(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property
Organization
International Bureau

(43) International Publication Date
21 September 2017 (21.09.2017)

WIPO | PCT

(10) International Publication Number
WO 2017/160917 A2

(51) International Patent Classification:
A63F 13/46 (2014.01) *A63F 13/798* (2014.01)

(21) International Application Number:
PCT/US2017/022423

(22) International Filing Date:
15 March 2017 (15.03.2017)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
62/308,823 15 March 2016 (15.03.2016) US
62/309,390 16 March 2016 (16.03.2016) US

(71) Applicant: SKILLZ INC. [US/US]; 1061 Market Street, 6th Floor, San Francisco, CA 94103 (US).

(72) Inventor: PARADISE, Andrew; 1061 Market Street, 6th Floor, San Francisco, CA 94103 (US).

(74) Agents: AMENDT, Kevin, C. et al.; Mintz Levin Cohn Ferris Glovsky and Popeo, P.C., One Financial Center, Boston, MA 02111 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— *without international search report and to be republished upon receipt of that report (Rule 48.2(g))*

(54) Title: ACROSS-MATCH ANALYTICS IN PEER-TO-PEER GAMING TOURNAMENTS

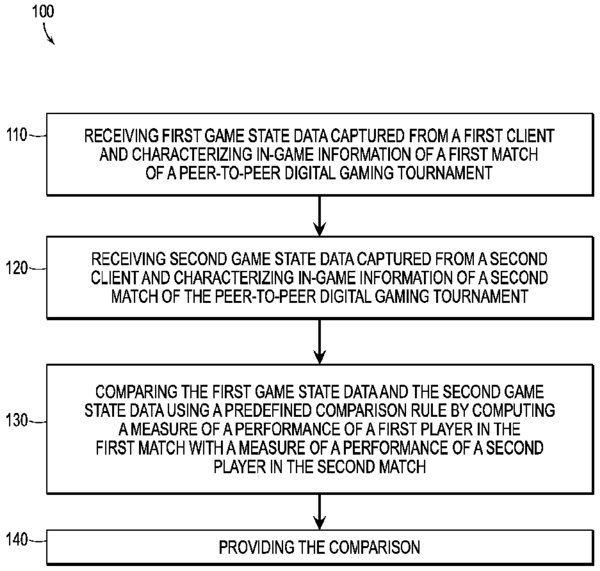

100
110
RECEIVING FIRST GAME STATE DATA CAPTURED FROM A FIRST CLIENT AND CHARACTERIZING IN-GAME INFORMATION OF A FIRST MATCH OF A PEER-TO-PEER DIGITAL GAMING TOURNAMENT
120
RECEIVING SECOND GAME STATE DATA CAPTURED FROM A SECOND CLIENT AND CHARACTERIZING IN-GAME INFORMATION OF A SECOND MATCH OF THE PEER-TO-PEER DIGITAL GAMING TOURNAMENT
130
COMPARING THE FIRST GAME STATE DATA AND THE SECOND GAME STATE DATA USING A PREDEFINED COMPARISON RULE BY COMPUTING A MEASURE OF A PERFORMANCE OF A FIRST PLAYER IN THE FIRST MATCH WITH A MEASURE OF A PERFORMANCE OF A SECOND PLAYER IN THE SECOND MATCH
140
PROVIDING THE COMPARISON

FIG. 1

(57) Abstract: First game state data captured from a first client and characterizing in-game information of a first match of a peer-to-peer digital gaming tournament is received. Second game state data captured from a second client and characterizing in-game information of a second match of the peer-to-peer digital gaming tournament is received. The first match and the second match being separate gaming instances. The first game state data captured during the first match and representative of an internal state of a digital video game. The first game state data and the second game state data is compared using a predefined comparison rule by computing a measure of a performance of a first player in the first match with a measure of a performance of a second player in the second match. The comparison is provided. Related apparatus, systems, articles, and techniques are also described and illustrated.

WO 2017/160917 A2

Across-Match Analytics in Peer-to-Peer Gaming Tournaments

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] This application claims priority under 35 U.S.C. § 119(e) to U.S. Provisional Patent Application Number 62/308,823 filed March 15, 2016, and claims priority under 35 U.S.C. § 119(e) to U.S. Provisional Patent Application Number 62/309,390 filed March 16, 2016, the entire contents of each of which is hereby incorporated by reference herein.

TECHNICAL FIELD

[0002] The subject matter described herein relates to comparing game states and player performance across disparate game matches.

BACKGROUND

[0003] Multiplayer online synchronous digital video games in which multiple players participate in the same match allow for direct comparison between player performances in the match because each player shares the same game state. For example, in a digital tennis match, each player interacts with a shared game state (and each other). Such multiplayer synchronous games also allow for automatic game analysis beyond each player's performance, such as the number of times a particular event occurs or identifying which player has the greatest change in score over a given time period because a single game state can be directly analyzed and each player's performance is referenced to the single and common game state.

[0004] In contrast, some peer-to-peer video game tournaments provide competitions in which players compete in separate game matches (possibly at the same or different times) but their performance is compared across matches to determine tournament outcome. For example, two players may each play a single player game *(e.g.,* a stage of Tetris) and their final score can be compared to determine a winner.

[0005]

SUMMARY

[0006] In an aspect, first game state data captured from a first client and characterizing in-game information of a first match of a peer-to-peer digital gaming tournament is received. Second game state data captured from a second client and characterizing in-game information of a second match of the peer-to-peer digital gaming tournament is received. The first match and the second match are separate gaming instances. The first game state data captured during the first match and representative of an internal state of a digital video game. The first game state data and the second game state data are compared using a predefined comparison rule by computing a measure of a performance of a first player in the first match with a measure of a performance of a second player in the second match. The comparison is provided. At least one of the receiving, comparing, and providing is performed by at least one data processor forming part of at least one computing system.

[0007] One or more of the following features can optionally be included in any feasible combination. The predefined comparison rule can define the measure of the performance of the first player and the measure of the performance of the second player. A first video capture can be received from the first client of a display of the first match. A second video capture can be received from the second client of a display of the second match. A portion of the first video capture or a portion of the second video capture can be selected based on the comparison. At least one of the portion of the first video capture and the portion of the second video capture can be streamed to at least one viewing client.

