



US 20120064974A1

(19) **United States**

(12) **Patent Application Publication**
Washington

(10) **Pub. No.: US 2012/0064974 A1**

(43) **Pub. Date: Mar. 15, 2012**

(54) **WEB APPLICATION THAT ALLOWS ONE
MILLION PLAYERS TO ENGAGE
SIMULTANEOUSLY IN PRIZE-ORIENTED
TOURNAMENTS**

Publication Classification

(51) **Int. Cl.**
A63F 9/24 (2006.01)
(52) **U.S. Cl.** **463/42**

(75) Inventor: **Karen Lynnette Washington**, Los Angeles, CA (US)

(73) Assignee: **Karen L. Wahington**, Los Angeles, CA (US)

(21) Appl. No.: **13/229,698**

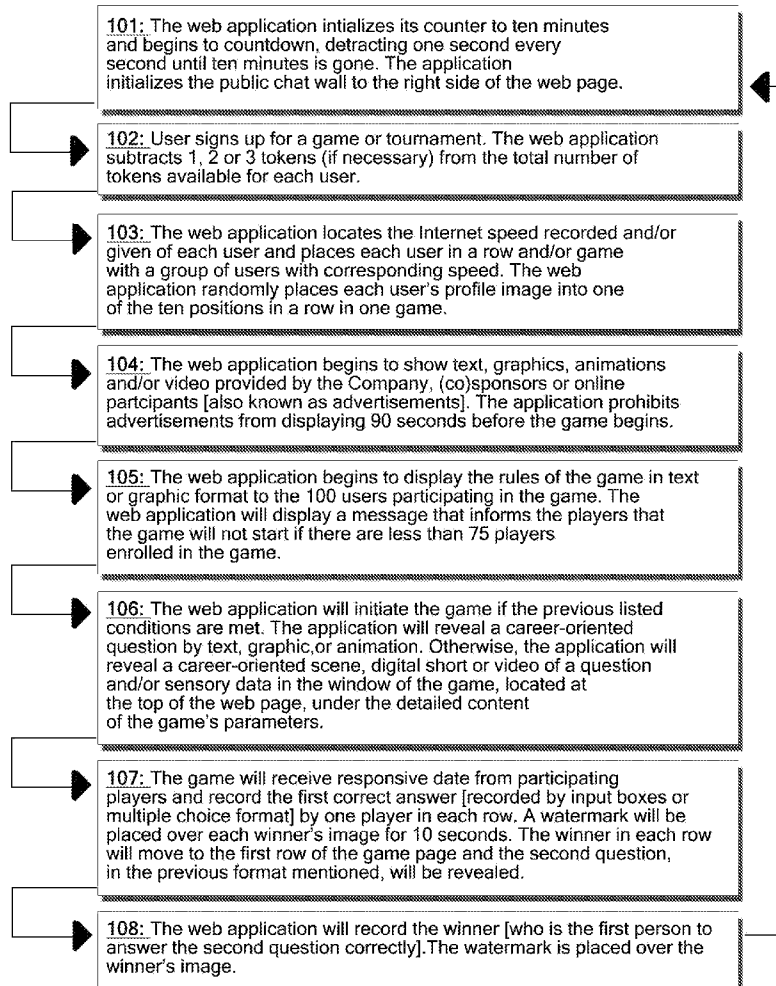
(22) Filed: **Sep. 10, 2011**

Related U.S. Application Data

(60) Provisional application No. 61/403,007, filed on Sep. 10, 2010.

