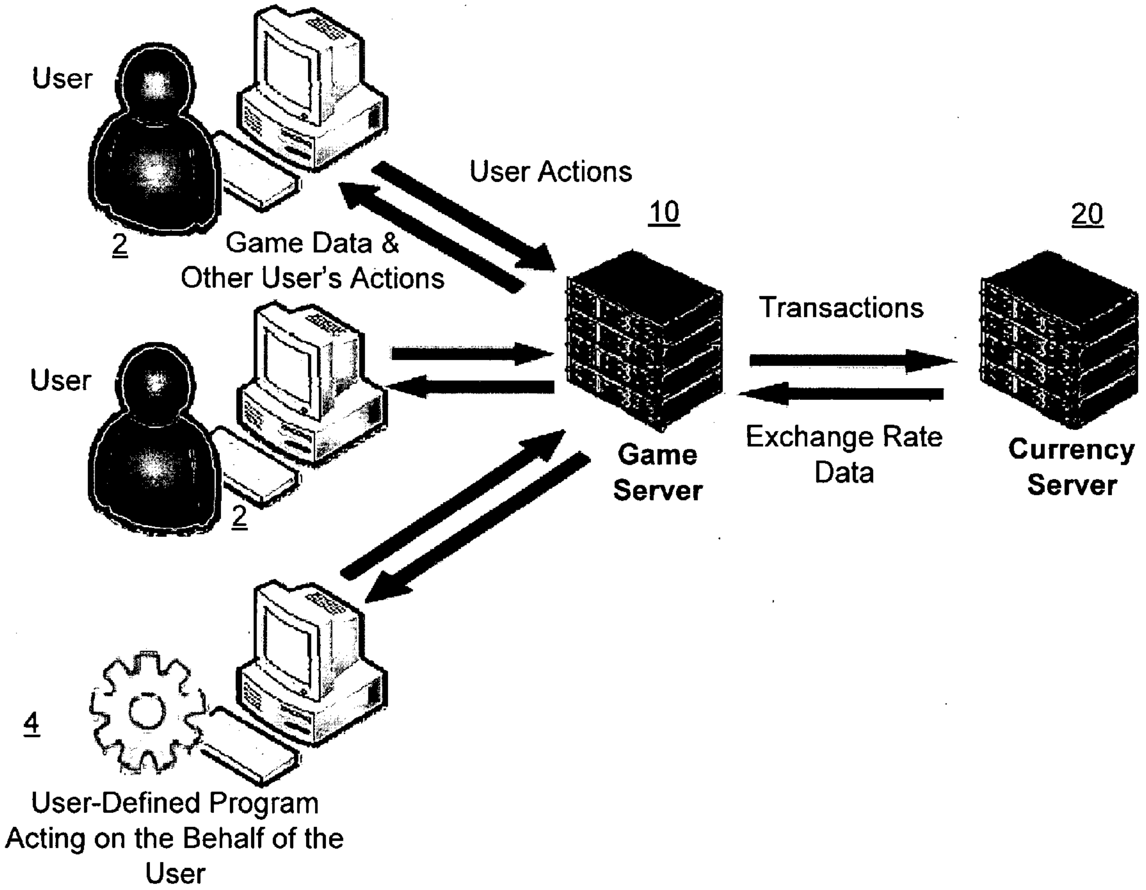
Publication Classification

(51) **Int. Cl.**
A63F 9/24 (2006.01)

(52) **U.S. Cl.** **463/42**

(57) **ABSTRACT**

A network-based computer game that both uses real-time financial data to affect game state and uses participants' actions to determine and engage in financial transactions. A game server receives financial data on market conditions (such as currency markets) from a financial server and regularly synchronizes user's game environments to match the financial data. The game server propagates financial data to the game or games, coordinates communication (if any) between the games, and can initiate the opening and closing of positions in the financial market based on players' game behavior. The financial market can be a currency market, where one nation's currency can be exchanged for another currency at a certain exchange rate.