[0008] The first game state data and the second game state data can be compared periodically and at multiple time points. The first match and the second match can begin at different times. The first game state data can be registered in time to the second game state data. Time can be referenced to a game time, an absolute time, or a tournament time. The first game state data can characterize a value of one or more in-game data objects at two or more points of time during the first match. The first client can receive game data from a third party game server. The game data can be for implementing the first match.

[0009] The first match and the second match can be separate instances of the digital video game. The digital video game can be an asynchronous game. The digital video game can be a single player asynchronous game. The first match and the second match can be played concurrently. The predefined comparison rule can be previously defined by a user. The

comparing can include identifying at least one period of time during the first match or the second match in which an event occurs.

[00010] The receiving of the first game state data can occur during the first match. The receiving of the first game state data can occur periodically during the first match. The first game state data can include: instantaneous score, change in score over a predefined period of time, character position, and/or character health.

[00011] The first match and the second match can be initiated with a common random number seed to provide common starting conditions. The first client and the second client can include separate computing systems in communication with a third party game server and an analytics server via a network, the third party game server separate from the analytics server. At least one of the receiving, comparing, and providing can be performed by the analytics server.

[00012] Non-transitory computer program products (i.e., physically embodied computer program products) are also described that store instructions, which when executed by one or more data processors of one or more computing systems, causes at least one data processor to perform operations herein. Similarly, computer systems are also described that may include one or more data processors and memory coupled to the one or more data processors. The memory may temporarily or permanently store instructions that cause at least one processor to perform one or more of the operations described herein. In addition, methods can be implemented by one or more data processors either within a single computing system or distributed among two or more computing systems. Such computing systems can be connected and can exchange data and/or commands or other instructions or the like via one or more connections, including but not limited to a connection over a network (e.g. the Internet, a wireless wide area network, a local area network, a wide area network, a wired network, or the like), via a direct connection between one or more of the multiple computing systems, etc.

[00013] The details of one or more variations of the subject matter described herein are set forth in the accompanying drawings and the description below. Other features and advantages of the subject matter described herein will be apparent from the description and drawings, and from the claims.

DESCRIPTION OF DRAWINGS

[00014] FIG. 1 is a process flow diagram illustrating a process of comparing disparate game state data from multiple matches in an online peer-to-peer digital gaming tournament;

[00015] FIG. 2 is a system block diagram illustrating a system for implementing a peer-to-peer digital gaming tournament;

[00016] FIG. 3 is a system block diagram of a system with an example eSports Management system utilizing cross-match analytics to provide automatic recommendations for which portions of video to stream of an online eSports tournament;

[00017] FIG. 4 is an example interface illustrating example content shared by a tournament commentator;

[00018] FIG. 5 is a user interface illustrating an example video control panel;

[00019] FIG. 6 is an illustration of an example unified leaderboard;

[00020] FIG. 7 is a process flow diagram illustrating a process of defending, challenging, and viewing the leaderboard; and

[00021] FIG. 8 is an illustrating of an example leaderboard prize page.

[00022] Like reference symbols in the various drawings indicate like elements.

DETAILED DESCRIPTION

[00023] While a final match score is comparable once both matches are complete, it may be challenging to automatically make meaningful comparisons between matches during gameplay because the players have separate and independent game states that are not in a format that can be easily and directly compared. Similarly, automatic game analysis of aspects beyond each player's performance, such as the number of times a particular event occurs or identifying which player has the greatest change in score over a given time period, is also challenging because the players have separate and independent game states that are not in a format that can be easily and directly compared.

[00024] The current subject matter relates to comparing otherwise unjoinable sets of data, for example, to enable comparing game states and player performance across disparate game matches. While multiplayer synchronous games allow for comparison between player performance when the players participate in a common match *(e.g.*, the same match or game instance), the current subject matter enables comparison and analytics across multiple disparate matches, such as multiple single player matches. The current subject matter enables across-match analytics in peer-to-peer digital video game tournaments using video games that do not easily allow for across-match comparison. In some implementations, the comparison can be performed by capturing game data from each client rather than requiring access to a third party game server, for example a third party game server that creates and/or controls the game state data.

[00025] The across-match analytics can be used, in an example implementation, in an eSports stream management system for automatically suggesting or recommending portions of video streams to broadcast in real time and during a live peer-to-peer digital videogame tournament.

[00026] FIG. 1 is a process flow diagram illustrating a process 100 of comparing disparate game state data from multiple matches in an online peer-to-peer digital gaming tournament. The peer-to-peer digital gaming tournament may be an online tournament in which players compete across separate game matches (also referred to as game instances) so that not all tournament participants share the same game state. The players may participate in the tournament by executing a local application of an online video game on their clients, which can exchange game data with a third party game server in order to implement the gaming experience. For example, in an online video game, the third party game server (which can be a game developer server or other server that supports game play but that is separate from one or more servers implementing a eSports stream management system consistent with implementations of the current subject matter) can provide data about game state features (*e.g.,* key objectives or milestones that a player or players must reach or overcome), scoring approaches, other game characteristics, game state data, and the like. Such data can come directly from a client machine on which a user is playing a game, or it can be provided from the third party server interacting with the client to administer the game being played by the user on the client machine.

[00027] At 110, game state data is received from a first client. The game state data can be captured at the first client rather than, for example, received from a third party game server when the associated video game is an online game. The game state data can include in-game information of a match in the peer-to-peer digital gaming tournament. The game state data can relate to the particular type of game being played and can include any game-specific parameter or metric. For example, game state data can include instantaneous score, change in score over a predefined period of time, character position, character health, objectives reached/overcome, bonuses gained, extra powers or equipment gained or used, and the like. The game state data can include one or more data objects used by and internal to the video game for execution of the match. Game state data includes state data for the game during match play *(e.g.,* in the middle of a match) rather than only a final state of the match *(e.g.,* a final score) and can include data for two or more points in time. In some implementations, the game data can be received during the match *(e.g.,* before the match ends) and may be received periodically *(e.g.,* multiple times during the match, every second, 10 seconds, minute, and the like).