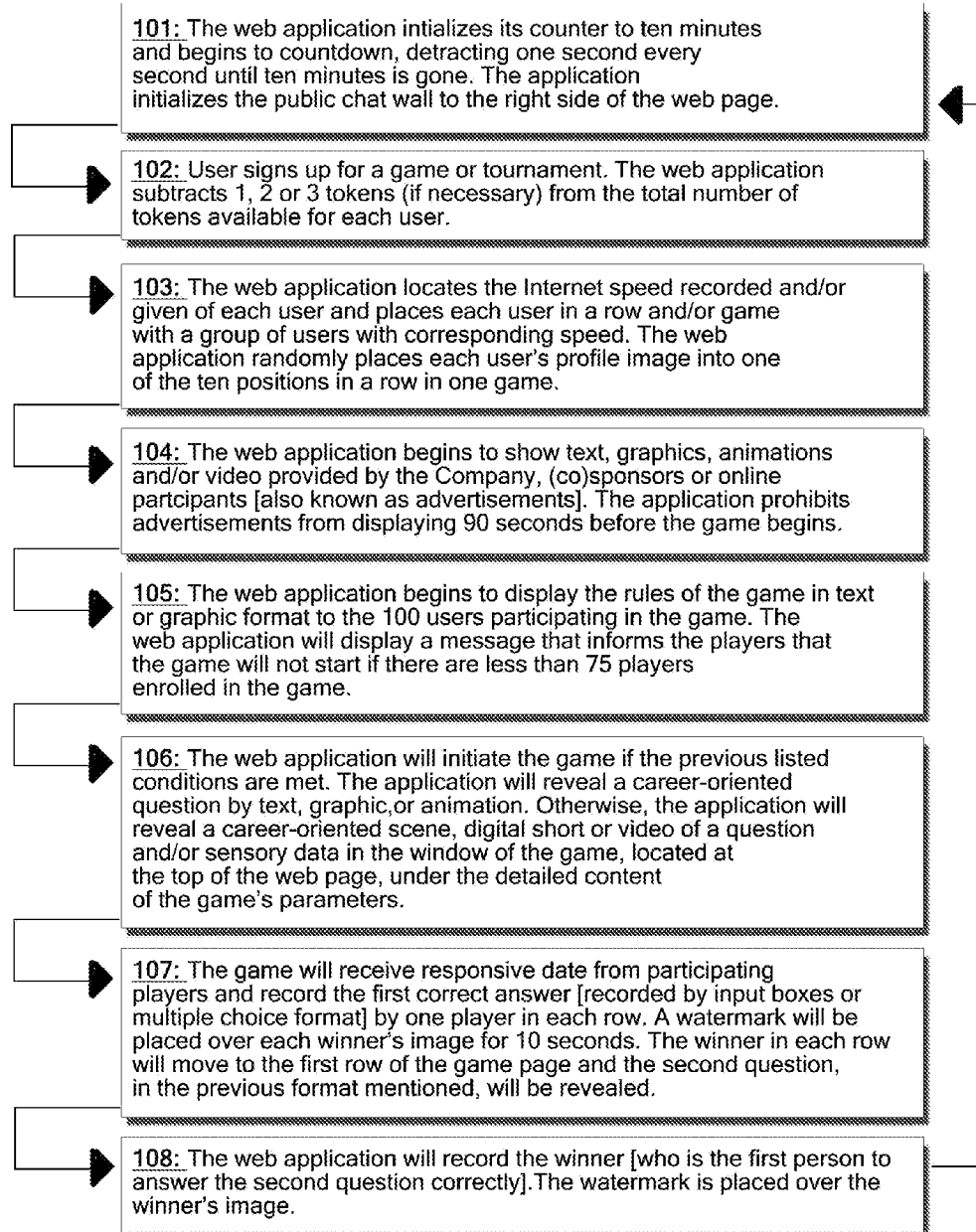
(57) **ABSTRACT**

There is provided a web application and/or software enabling one million users to engage in real-time tournaments that center around trivia and/or entertainment-oriented animations, graphics, text and videos. The web application accepts up to one million players in increments of 100 participants per web page and selectively places Internet users in 10 rows, categorized by Internet connection speeds. The tournaments consist of 1, 2 or 3 rounds with two segments in each round. The web application will record the first correct answer from one participant in a row of ten participants. Each winner [profile image] will be highlighted in his/her row and move on until one winner is identified. The web application will provide a chat area, video player, web cam features and private request chat interactions before each game start time.



*For tournaments, this process will repeat once or twice. Subsequent counter may decrease to 5 minutes.

Fig. 1



**For tournaments, this process will repeat once or twice. Subsequent counter may decrease to 5 minutes.*

Fig. 2a

Start of 3-round tournament

1,000,000 positions available / 100 positions per game = 10,000 games



Start of 2-round tournament OR start of 2nd round of 3-round tournament

10,000 positions available / 100 positions per game = 100 games



Start of 2nd round of 2-round tournament OR start of 3rd round of 3-round tournament OR
start of game