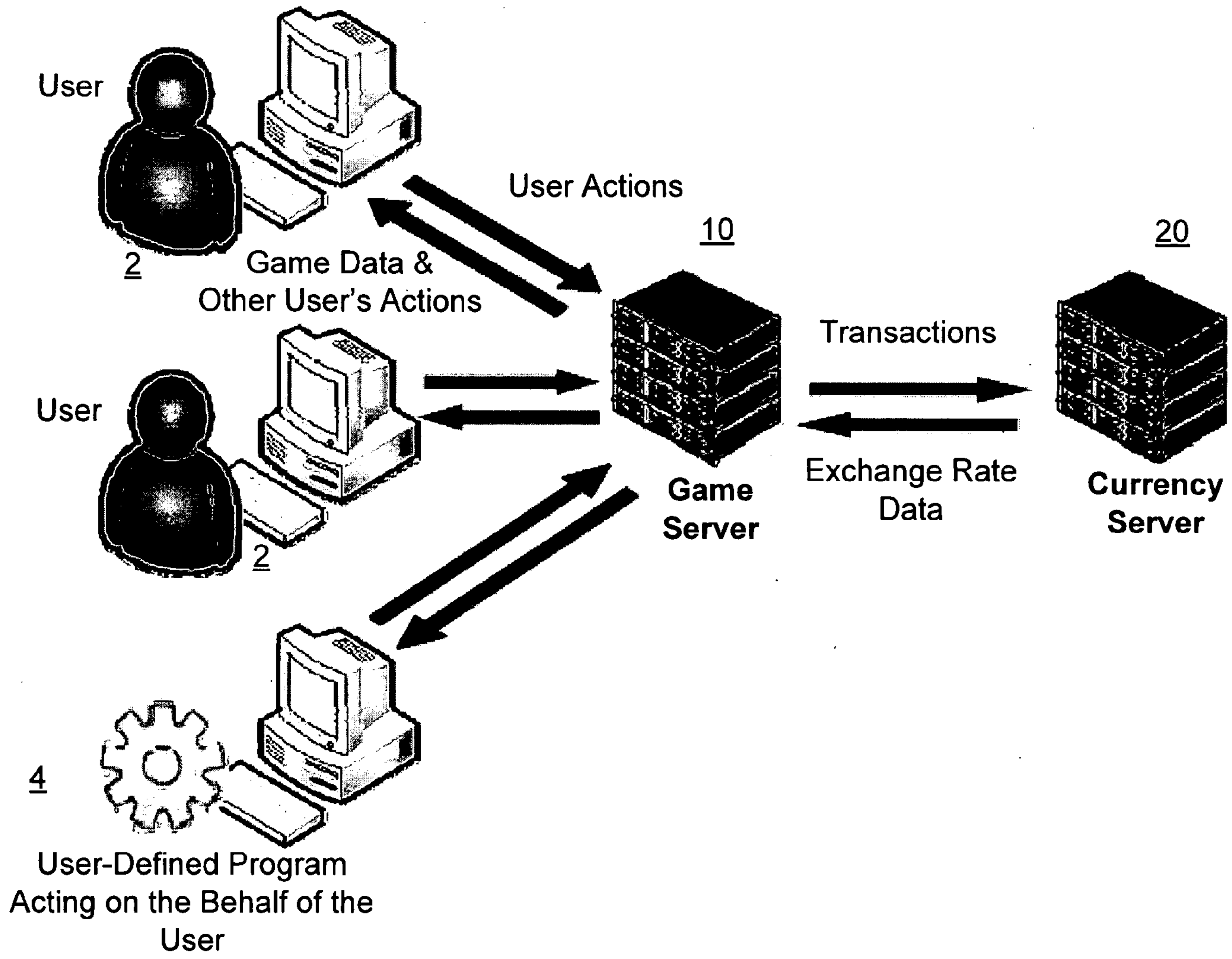
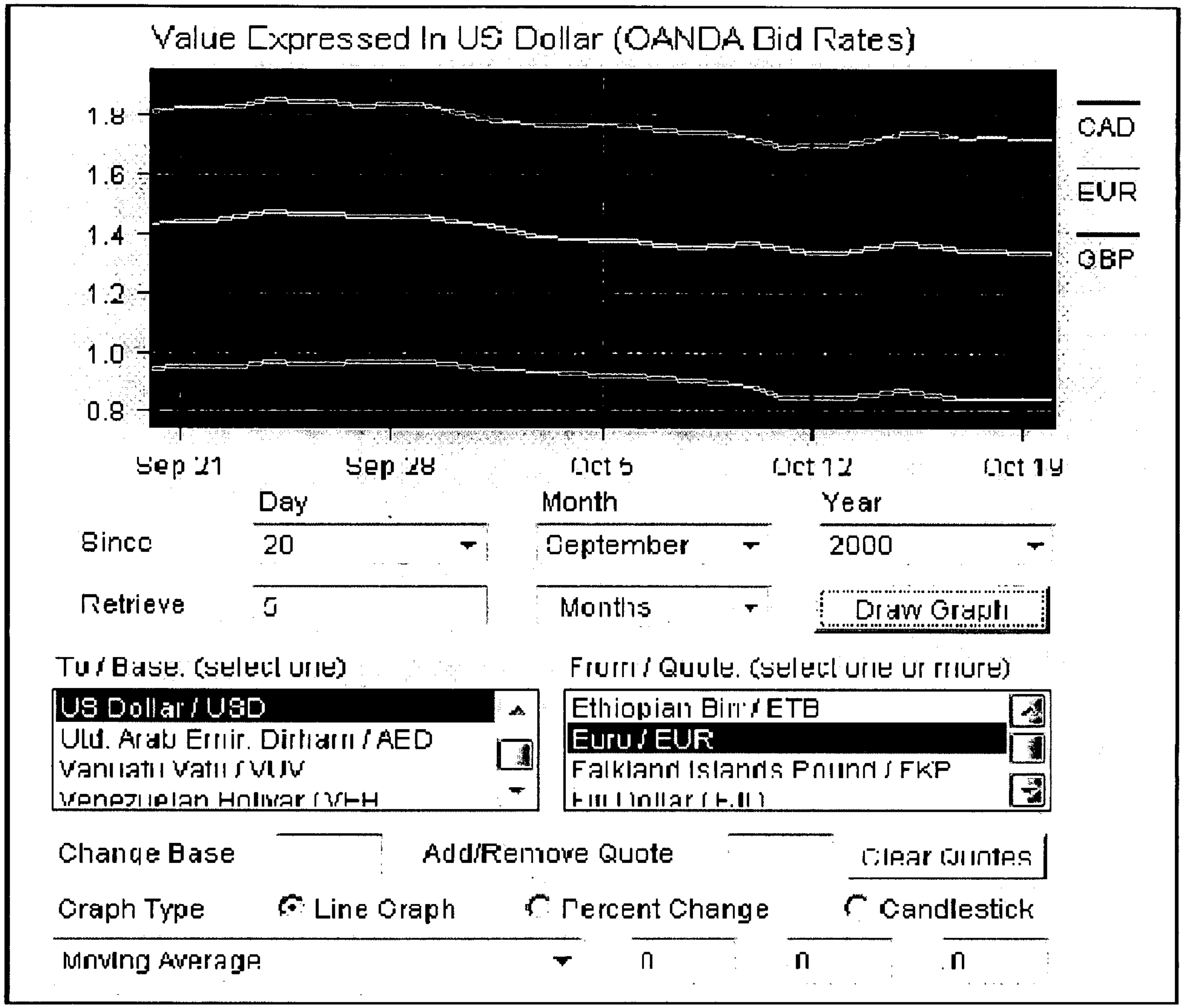