[00028] At 120, game state data from a second client can be received. This second game state data can be from a second match, different and separate from the first match described with respect to step 110. The second match can be part of the peer-to-peer digital game tournament. The second game state data can include in-game information of the second match.

[00029] At 130, the first game state data and the second game state data can be compared. The comparison can be performed using a predefined comparison rule, which can be pre-defined, for example, by a user or tournament host who defines a custom comparison metric. The comparison can include computing a measure of performance of a player in the first match and a player in the second match. For example, in an implementation in which the first and second matches are played substantially concurrently, the measure of performance can include a score for a given interval of time during the middle of game play. The measure of performance can include identifying an event. For example, an in-game achievement such as a "head shot" (seen in some first-person shooters) can be a measure of performance. In some implementations a time period during the match in which the event occurs can also be identified. In some implementations, the predefined comparison rule defines the measures of performance of the players.

[00030] The comparison can be performed at multiple points in time, for example, periodically such as when and if new game state data is received.

[00031] At 140, the comparison can be provided. The providing can include, for example, sending the comparison or a characterization of the relative performance of the players to other tournament participants. As described more fully below, the comparison can be used in an eSports stream management system to automatically recommend portions of video to broadcast or include in a highlights reel. In some implementations, the peer-to-peer tournament can be a peer wagering competition in which peers receive funds based on the outcome of the tournament and the comparison can be provided to a fund escrow server for determining and managing tournament payouts. For example, the participant having the most number of rows cleared in the first two minutes of Tetris play (as determined by an implementation of the current subject matter) can be receive a payout.

[00032] In some implementations, the game state data can be registered in time so that both the first and second game state data references to a common time. For example, if a first match begins play at 12:00 while a second match begins play at 12:06, a comparison between matches at 12:07 may provide some comparison but a user or tournament host may wish to compare the first match at 12:01 to the second match at 12:07 *(e.g.,* when both matches have had 1 minute of gameplay) to provide a more direct comparison. As part of a meaningful comparison, several different common time references can be used. The common time references can include a game time *(e.g.,* measured from the start of game play), a tournament time *(e.g.,* measured from the start of a tournament), an absolute time *(e.g.,* measured to a common clock, such as Coordinated Universal Time (UTC)), and the like.

[00033] In some implementations, the across-match comparison can be used in an eSports stream management system to automatically recommend portions of video to broadcast or include in a highlights reel. The cross-match comparison can provide automatic detection of portions of video or players in the game in which interesting or relevant or entertaining play occurs. In an eSports streaming management system, a first video capture of the first match and from the first client can be received. The video capture can be performed at the client, for example, by capturing user interface screen shots at a predetermine rate, capturing open graphics library (openGL) layers; and capturing data through a pixel buffer and forming video data from the captured data. The eSports streamlining management system can receive a second video capture from the second client of the second match. A portion of the first video capture and/or a portion of the second video capture can be selected using the comparison. By using the across-match comparison, the eSports Management System can identify portions of gameplay that may be interesting to a viewer. For example, at any given point during two matches, viewers may be interested in seeing the greatest "play", which may be identified as the greatest change in score. The selected portion of video capture can be streamed to one or more viewing clients.

[00034] In some implementations, a user or tournament host can specify or define the predefined comparison rule. This allows for customization of the comparison process.

[00035] In some implementations, the matches may be initiated with a common random number generator seed to cause the starting conditions of each match to be identical.

Use of a common random number generator seed can impose fairness in games having elements of randomness in that random elements will start in the same condition. By starting games in the same conditions, a player's performance is less dependent on the random number generator (or conversely, two players will have a level playing field). Using a common random number generator seed further allows for better comparison of game state data and performance between otherwise separate game matches.

[00036] FIG. 2 is a system block diagram illustrating a system 200 for implementing a peer-to-peer digital gaming tournament. The system 200 includes an across-match analytics server 260 that enables comparing disparate game state data from multiple matches.

[00037] A plurality of players 210$_i$ *(i=1, 2, . . . , N)* operate respective player clients 220$_i$. Each player client 220$_i$, includes a third party game application 230$_i$. The game application 230$_i$, can be any online digital game *(e.g.,* video game). In some implementations, multiple players 210, can compete against one another online. Games can be consistent across game applications 230, *(e.g.,* if the players 210, are playing chess, each game application 230, is an instance of an electronic chess game). Each game application 230, can be in communication with and receiving game data from a third party game server 250. The game server 250 provides game data necessary to operate the game. Clients 210, may include mobile devices *(e.g.,* smartphones, tablets, and the like) and the clients 210, third party game server 250, and across-match analytics server 260 can implement the tournament over a network, such as the internet.

[00038] Each game application 230, includes a peer-tournament module 240$_i$. Peer-tournament module 240, integrates into game application 230, and enables players 210, to enroll and participate in an online game competition. Peer-tournament module 240, communicates with and works in tandem with the across-match analytics server 260. While a game application 230, is running a match *(e.g.,* a game instance), the game application 230, has an internal game state that changes over time as the associated player 210, interacts with *(e.g.,* plays) the game. Peer-tournament module 240, can capture the game state data periodically, and/or continuously and transmit the captured game state to the across-match analytics server 260. The across-match analytics server 260 can receive the game state from multiple peer-tournament modules 240, and, as described above with reference to FIG. 1, compare the game states across matches. The matches may be separate in that they do not share a game state. For example, each game application 230, can be an asynchronous single player game.

[00039] In some implementations, the peer-tournament module 240, and the across-match analytics server 260 does not provide game data to the game application 230_i. In other words, in some implementations, they do not contribute to implementing a game instance but rather observe the game, capture game state, and enable a separate layer of multi-match tournament functionality.