100 positions available / 100 positions per game = 1 game



1 WINNER

Fig. 2b

Format of each game

100 positions available / 10 rows = 10 positions per row

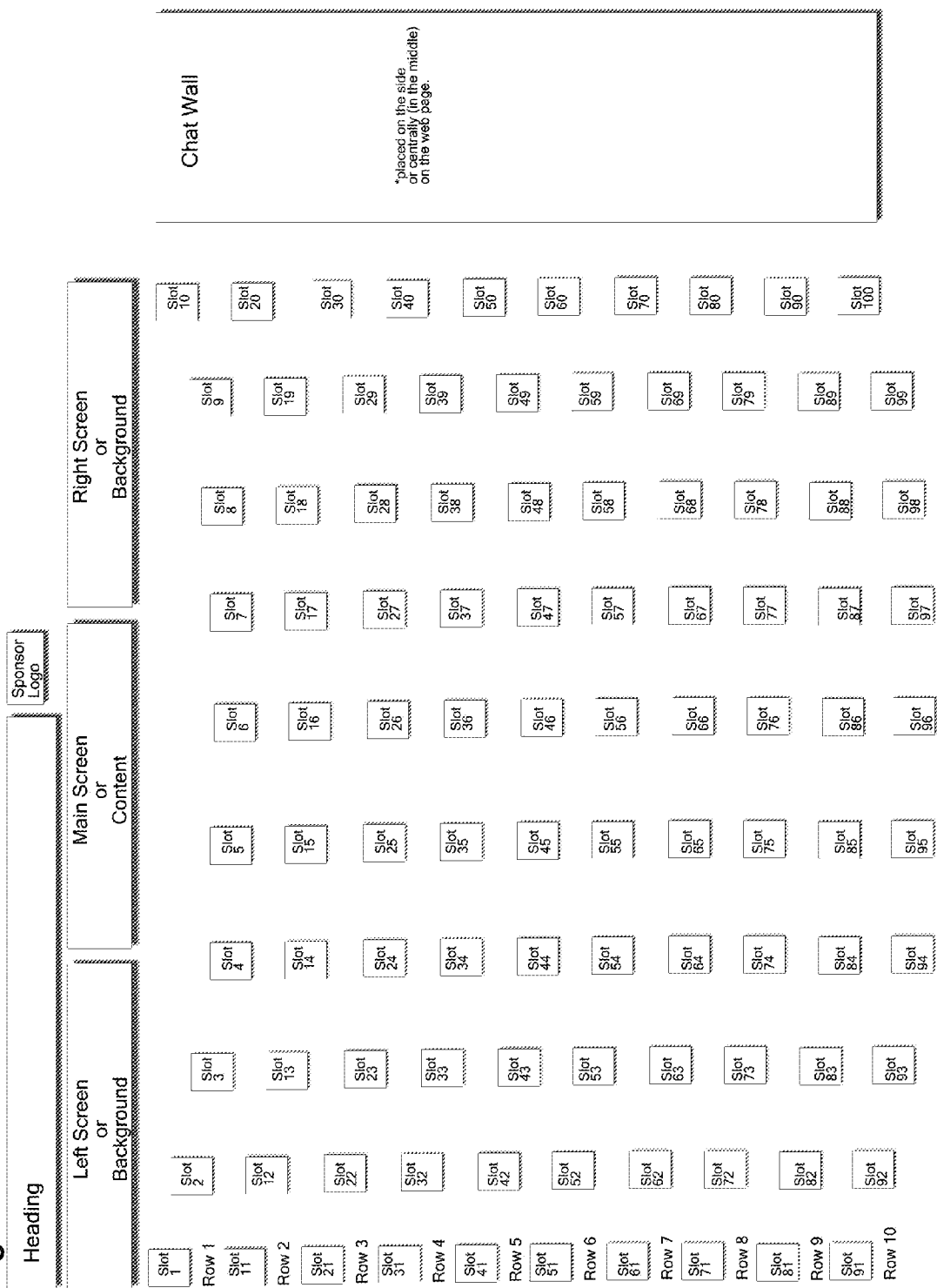


Formula of game

10 winners (1 winner per row) / 10 rows = 1 winner out of 10 rows = 1 winner

**Each tournament is synchronized with games within its own tournament. 10% chance of winning per question/game/tournament.*

Fig. 3



**WEB APPLICATION THAT ALLOWS ONE
MILLION PLAYERS TO ENGAGE
SIMULTANEOUSLY IN PRIZE-ORIENTED
TOURNAMENTS**

BACKGROUND OF THE INVENTION

[0001] 1. Field of the Invention

[0002] The present invention relates generally to online game systems. More particularly, the present invention relates to interactive transparent game systems.

[0003] 2. Background Art

[0004] Online games and tournaments have increased in participation levels in the last several years. With the popularity of mobile phones and the affordability of high-speed Internet service, online games and activities will continue to grow in usage. Most online content is offered complimentary with no substantial benefit to the user except for entertainment purposes or other related purposes. Online tournaments such as fantasy sporting leagues have gained notoriety due to the nature of the content provided and the opportunities presented to each individual player. These tournaments provide meaningful content and allow each person to obtain measurable benefits from online and mobile activities.

[0005] Most tournaments or video games online protect the user's identity through the use of avatars and other cartoonish graphics. However, with the success of many social networking sites, it is apparent that many users are willing to expose personal information and details about themselves in certain platforms and on the Internet. With a platform that provides transparent online tournaments, an online gaming or tournament-oriented site can remove the uncertainty about the game provider or lack of trust for the game provider through open communication and other transparent methods.

[0006] By providing an online environment that fosters authenticity, a game or tournament can offer great rewards based on performance, intellect, and other related skills. A web application can monitor the success of each player and relay information about the games to the members and/or users of the application in order to help foster confidence towards the system.

[0007] A system that tracks improper use, fraudulent behavior, and other less desirable actions can contribute to the reliability of an online game or tournament. With the requirement of proper use and disclosure, an online tournament can offer a level of assurance that each game or tournament is recording accurate and fair results.

[0008] A web application that provides immediate visibility of standings to participants of each round, game or tournament is a way to promote openness. It also provides each user the opportunity to communicate with others about the game. By providing areas on the website for participants to blog about the games or tournaments and allowing other participants access to the blogs for a specific period of time also fosters a more trusting environment.

[0009] A web application that provides a platform for diverse methods of animated trivia and sensory-related competitions provides a variety of possibilities for each player to win a game or tournament.

SUMMARY OF THE INVENTION

[0010] There are methods and technologies that provide user interaction with tournament-oriented online games utilizing text, graphics, animations and videos as show in and/or

described in connection with at least one of the figures, as set forth more completely in the claims.