User
2
User Actions
Game Data &
Other User's Actions
10
20
Game
Server
Transactions
Exchange Rate
Data
Currency
Server
User
2
4
User-Defined Program
Acting on the Behalf of the
User

FIGURE 1



Since Day 20 Month September Year 2000

Retrieve 5 Months

To / Base: (select one)
US Dollar / USD
Utd. Arab Emir. Dirham / AED
Vanuatu Vatu / VUV
Venezuelan Bolivar / VEB

From / Quote: (select one or more)
Ethiopian Birr / ETB
Euro / EUR
Falkland Islands Pound / FKP
Fiji Dollar / FJD

Change Base

Add/Remove Quote

Graph Type ● Line Graph ○ Percent Change ○ Candlestick

Moving Average 0 0 0

FIGURE 2

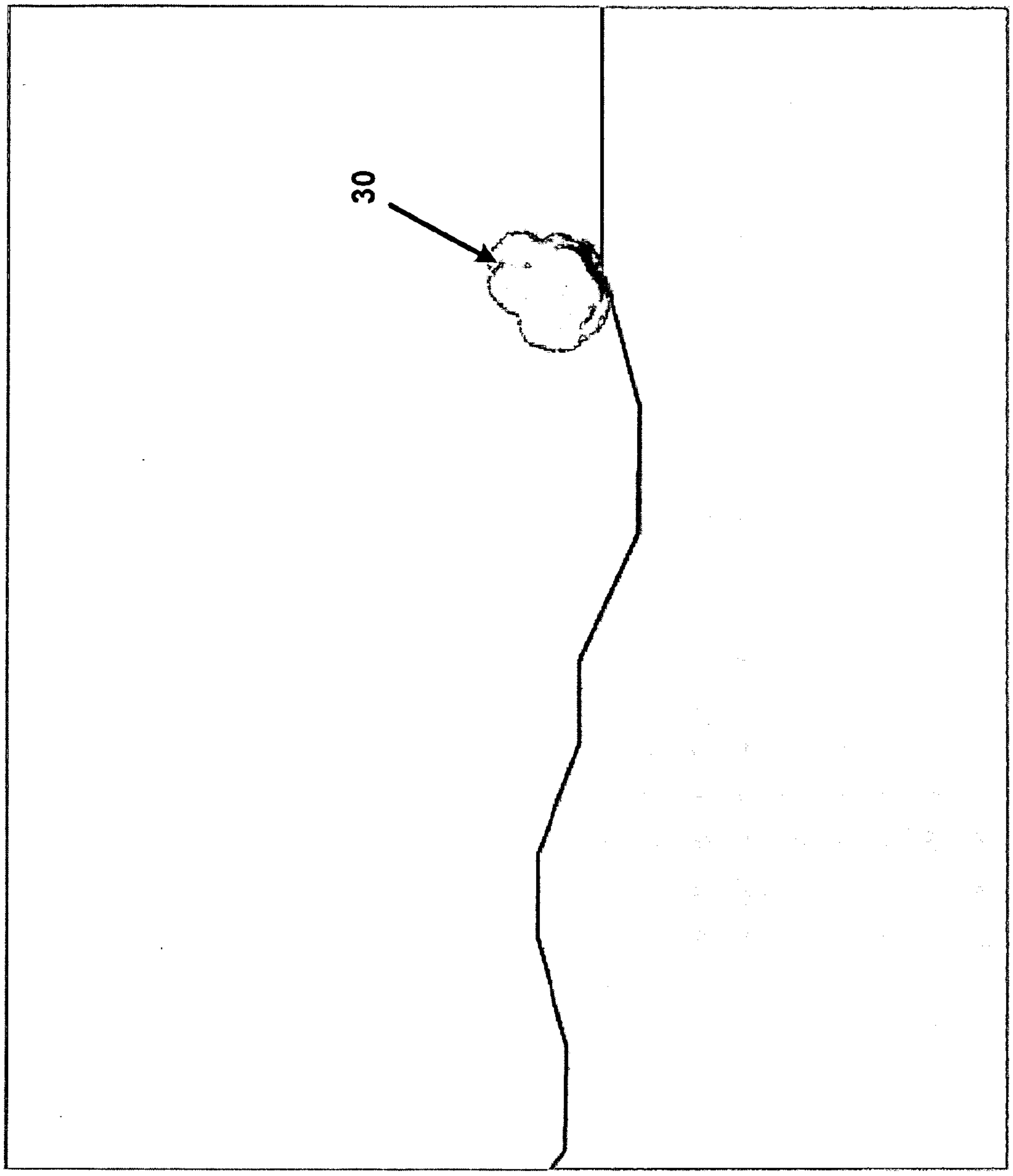
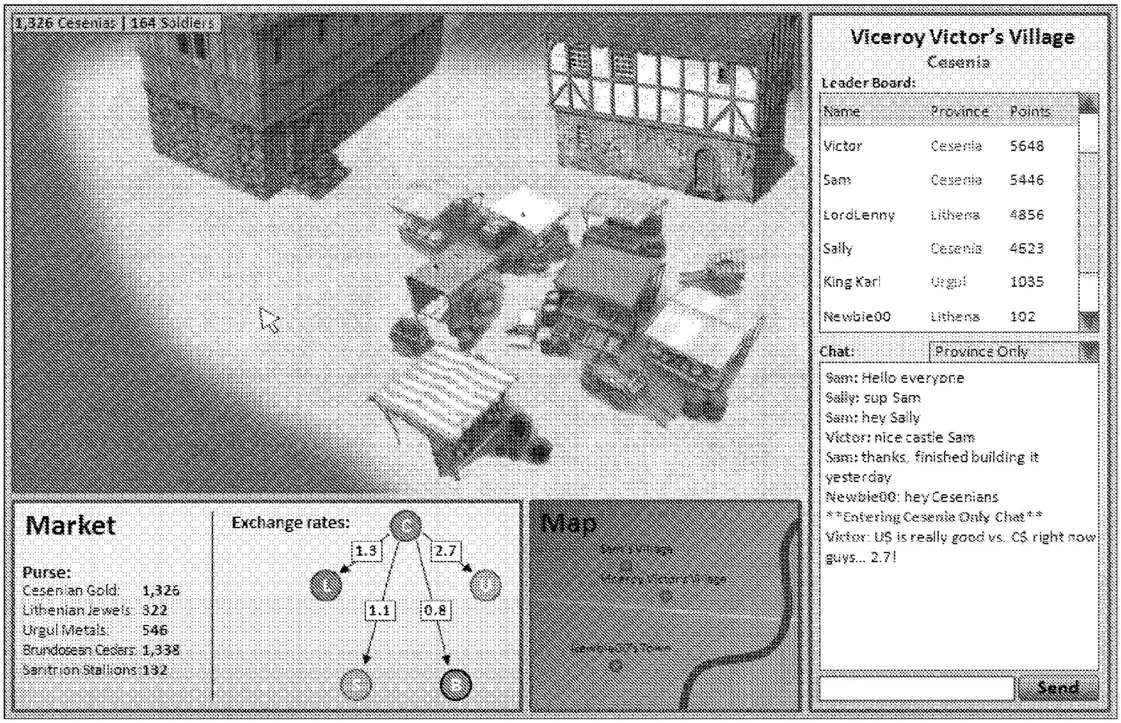