[00040] FIG. 3 is a system block diagram of a system 300 with an example eSports Management system 310 utilizing cross-match analytics to provide automatic recommendations for which portions of video to stream of an online eSports tournament. The example system 300 includes the components as described above with reference to FIG. 2 and further includes an eSports management server 310. ESports management server 310 interfaces with a tournament commentator 320, which can be the tournament host, participant, or third party (also referred to as a streamer 320). ESports management server 310 can provide an automated virtual broadcast studio for tournament commentator 320 to cover an eSports tournament. eSports management server 310 receives the cross-match comparison from the across-match analytics server 260 and can automatically break down video into highlight reels and key segments, provide the game analytics to viewing clients 330 and/or players 210, and assemble a suggested broadcast for the tournament commentator 320. eSports management server 310 can broadcast a live video feed (as controlled by tournament commentator 320) to players 210, and/or viewing clients 330.

[00041] In general, video game streamers can engage with their audience by showing live footage of their own gameplay. But in an eSports tournament, mere streaming of a display does not convey what is happening to other players in separate parts of a game or eSports competition. As eSports grow in popularity, the eSports management server 310 solves a need for a toolkit that allows streamers, as individuals, to perform the same level of commentary and gameplay analytics that is performed by analyst teams in the parallel industry of physical sports.

[00042] ESports management server 310 provides an automated broadcast studio for a streamer 320 to cover *(e.g.,* comment on) an eSports tournament in any game. The eSports Streaming Management System has two types of end users. The first is the stream viewer 330. The viewer 330 does not have exposure to stream tools that are available as part of the eSports management server 310. Instead, the viewers 330 are able to see the content shared by the tournament commentator 320. An example of the content shared by the tournament commentator 320 is illustrated in FIG. 4. This includes video replays 410, a feed of the streamer 420, tournament information 430, and current leaderboard status 440.

[00043] The second end type of user is the tournament commentator 320 or "streamer" who uses the ESports management server 310 to deliver a high quality and engaging video feed to their audience. The streamer is able to capture video from all tournament participants; preview the captured video to privately view the replay of a player without sharing that video to the live stream; interact with a video queue interface to view, select, and play videos or clips; and receive suggestions for automatically portioned and assembled video broken into highlight reels and/or key segments with game analytics.

[00044] ESports management server 310 in combination with peer-tournament module 240, automatically captures and records video replay of all players 210, who participate in a tournament. Through the eSports Management System 310, a streamer 320 hosting a tournament has access to a list of all competitors or players 210, in the event that he or she is hosting and each player's 210, associated video capture. For each player 210, the streamer 320 has at least two actions that they can take using the captured video replay. They can preview the captured video in which the streamer 320 privately views the captured video of a player 210, without sharing that video to the live stream. This tool gives the streamer 320 flexibility to review plays before having to speak about them, seek out a specific moment of the replay that he or she wants to show, or otherwise preview the video before showing it to his or her audience. The streamer 320 can also add the selected video to a video queue, which specifies the next video portions to be streamed on the live broadcast stream.

[00045] FIG. 5 is a user interface illustrating a video control panel 500. The video control panel 500 includes a video queue interface 510 for streamers 320 to view, select, and play videos or clips. The video queue interface 510 allows the streamer to interact with and control what is being broadcasted to his or her stream. The video queue interface 510 includes two components: a now playing stream 510B, and a queue 510A.

[00046] The now playing stream 510B shows the streamer 320 what replay is being currently broadcasted (*e.g.,* the live video stream or feed). This is a persistent space that allows the streamer 320 to monitor what they are currently sharing even while using the window to preview streams or adjust replays in a way that will not be shared with the audience.

[00047] The queue 510A gives the streamer 320 the capability to control the order that video portions or replays will be shown. The streamer 320 may add individual video portions that they would like to show next, or change the order of what videos are up next to be broadcasted. If the streamer 320 chooses not to add anything to the queue, or if the queue becomes empty, the system can automatically stream video one of players using the across-match comparison, for example, by streaming video of one of the top three performing players 210, the video of the player 210, having the greatest change in score over a one minute period of time, and the like.

[00048] Video control panel 500 includes a list of players 520 participating in tournament matches and for which video capture is available for broadcast to a wider audience. The list of players 520 includes options to preview 520A or add 520B their replay to the queue 510A. Video control panel 500 further includes a preview space 530 for viewing video and a list of historical game replays 540, with options to preview 540A or add 540B the historical game replay to the queue 510A.

[00049] ESports management server 310, in combination with cross-match comparisons provided by across-match analytics server 260, automatically breaks down video captured at the clients 220, into highlight reels and key segments, provides cross-match game analytics, and assembles a suggested broadcast for the streamer 320. Thus eSports management system 310 enables automated functionality that acts as the streamer's 320 own "behind the scenes analyst team." This allows the streamer 320 to work as a one person team and easily select the correct footage to play without having to take breaks from the live stream to look up past footage or stats. Automatic analytics and features include the rank ordering of replays that causes replays and player lists to appear in the queue 510A in order of their current rank in the tournament. This makes it easy to find the replay for any player based upon where they stand. Prior to watching the video the streamer can know what level of player they are selecting and what the outcome of their game might be. In addition, replays can be ordered according to a custom comparison metric specified by the streamer 320 and used by the across-match analytics server 260 to compare player performance across separate matches.

[00050] Automated eSports management system 310 functionality includes providing a list of historical game replays. This can include links to historical replays of the same game that occurred during previous tournaments. This allows the streamer 320 to bring up their favorite footage for easy reference. For example, the streamer 320 can show the best game ever recorded, keep a reference of themselves playing the game, or use footage with a specifically interesting or relevant play or strategy.

[00051] Automated eSports management system 310 functionality includes providing notifications when there is a change within top of leaderboard. The streamer 320 can be notified (in addition to the leaderboard changing) when there is a change in position amongst top players in a tournament.

[00052] Automated eSports management system 310 functionality can include highlighting of interesting segments. For example, the across-match analytics server 260 can detect changes in game state where change in score per unit time is abnormally high or low. This makes it easy for streamer 320 to sort to key plays where a player 210, secures victory or focus on areas where an otherwise good player struggled and may have lost the game.