BRIEF DESCRIPTION OF THE DRAWINGS

[0011] The features and advantages of the present invention will become more readily apparent to those ordinarily skilled in the art after reviewing the following detailed description and accompanying drawings, wherein:

[0012] FIG. 1 presents a diagram of the flowchart of the tournament-oriented online game for 0-100 participants, according to one embodiment of the present invention;

[0013] FIGS. 2a and 2b present diagrams of the operational structure of the tournament-oriented online games, according to one embodiment of the present invention;

[0014] FIG. 3 presents a diagram of the design of the game for each web page, according to one embodiment of the present invention; and

DETAILED DESCRIPTION OF THE INVENTION

[0015] The present application is directed to a system and a method for providing user interaction with other participating users of an online tournament-oriented game. The following description contains specific information pertaining to the implementation of the present invention. One skilled in the art will recognize that the present invention may be implemented utilizing methods other than the methodologies discussed in the present application. Moreover, some of the specific details of the invention are not discussed in order to obscure the invention. The specificities of the invention that are not described in the present application are within the knowledge of a person of ordinary skill in the art. The drawings and the explanations are directed to merely exemplary embodiments of the invention. To maintain brevity, other embodiments of the invention, which use the principles of the present invention, are not specifically described in the present application and are not specifically illustrated by the present drawings.

[0016] FIG. 1 presents a diagram of the flowchart of the tournament-oriented online game for 0-100 participants, according to one embodiment of the present invention. Section 101 of FIG. 1 includes information that may represent the first two requirements of the tournament-oriented online game.

[0017] Section 101 of FIG. 1 describes the process in which the web application initiates a new game. The game/tournament web page(s) is activated up to ten minutes before the start time which may be listed in several areas on the website or web application. Once the game is selected by up to 1,000, 000 players, the game web page, which contains 100 players per page, will load into the web browser of each authorized participant or player. The counter will display its current state. The front-end programming utilizing JavaScript or one of its related technologies will set the counter and will be closely integrated within the global methodologies of the application. At 9:59 minutes to go before the start of the game, the web application will load the public chat wall, which enables the players to interact with other players and/or viewers of the game. Another feature not described in Section 101 of FIG. 1 is the ability for players and/or viewers of the game to activate a web cam for 1-to-1 video chats and/or public video chats if authorized. The video chats will be deactivated utilizing front-end programming 90 seconds before the start of the game.

[0018] Section **102** of FIG. **1** describes the process in which the web application monitors the participation of each game and/or tournament. Through the open-source database or other related database or cloud-computing environment, the web application checks the status of the tournament every 30 seconds until the final 1:30 minutes prior to the beginning of the game. During this process, the web application is retrieving updates regarding the player count from the output source of the database or cloud-computing environment. The remote connection will stay connected until 3 minutes after the game concludes. The web application sends the profile name of each participant requesting access to its database and records the participant's request for access to the game. The web application waits for entity bean or other related methodology to relay the message through its remote connection regarding the eligibility of the participant. If the participant has the appropriate number of tokens [may be 1, 2 or 3 tokens] required in the participant's account, the enterprise bean or other related methodology will grant permission and update the status of the participant's account. If access is denied, the request for access by the participant is available for the garbage heap or other related methodology. If access is granted, the participant's profile name and identification number is added to the queue; the web application waits for further information from the database. The web application monitors suspicious or harmful activities from potential hackers who insert fatal SQL injections or utilize other harmful methods to deactivate, shut down or destroy the game, tournament, or web application.

[0019] Section **103** of FIG. **1** describes the process in which the global game controller of the web application is idle until the database or cloud-computing environment relays the information regarding the participant of the game and/or tournament. If the participant requests access to a game [or a 1-round tournament], the database will scan the number of requests for the game through its identification number [Game ID]. If the participant is request #000-099, then the request is finalized and the profile image of the participant is retrieved from the database or cloud-computing environment. The new object is recorded in the database and in the corresponding tables of the database. The web application records the object's player identification number [Player ID] and the Player ID is recorded under the participant's file in the database. The new object (Game ID) is assigned to a bean or other related methodology and its value is initialized to null. The object is a copy of a super object's class that is labeled in the programming of each game's design. The object is equated to the Player ID and the web application retrieves the Internet speed of the participant's system from the database.

[0020] Section **103** of FIG. **1** describes the process in which the web application initializes the access of each participant to the game environment. After the Internet speed value is obtained, the web application sets the new object's state to volatile or some other variable that indicates that the state of the object may change. The web application may search systematically up to 10,000 games for slot availability in a game (or web page) in which the average Internet speed closely equates to the speed of the participant's system. If the web application locates a corresponding game, the web application places the profile image of the participant into a randomly selected slot and records the Game ID and sends the Game ID to the stateless bean [or other related methodology] of the participant. The slot is marked filled and the database or cloud-computing environment records the information and

the system places a cookie onto the player's desktop to record the player's activity on the website. It caches certain video data and stores it on the player's desktop.