FIGURE 3



1,326 Cesenias | 164 Soldiers

Viceroy Victor's Village
Cesenia

Leader Board:

Name	Province	Points
Victor	Cesenia	5648
Sam	Cesenia	5446
LordLenny	Lithena	4856
Sally	Cesenia	4623
King Karl	Urgul	1035
Newbie00	Lithena	102

Chat: Province Only

Sam: Hello everyone
Sally: sup Sam
Sam: hey Sally
Victor: nice castle Sam
Sam: thanks, finished building it yesterday
Newbie00: hey Cesenians
Entering Cesenia Only Chat
Victor: U$ is really good vs. C$ right now guys... 2.7!

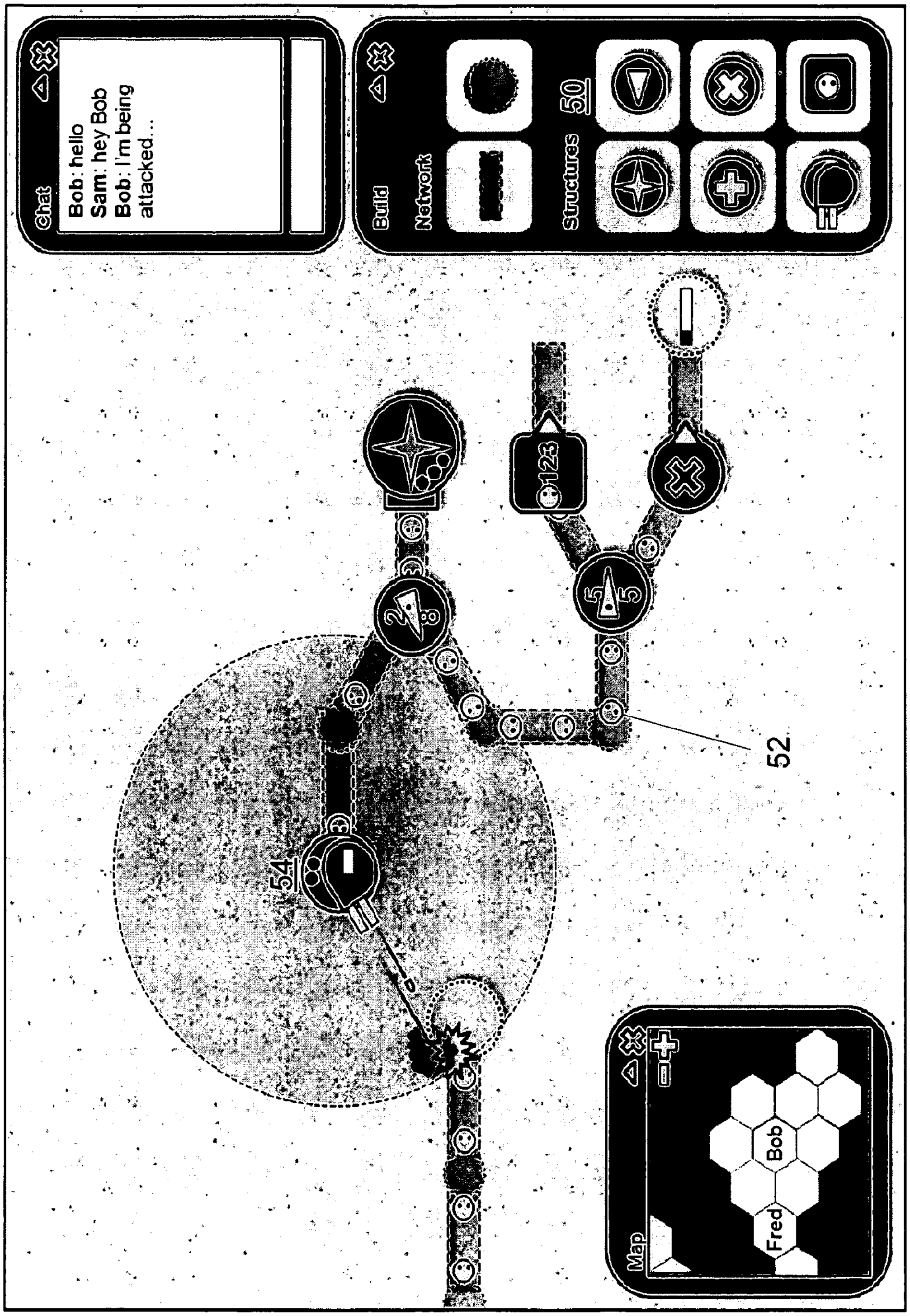
Market

Purse:

Cesenian Gold:	1,326
Lithenian Jewels:	322
Urgul Metals:	546
Brundosean Cedars:	1,338
Santrion Stallions:	132

Exchange rates:

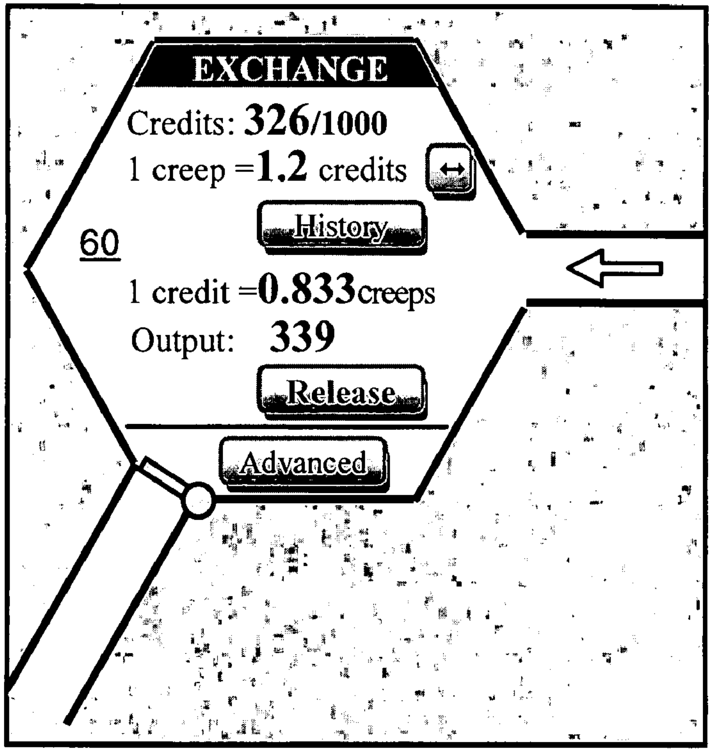
FIGURE 4



Chat

Bob: hello
Sam: hey Bob
Bob: I'm being attacked...

FIGURE 5



EXCHANGE

Credits: 326/1000

1 creep =1.2 credits

1 credit =0.833creeps

Output: 339

FIGURE 6

NETWORK COMPUTER GAME LINKED TO REAL-TIME FINANCIAL DATA

[0001] This application claims priority to U.S. Provisional Patent Application No. 61/020,911, filed Jan. 14, 2008, entitled "Network Computer Game Linked to Real-Time Financial Data," and is entitled to that filing date for priority. The complete disclosure, specification, drawings and attachments of U.S. Provisional Patent Application No. 61/020,911 are incorporated herein in their entirety by reference.

FIELD OF INVENTION

[0002] This invention relates to a network-based computer game that both uses real-time financial data to affect game state and uses participants' actions to determine and engage in financial transactions.

BACKGROUND OF THE INVENTION

[0003] Trading securities and currencies can be financially rewarding, but is a time-consuming and demanding activity. Good traders often spend long hours in front of a computer or monitor, postulating and identifying complex visual patterns, filtering large amounts of uninformative visual noise, and attempting to accurately time portfolio "movements" to maximize financial gains. Many traders also spend significant amounts of money and resources to automate their trading strategies, creating programs to modify their positions when specific criteria are met.

[0004] Internet or network-based games are experiencing increasing popularity. These games can be single-player games, or multi-player games involving two or more participants, playing cooperatively and/or competitively. Many computer gamers devote extraordinary amounts of time and energy on their computers mastering games, learning and identifying visual patterns, and honing their skills and timing to maximize scores or the equivalent for their particular game. Some gamers also automate portions of their gaming. These characteristics are similar to those of successful traders in financial markets.

[0005] What is needed is a system that masks the dry complexity of financial markets, allowing users to manage a portfolio through an interesting and easy-to-learn gaming interface, where the game interface may or may not resemble trading, and players may be completely unaware of the linkage to financial transactions and investment.

DESCRIPTION OF THE DRAWINGS

[0006] FIG. **1** shows a diagram of a system in accordance with an exemplary embodiment of the present invention.

[0007] FIG. **2** shows a screen from a commercial service for currency trading.

[0008] FIG. **3** shows a screen from a game in accordance with an exemplary embodiment of the present invention.

[0009] FIG. **4** shows a screen from another game in accordance with another exemplary embodiment of the present invention.

[0010] FIG. **5** is a an example of a multiplayer mini-game in accordance with another exemplary embodiment of the present invention.

[0011] FIG. **6** is a an example of an exchange node from a multiplayer mini-game in accordance with another exemplary embodiment of the present invention.

DETAILED DESCRIPTION OF EXEMPLARY EMBODIMENTS

[0012] FIG. **1** shows an exemplary embodiment of the present invention comprising a system wherein one or more users' or players' games, which are played on a user or player computer **2**, are connected to a centralized game server or servers **10** coordinating the game play, where the game server is, in turn, connected to a financial data and trading server (such as a currency server) **20**. In an alternative embodiment, the game server and financial server may reside on a single physical server.

[0013] The game server **10** receives financial data on market conditions (such as currency markets) from the financial server **20** and regularly synchronizes user's game environments to match the financial data. The game server **20** propagates financial data to the games **2**, coordinates communication (if any) between the games **2**, and can initiate the opening and closing of positions in the financial market based on players' game behavior.

[0014] In one exemplary embodiment, the financial server **20** is a currency server with one or more links to the currency markets, where one nation's currency can be exchanged for another currency at a certain exchange rate. Exchange rates in the currency markets change in real time, and investors use these rates and rate changes to decide in which currencies to invest or divest. Several commercial services exist for receiving currency data and executing trades automatically. One example of such a service is OANDA.com, which is shown in FIG. **2**.