[00053] Automated eSports management system 310 functionality can include providing live score updates via segmenting gameplay into discrete blocks of time (sometimes referred to as "heartbeating"). Across-match analytics server 260 detects player 210, scores or other game state information as they play, not just at the end of the tournament. By using this information, eSports management system 310 can post real-time score updates to the streamer 320 and indicate whether or not the player 210, is currently on track to beat their previous score or secure a top spot in the leaderboard.

[00054] As used herein, the term heartbeating refers to collection of snapshots of game state data relating to a specific game instance at regular (or irregular) intervals during the course of game play. The data included in the snapshots can advantageously be structured data and can be sent to the automated eSports management system server for further analysis. The sending of the data from the snapshots collected during a game instance can be sent via a real-time or near real-time streaming approach, or optionally in some examples using an asynchronous approach involving temporarily caching data at the client machine.

[00055] The aforementioned game state data snapshots can be collected at intervals such as several times a second or possibly longer or shorter. Collection of data at this granularity can allow comparisons across multiple game instances. In some examples, such as for example a tournament arrangement, a player competing against one or more other players can be started on a common game "seed" such that the initial game conditions for the multiple competing players are consistent. For a player vs. player example, the multiple players may progress through the game instance at a similar rate. In other examples in which the game does not include direct player vs. player competition, the tournament arrangement is still possible because each player competing in a player vs. game contest is compared to at least one other player experiencing similar starting game conditions. Using the game state data collected via the heartbeating process, the progress of the two or more players who are competing can be compared on a logically chosen progress scale. In some examples, the progress scale may be along a time axis, but in other examples, the progress scale can be distance through a virtual world, number of milestones or objectives encountered/defeated/achieved/etc., or the like.

[00056] Comparison of the game state data snapshots between players in a common game seed can allow predictive analytics to project when key moments in the game (e.g. events or the like that have a heightened impact on which of the two or more players ultimately "wins" the competition) to be identified (or at least predicted with some degree of confidence) before they actually happen. Such predictions can be enabled by analysis and comparison with historical for a larger set of players who have previously played the game or other games that are determined to be similar in structure. In effect, a machine learning algorithm, a regression or correlation analysis, a similarity analysis, or the like can be performed using historical data with comparisons to the "live" or nearly real time game state data snapshots for an in-progress game instance. The current game instance for one or more players can be mapped in this manner such that key states are correlated with those that have occurred in previously completed game instances to thereby enable predictions of likely outcomes, to identify upcoming parts of the game instance that are likely to be highly relevant to the outcome of a competition, and the like.

[00057] In an online video gaming system, unified leaderboard rankings can be determined based on the combination of virtual currency and real currency. In some implementations, the current subject matter provides a translation between cash and virtual currency matches on a shared leaderboard. Because players can play for both cash and virtual currency, there can be a unified leaderboard mixing players who compete in cash tournaments and players who do not compete in cash tournaments. Further, this direct translation can carry over to trophies awarded by the games or by the gaming network. A trophy can be a virtual good, in some cases non-transferable, which can be granted once per player account that achieves some milestone, such as number of matches won, cash games played, changing a setting, and the like.

[00058] In some online gaming networks or systems, a player can compete in one or many games via participation in a leaderboard. This participation may be as an individual or as part of a temporary or permanent alliance of players (relating to the common "clan" structure). Some implementations of the current subject matter can be used to encourage players to participate in multiple network games. Players who do so can have an identity and a presumed alliance, which can provide increased revenue and greater retention across the games played.

[00059] In an example implementation of an online gaming system, players in the online video gaming competition system share a common cash balance across all games that are cash enabled and these balances are treated similarly in terms of value and leadership across a multitude of games. In addition, the example system can offer a "practice" virtual currency, which are specific to each game, and which may or may not be associated with the game's own virtual currency. The system can grant virtual prizes for participation in tournament activities across all games and awards a variety of real-life prizes in a sweepstakes-style system of prizing.

[00060] The current subject matter can include a system of advanced leaderboards; leagues; trophies; and prizing that together provide the network. The system can include a tool that provides a common leaderboard structure, which: provides logic for challenge provision of players in league or non-league circumstances; contemplates and provides rules around surfacing available players to challenge; provides a translation between cash and virtual currency matches on a shared leaderboard; and provides for prizes, including medals (the primary measure of success on the leaderboards), trophies/achievements on both a game and a system-wide basis, non-cash related prizes on the platform layer, and any in-game rewards or prizes assigned on an ad-hoc basis.

[00061] FIG. 6 is an illustration of an example unified leaderboard. The example unified leader board shows a common leaderboard structure that provides a translation between cash and virtual currency matches on a shared leaderboard. This includes a common medal count (right) and ranking (left) on a system-wide basis.

[00062] The current subject matter can provide logic for challenge provision of players in league or non-league circumstances. For example, leaderboards exist across many games and in some multi-game systems (notably Xbox Live gamerscore and related concepts). Using the current subject matter, however, players play across multiple games of every sort and size, and that players are typically competing for: cash outcome on the match; virtual currency outcome on the match, whether system-level and/or a game's individual currency (or currencies); and prizes, which may or may not be cash, virtual currency or even a real-world item. FIG. 7 is a process flow diagram illustrating an example logic flow for defending, challenging, and viewing the leaderboard in league or non-league circumstances. Players can select "play now" from a menu, which will challenge the next opponent. Similarly, a challenged player can defend the challenge. The outcome results in a change in the leaderboard standings.

[00063] The current subject matter can explicitly allow players to compete against each other for both virtual currency and cash, with a valuation currently pegged at a unit of virtual currency being 1/100th of the value of a dollar for purposes of determining the number of medals awarded for the winner of a match. In some implementations, the combination of virtual currency and real currency is used to implement a unified leaderboard.

[00064] These challenges may occur as a random 1 vs. 1 or team vs. team or clan vs. clan match, but they also occur within a league structure, where players are required to challenge other members of their league and this combined method is used to award medals.

[00065] Further, the current subject matter makes the medal awards in a way that is consistent across different games. Thus, winning a match for the same stakes where the players have the same medal counts in different games (*e.g.*, 500 medals) can pay exactly the same number of medals. In addition, virtual currency matches can also be pegged to a standard ratio (currently, virtual currency worth $.01 would be assigned the value of unit of virtual currency), whether the currency is actually available for purchase for cash or not.