[0021] Section **104** of FIG. **1** describes the process in which the web application displays game activities at certain intervals after the receipt of the first request from a game participant. During the timeframe before the game starts, the web application will display the information of each game's topic, the date of the game, the start time of the game, the prize, the (co)sponsor's logo and provide a rollover graphic for the (co)sponsor's information [also known as the heading]. The web application may increase the area located directly under the heading to an area of no larger than 600 pixels in length and no larger than 75% of the total width of the window size of each participant. The calculations will be processed by the front-end technology of JavaScript or other related technologies. All text, graphics, animations or videos [also known as advertisements or pre-role] will not exceed a total of 8 minutes and 30 seconds. The web application will extract the advertisements from the database or cloud-computing environment. The web application will record the Game ID and record the advertisement's identification [Video ID] and send the information to the database file of each advertiser and/or sponsor. The database will cache the information. The database will send all related game data to the servlet [or other related methodology] and the JavaScript object [or other related methodology] will cache all related data to each participant's hard drive. The web application may disable the web cam chat if the system is overloaded by a certain number of bytes. However, if the system is operating in its fullest capacity or mode, the participant will have access to enable web cam chatting and have access to type messages onto the public chat wall.

[0022] Section **105** of FIG. **1** describes the process in which the web application displays specific information regarding the rules, operational methods and requirements and other related information. During the pre-game period, which refers to the 10:00 to 1:30 mark on the counter, the entity bean (or game object) is checking the status updates of the secondary beans [or other related methodology] to establish the eligibility of each game via set time outs or other related technologies. If the total number of games has not been filled to a specific requirement (for example, 75%), then none of the games will begin and the game object will extend the life of the game beyond 10 minutes. The low controller or servlet [or other related methodology] will send a message to the mid controller that initialized its game page; the mid controller [or other related methodology] is responsible for the synchronization of the 100 objects or servlets or low controllers and sends alerts to the mid-mid controller. If the mid-mid controller does not receive the specified number of alerts out of the total number of authorized mid controllers, then it is capable of suspending the start time until the requirement is met. The entity bean will suspend the start time of the game if the specified requirement (of player registration) is close to being met, with a difference of 1-10% of the total requirement level needed. For example, if the game requires 75% participation and at the 90-second mark, has 65% participation, the game may be extended for two minutes, in order to allow the system to search for additional players. If the game(s) is suspended for two minutes, the mid-mid controller sends out an alert to the mid controllers, then each mid controller sends a message to all servlets [one servlet equals one game page]. The super high controller then sends an alert to reactivate the game link

in the designated areas of the website and/or searches for online players by extending an invitation to additional players who are online and have not registered for the tournament. Each servlet [or other related methodology] checks the status of the game every **10** seconds to receive the new start time from the mid controller. The JavaScript object [or other related methodology] reports the player's activity to the servlet. If a player exits the game, the servlet deactivates the player's slot and sends the data up the chain until the mid-mid controller receives it. After the mid-mid controller acquires the appropriate number of participants, the entity bean sends a message to the mid controllers that the game is authorized to begin and the mid controllers updates each low controller, and the game will begin 60 seconds thereafter, equaling to a total of a three-minute delay or suspension. The specified requirement will be determined by the historical data of the system's activities. If the mid-mid controller needed to cancel the game due to the lack of participation, then an alert is sent to the super high controller and the super high controller will destroy the mid-mid controller after the mid-mid controller sends a message indicating that the low controllers need to display the cancellation message to each player.

[0023] Section **105** of FIG. **1** describes the process in which the entity bean [or other related methodology] may dissolve the game/tournament if it does not have enough participants after a set amount of time and/or updates. When the entity bean is initialized, it sets the number of authorized game objects. As requests are sent to the entity bean, the entity bean places the authorized game object into an array [or other related methodology] and sends its Game ID to the mid controller, and it sends it to the servlet (or low controller). This methodology may change according to the specification requirements of the system during development. The Game ID is recorded in the database or cloud-computing environment. The database is required to record all game object data and maintain the records indefinitely.

[0024] Section **106** of FIG. **1** describes the process in which the web application may begin each game according to the requirements and/or protocol of each entity bean [or other related methodology]. The appropriate personnel will enable, monitor, suspend or disable each entity bean according to the methods and procedures in which each employee has been authorized to complete. During the initialization stage of each entity bean, the number of participants authorized to compete will be stated. Each method will substantiate the requirements for each tournament. Each entity bean will set the specific details for each game. The specifics are displayed in the heading portion of the web page.

[0025] Section **106** of FIG. **1** describes the process in which the web application initializes the start of each game. After the conditions listed previously are met, the entity bean [or other related methodology] will send the final start time to the secondary controllers. The JavaScript object [or other related methodology] will display and update the data in real-time on each player's web browser. The JavaScript object will utilize its counter to count down to the final start time. The JavaScript object will display the text, graphic, animation or video. Each text, graphic, animation or video [also known as Game Trivia] will be displayed through video players, third party remote connection outlets, or other forms of transmission. Each game trivia may primarily be drafted around career-oriented trivia. Each text will display a question related to one professional area. Each graphic will display an image related to the game trivia's professional topic. Each animation will display

an animation in which a person, place or thing is depicted. The person, place or thing closely represents the professional topic of the game trivia displayed in the graphic. Each video will depict a digital short or scene that closely represents the professional topic of the game trivia displayed in the video. For example, if one game/tournament is titled, "Things a Detective Would Know" or "Detective", the game trivia could consist of a text that would ask a specific detail that only a trained police officer would know; a graphic in which a trained police officer could view and locate a specific detail of the graphic that only a trained professional would locate; an animation of an incident, or other related material that would represent an experience that the professional trained person would recognize and/or relate to; or a video that would depict a familiar environment and/or situation that a police officer would operate in like a crime scene. The JavaScript object will increase or decrease the area under the heading to the appropriate size of the game trivia. The application will display a variety of games ranging from trivia to video games.