[0015] The player's movement or actions in the game are regularly gauged according to the performance of a corresponding "virtual portfolio." By mastering the game, the players also are mastering the disciplines of financial trading, including but not limited to currency, commodities, or securities trading.

[0016] In one embodiment, the game software executes on a local computer or game console **2** (including, but not limited to, a game console using a television) and communicates over a network (which may be wired or wireless) with the centrally-managed game server or servers **10**. The game software can be downloaded in any variety of ways known for such games. Thus, as non-limiting examples, the game software may be downloaded onto a computer hard drive and run as a client program, downloaded in memory and run through an Internet web browser, or downloaded and run through a game console, including but not limited to a Playstation or Nintendo Wii. Players may automate their participation in the game to various extents, including using macros for game tasks or actions, or using a user-defined or created program to act on their behalf **4**.

[0017] The game server **10** coordinates players of the game and their interaction with the currency or financial server **20**. In online multiplayer games, where several human players are playing against each other at the same time across Internet connections, there are two ways to coordinate players. Each player's client software could maintain its own copy of the game state, and as players perform actions in the game, those actions are broadcast out to all the other players' clients and they update their state accordingly. Alternatively, a centralized server could maintain the game state, and as players perform actions in the game, the server updates its state and broadcasts the updated state out to each player's client software.

[0018] In one exemplary embodiment, because the currency server will be mimicking best players' actions in the real currency markets, it may be inefficient to have every player keep track of the game state and connect to the currency server. In this embodiment, the game server handles all the communication between the players and the currency server. Another advantage to a centralized game server is that it is much more difficult for players to modify the game state illegally or improperly than if the game state was stored somewhere on their machine.

[0019] The game server can connect to the currency or financial server via whatever interface the currency server exposes (for example, web services or direct port-to-port communication). The game client software can communicate with the game server via a commercially available socket server (such as SmartFoxServer), which handles common multiplayer game functionality like chat rooms, game lobbies, and matchmaking, as well as communication between the clients in-game. As players perform actions in the game, the client software dispatches messages to the game server, and the game server updates the game state and broadcasts the new game state out to all the clients.

[0020] The game server **10** regularly updates the game software based on financial market data obtained from the financial server **20**, which manages financial data and financial trading applications. If the financial transactions involve currency, for example, the financial server provides real time currency market data, and can open and close positions in the currency market. The financial data may be encrypted or derived. In one embodiment, the financial data may be universally broadcast to all game software clients, or modified or customized in some way. The game software then incorporates the financial market data into their respective corresponding game states, which affects the array of choices, constraints, or visual stimuli presented to the game player. This incorporating and mapping of the financial market data into the game state may or may not be transparent to the game player (e.g., a numeric price value may correspond to the color of a tile within the game), and the use of such financial market data may not even be known by the player.

[0021] Game players react to the mapped financial market data in the game state, and make decisions or take actions accordingly. Thus, players are making decisions or taking actions that correspond to trading events. These actions are related to the game server, which evaluates them in the context of the player's corresponding "virtual portfolio." The game's interface may support players performing these actions manually or automatically, so that players can automate their game strategy so they do not have to be physically present to take game actions (and thereby perform trades in their virtual portfolio). As the game proceeds, the composition of each player's virtual portfolio will change in response to their actions, so that the virtual portfolio of one player may differ from that of another. Players' scores may be directly tied to performance of the corresponding virtual portfolio.

[0022] The performance of all players may be monitored by the game server. High-performing players can be identified, and if a certain standard or standards are met (for example, if their virtual portfolios pass a sufficient profitability threshold), real-life financial trades and transactions can be made based on these players' subsequent actions. In one embodiment, the financial server mirrors these players' in-game actions with actual financial transactions and investments. These actual transactions can be made individually, so that trading in an actual portfolio can be based on the actions of a single player, such as the currently best-performing player. Alternatively, the trading can be made as a portfolio based on the balanced or weighted consolidation of many players' actions.

[0023] The client software interacts with players directly and is the graphical interface to the game. This software is the visualization of the game state maintained by the game server and it provides the interface for players to manipulate game state and communicate with other players. There are several technologies available for developing client software, such as Adobe Flash. Flash runs inside a web browser and is platform independent and has support for vector graphics, which makes it a good choice for "casual" online games (i.e., games that attract a wider audience, are short in duration, and do not require high-performance graphics capabilities).

[0024] Connection with financial or currency markets is not exclusive to one kind of game. However, some game play models create a better analogy to certain financial markets, such as currency trading, than others. Several examples of games that provide a good analogy to trading in currency markets include the following:

> **[0025]** A real-world simulation where players create a virtual society and economy. This is the most direct analogy to the currency markets as it is a virtualization of the real world with few layers of game play abstraction.
>
> **[0026]** A real-time strategy game (RTS) where players are citizens of various fictional countries or territories. Players expand their territory by conquering other villages and cities, financing their armies with money earned from wisely investing in other countries' fictional currencies, where the exchange rates are driven by real-world exchange rates. This kind of game could have multiple themes (such as medieval kingdoms or space exploration).
>
> **[0027]** An arcade game where real-world exchange rates drive the position of a target. Players must accurately predict the behavior of the markets to hit the target and score points.
>
> **[0028]** A game where players develop a trading "engine" that (either automatically or manually) executes trades on their behalf in the currency markets, and then players use earned resources to compete in various RTS or arcade-style mini-games.

[0029] The game software provides, in effect, an analogy to financial markets. The analogy can be fairly abstract, as seen in FIG. **3**. In this exemplary embodiment, the object of the game is to maintain momentum of a boulder **30** as it rolls over terrain. The financial data (exchange rate data, in this example) from the financial server drives the algorithm that generates game terrain. For example, rising currency value may be mapped to downward-sloping terrain, and vice versa. Players have control over the boulder's path, and can respond to terrain conditions. For exchange rate data, for example, players can invert the upcoming terrain (which would correspond to closing a long position and opening a short position, or vice versa), or can push the boulder to a different, nearby slope (which would correspond to closing their current position and opening a new one on a new currency pair). By observing trends in the terrain, players can make informed decisions in the game, with the goal of building ever more momentum for the boulder. The game would terminate when the exchange rates of the gamer's open positions become so unfavorable that the boulder has slowed to a stop.