[00066] In some implementations, the current subject matter contemplates and provides rules around surfacing available players to challenge. The example system can surfaces suggested matches based on a combination of cash and virtual currency.

[00067] In some implementations, the current subject matter provides a translation between cash and virtual currency matches on a shared leaderboard. Because players can play for both cash and virtual currency, there can be a unified leaderboard mixing players who compete in cash tournaments and players who do not compete in cash tournaments. Further, this direct translation can carry over to trophies awarded by the games or by the gaming network. A trophy can be a virtual good, in some cases non-transferable, which can be granted once per player account that achieves some milestone, such as number of matches won, cash games played, changing a setting, and the like.

[00068] In some implementations, the current subject matter provides for prizes, including medals, which can be a primary measure of success on the leaderboards, trophies/achievements on both a game and a network-wide basis, non-cash prizes on the platform layer, and any in-game rewards or prizes assigned on an ad-hoc basis. For example, FIG. 8 is an illustrating of an example leaderboard prize page.

[00069] As described, some implementations of the current subject matter can offer multiple rewards to the victor(s), on both a game by game and a network-wide system basis. These awards include medals, which are the current measure of a player's rank in any associated leaderboard (global, local, clan, team, friends, and the like) and are properly seen as a measure of skill. Further, the award of trophies/achievements can contain a large set of items unique to the network, where many of these will reward standard actions, including cross-game or network actions. A simple example of the latter is to change one's profile avatar for a particular trophy/achievement, where the action contemplated is only partially relevant to the game in question.

[00070] Although a few variations have been described in detail above, other modifications or additions are possible. For example, although some examples have been described with reference to asynchronous single player games, the current subject matter can apply to any tournament where there are separate matches occurring. These separate matches can be multiplayer matches. For example, the current subject matter can apply to an online tournament of digital tennis in which two players compete directly sharing the same game state but their performance is compared to another match having two different and competing players.

[00071] One or more aspects or features of the subject matter described herein can be realized in digital electronic circuitry, integrated circuitry, specially designed application specific integrated circuits (ASICs), field programmable gate arrays (FPGAs) computer hardware, firmware, software, and/or combinations thereof. These various aspects or features can include implementation in one or more computer programs that are executable and/or interpretable on a programmable system including at least one programmable processor, which can be special or general purpose, coupled to receive data and instructions from, and to transmit data and instructions to, a storage system, at least one input device, and at least one output device. The programmable system or computing system may include clients and servers. A client and server are generally remote from each other and typically interact through a communication network. The relationship of client and server arises by virtue of computer programs running on the respective computers and having a client-server relationship to each other.

[00072] These computer programs, which can also be referred to as programs, software, software applications, applications, components, or code, include machine instructions for a programmable processor, and can be implemented in a high-level procedural language, an object-oriented programming language, a functional programming language, a logical programming language, and/or in assembly/machine language. As used herein, the term "machine-readable medium" refers to any computer program product, apparatus and/or device, such as for example magnetic discs, optical disks, memory, and Programmable Logic Devices (PLD5), used to provide machine instructions and/or data to a programmable processor, including a machine-readable medium that receives machine instructions as a machine-readable signal. The term "machine-readable signal" refers to any signal used to provide machine instructions and/or data to a programmable processor. The machine-readable medium can store such machine instructions non-transitorily, such as for example as would a non-transient solid-state memory or a magnetic hard drive or any equivalent storage medium. The machine-readable medium can alternatively or additionally store such machine instructions in a transient manner, such as for example as would a processor cache or other random access memory associated with one or more physical processor cores.

[00073] To provide for interaction with a user, one or more aspects or features of the subject matter described herein can be implemented on a computer having a display device, such as for example a cathode ray tube (CRT) or a liquid crystal display (LCD) or a light emitting diode (LED) monitor for displaying information to the user and a keyboard and a pointing device, such as for example a mouse or a trackball, by which the user may provide input to the computer. Other kinds of devices can be used to provide for interaction with a user as well. For example, feedback provided to the user can be any form of sensory feedback, such as for example visual feedback, auditory feedback, or tactile feedback; and input from the user may be received in any form, including, but not limited to, acoustic, speech, or tactile input. Other possible input devices include, but are not limited to, touch screens or other touch-sensitive devices such as single or multi-point resistive or capacitive trackpads, voice recognition hardware and software, optical scanners, optical pointers, digital image capture devices and associated interpretation software, and the like.

[00074] In the descriptions above and in the claims, phrases such as "at least one of or "one or more of may occur followed by a conjunctive list of elements or features. The term "and/or" may also occur in a list of two or more elements or features. Unless otherwise implicitly or explicitly contradicted by the context in which it is used, such a phrase is intended to mean any of the listed elements or features individually or any of the recited elements or features in combination with any of the other recited elements or features. For example, the phrases "at least one of A and B;" "one or more of A and B;" and "A and/or B" are each intended to mean "A alone, B alone, or A and B together." A similar interpretation is also intended for lists including three or more items. For example, the phrases "at least one of A, B, and C;" "one or more of A, B, and C;" and "A, B, and/or C" are each intended to mean "A alone, B alone, C alone, A and B together, A and C together, B and C together, or A and B and C together." In addition, use of the term "based on," above and in the claims is intended to mean, "based at least in part on," such that an unrecited feature or element is also permissible.

[00075] The subject matter described herein can be embodied in systems, apparatus, methods, and/or articles depending on the desired configuration. The implementations set forth in the foregoing description do not represent all implementations consistent with the subject matter described herein. Instead, they are merely some examples consistent with aspects related to the described subject matter. Although a few variations have been described in detail above, other modifications or additions are possible. In particular, further features and/or variations can be provided in addition to those set forth herein. For example, the implementations described above can be directed to various combinations and subcombinations of the disclosed features and/or combinations and subcombinations of several further features disclosed above. In addition, the logic flows depicted in the accompanying figures and/or described herein do not necessarily require the particular order shown, or sequential order, to achieve desirable results. Other implementations may be within the scope of the following claims.