[0026] Section **107** of FIG. **1** describes the process in which the web application retrieves the data from each synchronized object of the web page. The servlet [or other related methodology] monitors each JavaScript object. The servlet is a class with certain methods. One of its methods accepts the concurrent data from each participant/object of the game. The servlet will create new nodes for an array set [or other related methodology] and label each array set. The participant will enter data into a text box or input box or choose one answer from a multiple-choice format after he/she views the game trivia. The first game trivia display and the next game trivia display do not start until one of the method's in the servlet initializes it. If the game trivia's content requires interaction from each participant, then the area of the game trivia will be monitored through image mapping or other methodologies to record the input of each participant. Once the first participants in each row enter data, the servlet records the data as a node into the array set and displays each participant's results individually. For example, if participant in slot **51** answered the question correctly but was 3.4 seconds behind the winner of the row, the participant will receive a private information box displaying his/her results. The servlet will access the watermark located on the hard drive of each system and display **10** watermarks; one in each row or highlight the area known as a virtual seat around each of the ten finalists. The servlet collects data from the JavaScript object and sends updates to the JavaScript object as needed.

[0027] Section **107** of FIG. **1** describes the process in which the servlet [or other related methodology] will send the data to the entity bean [or other related methodology]. The entity bean will record the Player ID of each winning participant and place the data into the database and/or cloud-computing environment. The servlet will activate the JavaScript object and the JavaScript object will place each winning participant's profile image into the first row and remove the watermark or highlighted portion. The JavaScript object will activate this method **10-50** seconds after the watermarks or highlights are placed. The servlet will utilize its counter and count down **60** seconds after the first game trivia is displayed. After this period of time, the servlet will display the second game trivia.

[0028] Section **108** of FIG. **1** describes the process in which the web application will record the data from the second game trivia. The servlet [or other related methodology] will repeat the same process as stated in the previous section of **[0026]** and **[0027]**. The servlet places one watermark over the win-

ner's image or highlights the region around the profile image. Each winner is determined by the following criteria: the ability for a participant to answer the game trivia correctly and be the first participant in his/her row to answer it correctly. The servlet records the Player ID. If the game is not part of a 2 or 3 round tournament, the winner's information is sent to the database and the servlet creates a new object on a new web page under a section on the website called, "The Winner's Circle" or another similar section. The winner will have an opportunity to blog about his/her experience in that section, under his/her name. If the game is part of a 2 or 3 round tournament, then the winner is given a chance to make a "public" chat onto the web page for the other participants and/or viewers to read and then the servlet disables the slots of the web page. The servlet directs the winner to the next page and the process starts over again until there is one winner of the tournament. The counter may decrease itself to one-two minutes instead of ten minutes for the additional rounds of play.

[0029] FIGS. 2a and 2b present diagrams of the operational structure of the tournament-oriented online games, according to one embodiment of the present invention. FIG. 2a describes the mathematical formula that each entity bean will compute throughout the life cycle of each game or tournament.

[0030] FIG. 2a describes the process in which the web application formulates a 3-round, 2-round or 1-round tournament. In the first segment of FIG. 2a, it describes the first method utilized by the entity bean [or other related methodology] also known as the mid-mid controller. The actual game or tournament is the main object class in the web application. The class initializes each game object or the entity bean [or other related methodology] and the object has access to and/or controls all related non thread-safe and thread-safe methods. All game objects or entity beans are initialized by the application bean [or other related methodology], also known as the super high controller, which is the global controller of the entire application. The global controller gathers application-oriented data through set time outs or other methodologies from secondary beans. Once the game object is initialized, all subsequent beans (mid and low controllers) must register with the stub of the entity bean. The mid-mid controller activates or authorizes 1, 100 or 10,000 mid controllers, depending on the tournament size. Activation or authorization refers to the space that is set aside for the total number of slots to be utilized during the game object's life span. Each mid controller activates 100 low controllers or servlets. The entity bean will atomically initialize all games slots as the requests from the mid controllers are received. The entity bean will initialize one mid controller per 100 requests. In a situation whereas the mid-mid controller only has 80 mid controllers, due to the fact that only 80% of the slots are filled in a 2-round tournament, the mid-mid controller will authorize or deny the continuance of the tournament. After the final conclusion of each game object, the servlet will collect the data from the JavaScript object. The servlet will send the Player ID of the winner of the game object, as well as the Game ID, to the entity bean. Once the entity bean has the winner of each Game ID subset, the entity bean will create 100 new game objects and place each Player ID into a newly created game object with a new Game ID subset.