[0030] Whatever the game's style, it should be entertaining and intriguing while maintaining a link to the financial or currency markets so that players will want to become more skilled at the game. Greater skill in the game translates to greater skill at trading in the markets. The example of a trading engine mentioned above, for example, would be one way of accomplishing the goal of creating a loyal base of players that become skilled at trading in the currency markets.

[0031] FIG. **4** shows another example of a real-time strategy game in accordance with the present invention. In this game, each player takes the role of a citizen of a virtual province or country with a virtual currency. The virtual currency correlates to a value in the associated financial market. Where the game is associated with a currency market, for example, the virtual currency would correlate to a single real-world currency, such as the Euro or the Japanese Yen.

[0032] In this strategy game, players would control and grow their sociopolitical unit (such as a village or city or city-state) by investing and spending resources, forging alliances with other players, and attacking and plundering enemies. Players also can trade resources. Although players can gather a small amount of resources from the land around the sociopolitical unit, successful players must invest resources in the virtual currency or financial markets **32** to have the means to build advanced units, buildings, and the like. Players may trade resources through their market, which may be a specialized building owned by each player, where in-game exchange rates are driven by real-world currency rates. In this example, the parallel between the real-world currency market and the game's environment is more direct, enabling a straightforward path for game administrators to mirror trades for the best-performing players.

[0033] Incentives for succeeding may be provided. High scores may be posted, fostering competition among players. Special items or rewards may be received in the game. High performing players may be financially rewarded, such as receiving payments based on their score, or receiving $100 or some other amount in iTunes credits or Amazon dollars (or similar credits) for achieving a high score, or for every day they occupy the weekly high-score within the game, or some similar measure.

[0034] In another embodiment of the invention, the client software is hosted at an Internet website. When players visit the website on which the client software is hosted, they are prompted to create an account or log in, and then taken to a main lobby where players socialize when they are not playing a mini game or editing their engine. The website could also provide social networking features like profiles, avatars, forums, and groups.

[0035] From the lobby page or screen, players can enter a page or screen where they edit their resource engine, which is the in-game connection to the currency market. All players are given a fixed rate of resources all of the time, regardless of whether or not they are logged on to the game website. These resources allow players to participate in mini-games throughout the website, so it is to the player's advantage to maximize their resources. The simplest engine would simply stockpile resources as they were earned. Other players would monitor exchange rates as they were fed in from the currency server and manually manipulate their engine to take advantage of trends in the markets.

[0036] The most advanced players may set up automated engines that make decisions about where to allocate resources using a graphical programming language. The graphical programming language comprises of nodes wired together with pipes through which resources travel. There are several types of nodes available to players, including:

[0037] Emitter: this node provides players with resources at a fixed rate, starting as soon as players create an account and continuing until they cancel their account. There is only one of these nodes per player, and they cannot be edited by the player. The output of this node (represented by a small creature, called a "creep," since it is used to build structures and attack enemies in addition to being a form of virtual currency) is what gives players the ability to do other things around the website, like play games.

[0038] Storage: like a capacitor in an electrical circuit, the storage node simply queues up creeps as they enter. The simplest engine consists of an emitter feeding into a storage node. The player withdraws creeps from storage to use them around the website. If a storage node is used elsewhere in the engine, the withdrawal of creeps (so that they continue downstream) can be automated if a condition specified by the player is met.

[0039] Switch: this node re-routes creeps along different paths out of the node. The state of the switch can be set manually or it can be conditioned on current exchange rates and the state of creeps throughout the engine.

[0040] Exchange: the exchange node **60** (as seen in FIG. **6**) is the link to the currency markets. In one exemplary embodiment, creeps correspond to US Dollars (USD), and each exchange node corresponds to a different foreign currency (relative to USD, like Japanese Yen or Australian Dollars). Exchange nodes could explicitly show which currency they represent, or they could abstract currencies with colors or icons, depending on if the connection with the real world markets is revealed to the players. As players direct creeps to flow into exchange nodes, they are investing USD (or "opening a position") in that foreign currency. When the player closes their position, their investment is converted back to creeps at the current exchange rate and those creeps flow down the network. Players can manually close positions, or define rules that close and open the exchange nodes automatically. It is also possible to open a "short" position, meaning the player forecasts that a foreign currency will continue to do poorly versus the USD. Instead of converting creeps into foreign currency as they flow in, creeps stay as creeps, but players "borrow" the equivalent amount in the foreign currency according to the rate when each creep flows in. Then, when a player closes their position, stored creeps are converted to the foreign currency, any "debt" is paid off and the remaining profit (or loss) flows down the network (after it is converted back to creeps).

[0041] In one embodiment, the programming language is extensible, so that as players become more advanced in their logic, they have the programming tools they need to create an engine that implements their trading strategy. The best players will compound their fixed income with a sophisticated engine which allows them greater access to other features of the website.

[0042] A player's motivation for building an efficient engine is so that he or she will have the resources to participate in other activities around the site. A game where players build a network of structures **50** with creeps **52** would be familiar to them after using the engine-building interface, as seen in FIG. **5**.

[0043] Instead of building a network to multiply resources, players compete against each other to control territory and win military battles against one another. In addition to building storage nodes and switches as in the engine, players construct various kinds of towers that can destroy opponents' networks and structures. Players pull creeps from their engine and deploy them on to a terrain where they are used to build the player's network and provide ammunition for their weapons **54**. The game could end once a time limit was reached, once a certain condition is met (for example, a "king of the hill" objective where a player must build and defend a special tower in a central location), or once all players but one have been eliminated.