WHAT IS CLAIMED IS:

1. A method comprising:

receiving first game state data captured from a first client and characterizing in-game information of a first match of a peer-to-peer digital gaming tournament;

receiving second game state data captured from a second client and characterizing in-game information of a second match of the peer-to-peer digital gaming tournament, the first match and the second match being separate gaming instances, the first game state data captured during the first match and representative of an internal state of a digital video game;

comparing the first game state data and the second game state data using a predefined comparison rule by computing a measure of a performance of a first player in the first match with a measure of a performance of a second player in the second match; and

providing the comparison;

wherein at least one of the receiving, comparing, and providing is performed by at least one data processor forming part of at least one computing system.

2. The method of claim 1 wherein the predefined comparison rule defines the measure of the performance of the first player and the measure of the performance of the second player.

3. The method of claim 1 further comprising:

receiving a first video capture from the first client of a display of the first match;

receiving a second video capture from the second client of a display of the second match;

selecting, based on the comparison, a portion of the first video capture or a portion of the second video capture; and

streaming at least one of the portion of the first video capture and the portion of the second video capture to at least one viewing client.

4. The method of claim 1 further comprising:

comparing the first game state data and the second game state data periodically and at multiple time points.

5. The method of claim 1, wherein the first match and the second match begin at different times, the method further comprising:

registering in time the first game state data to the second game state data, wherein time is referenced to a game time, an absolute time, or an tournament time.

6. The method of claim 1 wherein the first game state data characterizes a value of one or more in-game data objects at two or more points of time during the first match.

7. The method of claim 1 wherein the first client receives game data from a third party game server, the game data for implementing the first match.

8. The method of claim 1 wherein the first match and the second match are separate instances of the digital video game.

9. The method of claim 8 wherein the digital video game is an asynchronous game.

10. The method of claim 8 wherein the digital video game is a single player asynchronous game.

11. The method of claim 1 wherein the first match and the second match are played concurrently.

12. The method of claim 1 wherein the predefined comparison rule is previously defined by a user.

13. The method of claim 1 wherein the comparing includes identifying at least one period of time during the first match or the second match in which an event occurs.

14. The method of claim 1 wherein the receiving of the first game state data occurs during the first match.

15. The method of claim 14 wherein the receiving of the first game state data occurs periodically during the first match.

16. The method of claim 1 wherein the first game state data includes: instantaneous score, change in score over a predefined period of time, character position, character health.

17. The method of claim 1 wherein the first match and the second match are initiated with a common random number seed to provide common starting conditions.

18. The method of claim 1 wherein the first client and the second client are separate computing systems in communication with a third party game server and an analytics server via a network, the third party game server separate from the analytics server, at least one of the receiving, comparing, and providing is performed by the analytics server.

19. A computer program product comprising a non-transitory machine-readable medium storing instructions that, when executed by one or more programmable processors, cause the one or more programmable processors to perform operations as described in any preceding claim.

20. A system comprising a machine-readable medium storing instructions and one or more programmable processors configured to perform operations as described in any of claims 1 to 18.