[0031] In the second segment of FIG. 2a, it describes the mathematical formula utilized by the entity bean for the second round of a 3-round tournament or the first round of a

2-round tournament. The entity bean [or other related methodology] will retrieve [from the database] and record the new Game ID subset of each of the 100 or 10,000 mid controllers. The entity bean [or other related methodology] will connect to the newly created mid controller and the mid controller will create up to 100 servlets or low controllers. The servlet will initialize the class that creates a new game page when it receives a request from the system and send an alert to the mid controller with its status. The servlet will extract the system data of each Player ID from the database and categorize the 100 participants according to their Internet connectivity speed. For example, Player 001 can have a data rate of 0.8 mbps and Player 002 can have a data rate of 0.5 mbps. Based on that information, the servlet will place each player into a separate row (of up to 10 rows) and place new players into a corresponding row. If Player 003 has a data rate of 0.7 mbps, then Player 003 is placed into an empty row. If Player 004 has a data rate of 0.8 mbps, then that player goes into same the row of Player 001. The servlet will fill each page until its completely full and create each page as the participants are recorded as active players. Active players will be monitored for the game page activity via mouse movement, session timeouts, etc. The process described in sections[0022], [0023], and [0031] will be repeated, with the exception of each game being filled to capacity. The entity bean will destroy any game objects (servlets) created by the mid controllers that have 0 out of 100 slots filled. Once the winner in each game object is recorded, the entity bean will complete the process and start the tournament. The timer will be set for two minutes after the first segment begins and the system will automatically begin whether all finalists are recorded as active or inactive.

[0032] In the third segment of FIG. 2a, it describes the mathematical formula utilized by the entity bean for the third round of a 3-round tournament, the second round of a 2-round tournament or the first round of a 1-round tournament/game. The entity bean [or other related methodology] will create one mid controller and the mid controller will create one servlet or low controller per player. The servlet [or other related methodology] will collect the data from each new player or the finalists from the previous round. The process described in sections[002], [0023], and [0031] will initialize for this round. Once the winner is identified and the servlet sends the final outcome of the Game ID to the mid controller, and the mid controller sends it to the entity bean, it will record the Player ID of the winner and send the information to the database. The fourth segment of FIG. 2a illustrates the identity of the winner.

[0033] In FIG. 2b, the two segments describe the mathematical formula the servlet utilizes as it records the data from each game object. In the first segment of FIG. 2b, it describes the 100 positions/slots available in one game object/Game ID. It divides the number of positions by the number of rows in a game object. The number of rows in each game object equals ten. The second segment of FIG. 2b describes the number of winners the game object records during Part A of a game object. There is one winner (or finalist) per row, which equals 10 winners. During Part B of a game object, the ten winners compete against each other and there is one final winner of a game object. Therefore, each participant has a ten percent chance of winning per game trivia or Part. Each participant will not compete against more than 54, 36 or 18 other participants for the prize.

[0034] FIG. 3 presents a diagram of the design of the game for each web page, according to one embodiment of the present invention. Each segment represents an object of each game object/Game ID. The diagram outlines a graphic similar to the physical setting of a stadium, concert hall or other public gathering places.

[0035] In the top, left portion of FIG. 3, the Heading represents the data object described in section[0021]. The Sponsor Logo represents the same data object listed in section [0021].

[0036] In the middle-tier portion of FIG. 3 labeled “Left Screen or Background,” “Main Screen or Content” and “Right Screen or Background” represents the game trivia section described in section[0025]. The text version of the game trivia will appear in the “Main Screen or Content” section surrounded by a colored background on both sides. The graphic version of the game trivia will appear in the “Main Screen or Content” section surrounded by a colored background or peripheral views of the graphic on both sides. The animation version of the game trivia will appear in the “Main Screen or Content” section surrounded by a colored background. The video version of the game trivia will appear in all three sections or solely in the “Main Screen or Content” section surrounded by a colored background on both sides.

[0037] The graphic and video versions of the game trivia are one of the most vital aspects of the present invention. The game trivia is formulated after career-oriented trivia questions or experiences that a professional would face in professional situations. The graphic and/or video will be a representation of a situation or atmosphere that a certain professional would face. The “Left Screen” and “Right Screen” represent the peripheral, or lateral views of a person’s eyesight. The graphic and/or videos will be displayed from the view of each participant or the view of the professional. For example, if a game is entitled, “Things a Detective Would Know,” the background of the scene for the video would show the frontal view of the participant’s eyesight. The “Left Screen” would show the view the participant would see from his/her left peripheral vision. The “Right Screen” would show the view the participant would see from his/her right peripheral vision. The video will show the view of each participant, therefore, if the person theoretically turns his/her head to the right, then the frontal view of the complete scene would appear in the “Right Screen” section. The left peripheral view would move to the “Main Screen” section and the “Left Screen” would blackout. Vice versa for the other view.

[0038] The video version of the game trivia will display second-by-second movements of the first-person view of the scene or atmosphere. The images will display similarly to the functionality of the brain as it processes images and information. For example, if the video scene is displaying a scene that is in a fast-pace environment like a motorcycle ride, the images will move quicker than an environment of a person standing still. Therefore, the participants need to process the images from the video scene or atmosphere quickly and the game trivia questions may be formatted around images in the scene.