[0044] Players will be rewarded for winning games with rankings, scores, and possibly desirable virtual objects, like custom clothes for the player's avatar. This will motivate them to optimize their resource engine so that they will be able to play more games and earn more rewards. The decisions that the best players' engines make will be sent to the currency server, and those decisions will be replicated with actual capital for a profit.

[0045] An example of one embodiment of how in-game actions correspond to currency server actions is as follows:

[0046] Player iamthebest is currently the best player in the leader boards that track which players maximize the resource returns they earn on the fixed income given to all players. Because he is a successful investor in-game, his engine design is leveraged on a real-world currency server. The following chart walks through an example of iamthebest adjusting his resource engine (note that this example ignores trading fees):

[0047] Alternatively, if iamthebest had accurately predicted the behavior of the exchange rate, he could have "flipped" the exchange node and opened a "short" position, as follows:

[0048] Creeps begin flowing into the node, at a rate of 1 creep=1 orange credit. After 4 creeps flow in, he has "purchased" 4 creeps and owes 4 orange credits.

[0049] The exchange rate drops to 1 creep=2 credits, and 2 more creeps flow in. He has now purchased 6 creeps and owes 8 credits.

[0050] He decides to close his position. He sells his 6 creeps for 12 credits, his "debt" of 8 credits is paid, and the profit of 4 credits (converted back to 2 creeps) flows down the network, in addition to the original 6 creeps that were held as collateral against his 8 credit "loan."

[0051] Neither of these exemplary scenarios, for simplicity, factor in the "spread" involved with both long and short positions. The spread is the (typically) small difference in exchange rate depending on whether one is purchasing or selling the base currency in a trade. There are also brokerage fees to be considered, as well as market impact (i.e., where the size of a purchase or sale affects the price available). Market impact, however, is expected to be negligible in the highly-liquid currencies exchange markets. The effect of fees and other "friction" in trades makes an "even trade" from one currency to another and then back to the base currency result in a small loss to the player. Friction forces players to make more profitable trades, so that they still earn resources.

In-Game Actions	Game Server/Currency Server Communication
Player iamthebest logs on to his account.	The game server opens a new socket connection to the currency server, and logs into iamthebest's corresponding account on the currency server.
As he is building his engine, he monitors the various types of exchange nodes. He notices the orange exchange node 60 has a favorable exchange rate (FIG. 6).	The game server receives periodic exchange rate updates from the currency server and updates the exchange nodes in the engine design interface.
	The game server sends periodic "heartbeat" requests to keep its connection to the currency server alive.
He changes a setting on a switch to direct his resource flow into the orange exchange node.	The game server sends a request to open a position in Japanese Yen, and the currency server sends a confirmation response.
Creeps begin flowing in to the node, at a rate of 1 creep = 1 orange credit. After 4 creeps flow in, the player has "sold" 4 creeps and "purchased" 4 orange credits.	The game server sends four requests to invest $1,000 as Japanese Yen (¥). For each request, the currency server confirms that iamthebest has sold $1,000 and purchased ¥1,000.
The exchange rate drops, so iamthebest redirects the resource flow along another route, but not before 2 creeps flow in at a rate of 1 creep = 2 credits, for a total of 6 creeps and 8 credits stored in the node.	The game server sends two requests to invest $1,000 in Japanese Yen. For each request, the currency server confirms that iamthebest has sold $1,000 and purchased ¥2,000.
He decides to cut his losses and flush the node. He sells his 8 credits for 4 creeps, for a loss of 2 creeps from the original 6 that flowed into the node.	The game server sends a request to close the open position. The currency server confirms that iamthebest sold ¥8,000 for $4,000, for a net loss of $2,000 on his account balance.
Disheartened, iamthebest logs off.	Because of this misstep, iamthebest loses his top place on the leader boards. The game server logs his account out of the currency server and shuts down his connection since his virtual investments are now too risky to mirror in the real-world markets.

[0052] Thus, it should be understood that the embodiments and examples have been chosen and described in order to best illustrate the principles of the invention and its practical applications to thereby enable one of ordinary skill in the art to best utilize the invention in various embodiments and with various modifications as are suited for particular uses contemplated. Even though specific embodiments of this invention have been described, they are not to be taken as exhaustive. There are several variations that will be apparent to those skilled in the art. Accordingly, it is intended that the scope of the invention be defined by the claims appended hereto.

We claim:

1. A system for trading in a financial market, comprising:

a financial server for effecting trades or transactions in a financial market; and

a game server in electronic communication with the financial server;

wherein conditions in a game on the game server change in response to changes in conditions in the financial market.

2. The system of claim **1**, further wherein the financial server effects trades or transactions in the financial market in response to actions taken in the game by one or more players.

3. The system of claim **1**, wherein the financial market is a currency market.

4. The system of claim **1**, wherein the game is a multiplayer game.

5. The system of claim **1**, wherein the game is accessible to players through the Internet.

6. The system of claim **1**, wherein players develop a trading engine in the game.

7. The system of claim **1**, wherein players develop resources in the game or for use in the game through exchange nodes.

8. The system of claim **1**, wherein players are aware of the connection between the game and the financial market.

9. The system of claim **1**, wherein players are unaware of the connection between the game and the financial market.

10. A computer-implemented method for effecting trades in a financial market, comprising the steps of:

receiving financial data about conditions in a financial market;

changing conditions in a game on a computer game server in response to changes in conditions in the financial market;

receiving game data about actions taken by one or more players in the game in response to changes in conditions in the game; and

making one or more trades in the financial market based on the received game data.

11. The method of claim **10**, wherein the financial market is a currency market.

12. The method of claim **10**, wherein the game is a multiplayer game.

13. The method of claim **10**, wherein the game is accessible to players through the Internet.

* * * * *