[0039] Once technology advances, the video version may include 3-D imagery and other high-tech visual aids or objects. The videos, animations or graphics may include image-mapping programming so it will record the data from each participant’s Player ID. The image-mapping programming may assign the Player ID to the mouse event of each

participant by gathering the Player ID from the URL listed in the newly created game object.

[0040] The lower portion of FIG. 3 represents the rows and slots of the game object. Each participant is placed in a slot. The slots are categorized 1-100. Each slot will synchronize with the Game ID and the JavaScript object will record the slot number of each Player ID.

[0041] The middle, right portion of FIG. 3 represents the public chat wall. The chat wall is an input/output object that records and displays participant and/or viewer comments. The chat wall object will monitor the comments and prohibit certain comments with inappropriate language or phrases to appear on the wall. The chat wall object will be disabled 90 seconds before the game starts and will not be re-activated except for system updates or an entry from the winner of the game/tournament.

[0042] From the above description of the invention it is manifest that various methods can be used for implementing the concepts of the present invention without departing from its scope. Moreover, while the invention has been described with specific reference to certain embodiments, a person of ordinary skills in the art would recognize that changes can be made in form and detail without departing from the spirit and the scope of the invention. As such, the described embodiments are to be considered in all respects as illustrative and not restrictive. It should also be understood that the invention is not limited to the particular embodiments described herein, but is capable of many rearrangements, modifications, and substitutions without departing from the scope of the invention.

I claim:

1. A web application will collect input data and systematically place Internet users in slots in a game according to the tested and/or listed Internet speed of each Internet user; the web application will pull text, graphics, animations or videos from its database and present the content at the onset of each game; the web application will pause at set intervals and collect data from participants; the web application will record findings and continue until the operation is complete; the web application will send data to its database and direct winners to appropriate destinations in website.

2. The web application of claim 1, wherein the software program will limit each game or web page to 100 participants.

3. The web application of claim 1, wherein the software program will not begin unless a specified percentage of the total number of slots are filled in the tournament, game or web page.

4. The web application of claim 1, wherein the software program will cache selected text, graphics, animations, or video for each individual game or tournament.

5. The web application of claim 1, wherein the software program will place game information into the appropriate areas on the game web page.

6. The web application of claim 1, wherein the software program will place the sponsor’s logo in the video player’s data area, near or around the top of the game’s format and/or design.

7. The web application of claim 1, wherein the software program increases or decreases the game information area’s height and width to the appropriate size for each text, graphic, animation or video. The program activates the chat wall located in the central or middle perimeter of the web page, no larger than 450 pixels in width and not longer than 40% of the height of the web page. The program activates the private chat

and records mouse event requests from participants [a mouse event is an activity from a participant that is recorded by the program such as a click on the image of another participant in effort to request a chat]. If the participant has marked his/her image "private," the private chat will not be initiated. The web cam feature will activate if a participant requests a web cam chat 90 seconds before the game begins. All chats will be disabled 90 seconds before the game begins.

8. The web application of claim 1, wherein the software program does not display any content in the game information area 90 seconds before the game begins.

9. The web application of claim 1, wherein the software program displays text, graphic, animation or video according to the game's specifications. The program displays three videos in the video player on each web page. The video to the far left will show a visual angle that would encompass the left peripheral vision of a person's sight and/or mental imaging. The video in the center will display a visual angle that would encompass the frontal view or vision of a person's sight and/or mental imaging. The video to the far right will show a visual angle that would encompass the right peripheral vision of a person's sight and/or mental imaging. The program may show one larger-sized video instead of three smaller sized videos.

10. The web application of claim 1, wherein the software program collects the input from the participants, then highlights each winner in each row by placing a watermark image over the profile image of each winning participant or highlights the area around each finalists or winner's profile image.

11. The web application of claim 1, wherein the software program moves each winner to the front or first row of game

format or highlights the winner's profile image and records the next piece of data. The program records the final winner and offers the winner a text area on the chat wall on the web page, allowing the winner to write a message to other participants. The program records all data and places results into its database. The program closes the text area option after 30 seconds or less and places the winner's image into the next game format. The program repeats the previous steps until the final winner is selected.

12. The web application of claim 1, wherein the software program synchronizes tournaments (2 or 3 rounds) with other games in real-time. In 2-round tournaments, a game will hold 100 positions per game and equal 10,000 positions total with 100 games displaying the same data to 10,000 different participants. In 3-round tournaments, games will hold 100 positions per game and equal 1,000,000 positions total with 10,000 games displaying the same data to 1,000,000 different participants.

13. The web application of claim 1, wherein the software program will accept tokens from each participant, granting access to a game or tournament. The program will initially process credit or debit information from each participant prior to the start of each game.

14. The interface web application of claim 1, wherein the software program offers stand-alone games and holds 100, 10,000 or 1,000,000 positions for each game.

15. The interface web application of claim 1, wherein the software program accepts keyed input for each game.

* * * * *