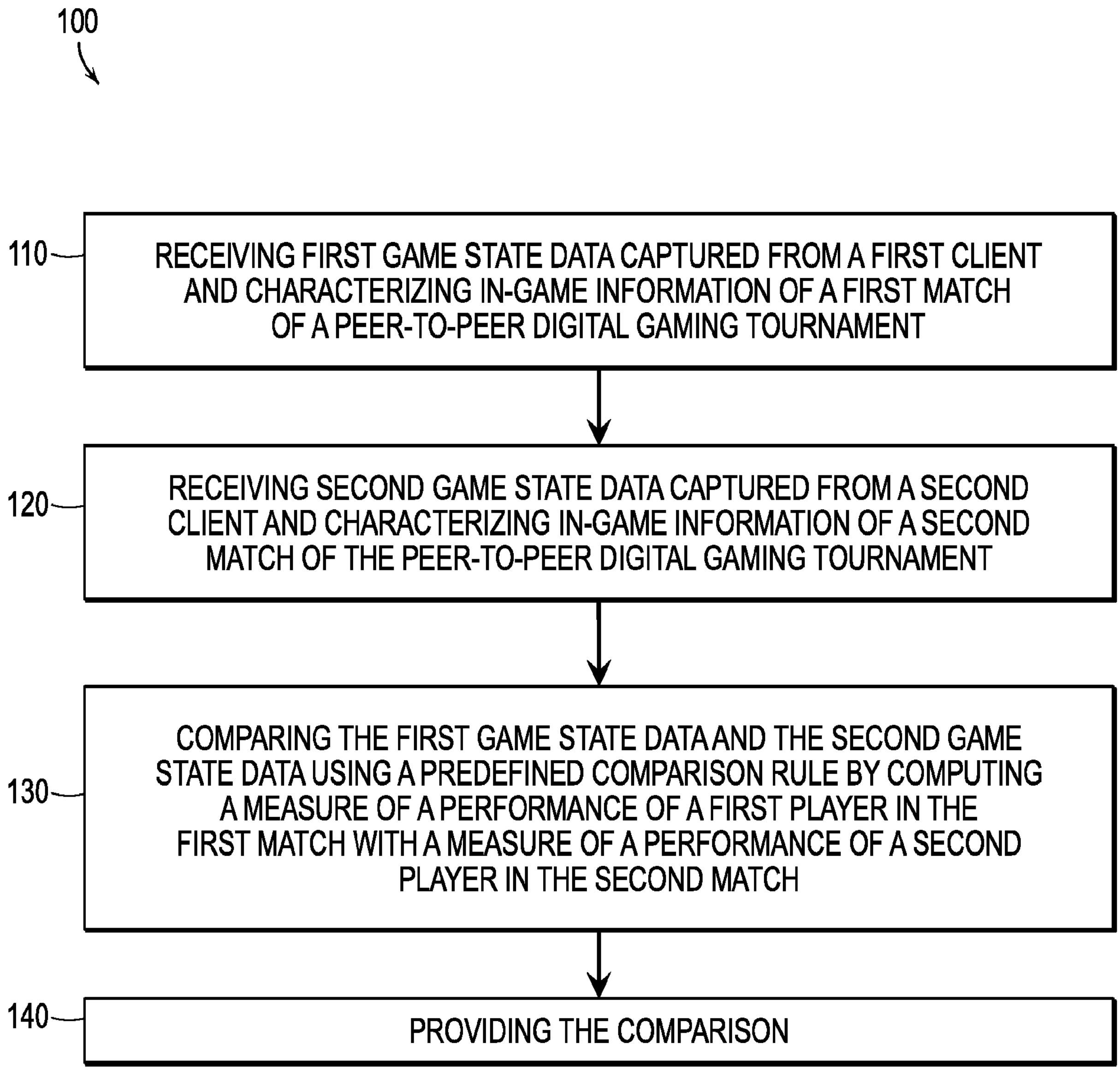

100
110
RECEIVING FIRST GAME STATE DATA CAPTURED FROM A FIRST CLIENT AND CHARACTERIZING IN-GAME INFORMATION OF A FIRST MATCH OF A PEER-TO-PEER DIGITAL GAMING TOURNAMENT
120
RECEIVING SECOND GAME STATE DATA CAPTURED FROM A SECOND CLIENT AND CHARACTERIZING IN-GAME INFORMATION OF A SECOND MATCH OF THE PEER-TO-PEER DIGITAL GAMING TOURNAMENT
130
COMPARING THE FIRST GAME STATE DATA AND THE SECOND GAME STATE DATA USING A PREDEFINED COMPARISON RULE BY COMPUTING A MEASURE OF A PERFORMANCE OF A FIRST PLAYER IN THE FIRST MATCH WITH A MEASURE OF A PERFORMANCE OF A SECOND PLAYER IN THE SECOND MATCH
140
PROVIDING THE COMPARISON

FIG. 1

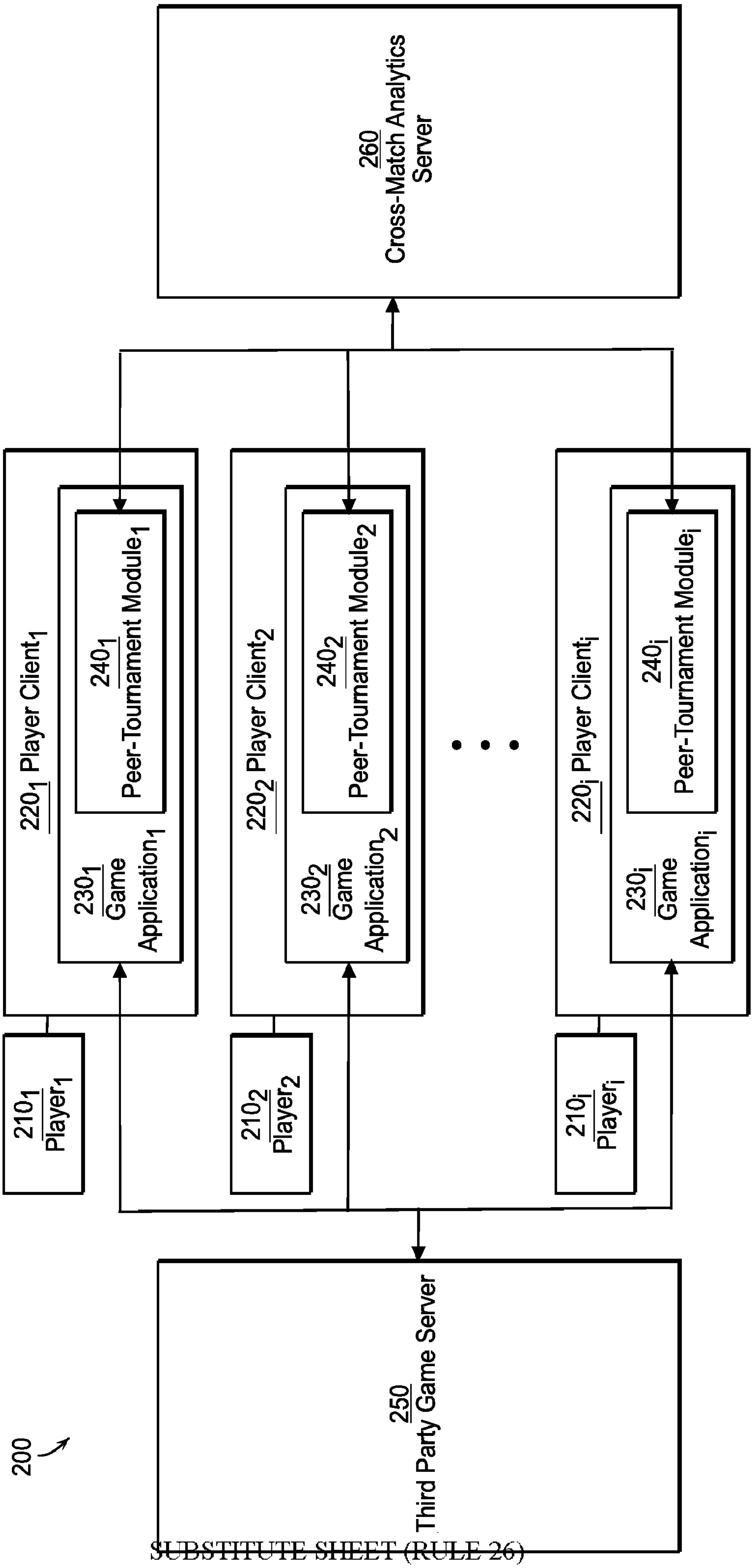

260
Cross-Match Analytics Server
220_1 Player Client_1
230_1 Game Application_1
240_1 Peer-Tournament Module_1
220_2 Player Client_2
230_2 Game Application_2
240_2 Peer-Tournament Module_2
220_i Player Client_i
230_i Game Application_i
240_i Peer-Tournament Module_i
210_1 Player_1
210_2 Player_2
210_i Player_i
250
Third Party Game Server
200

FIG. 2

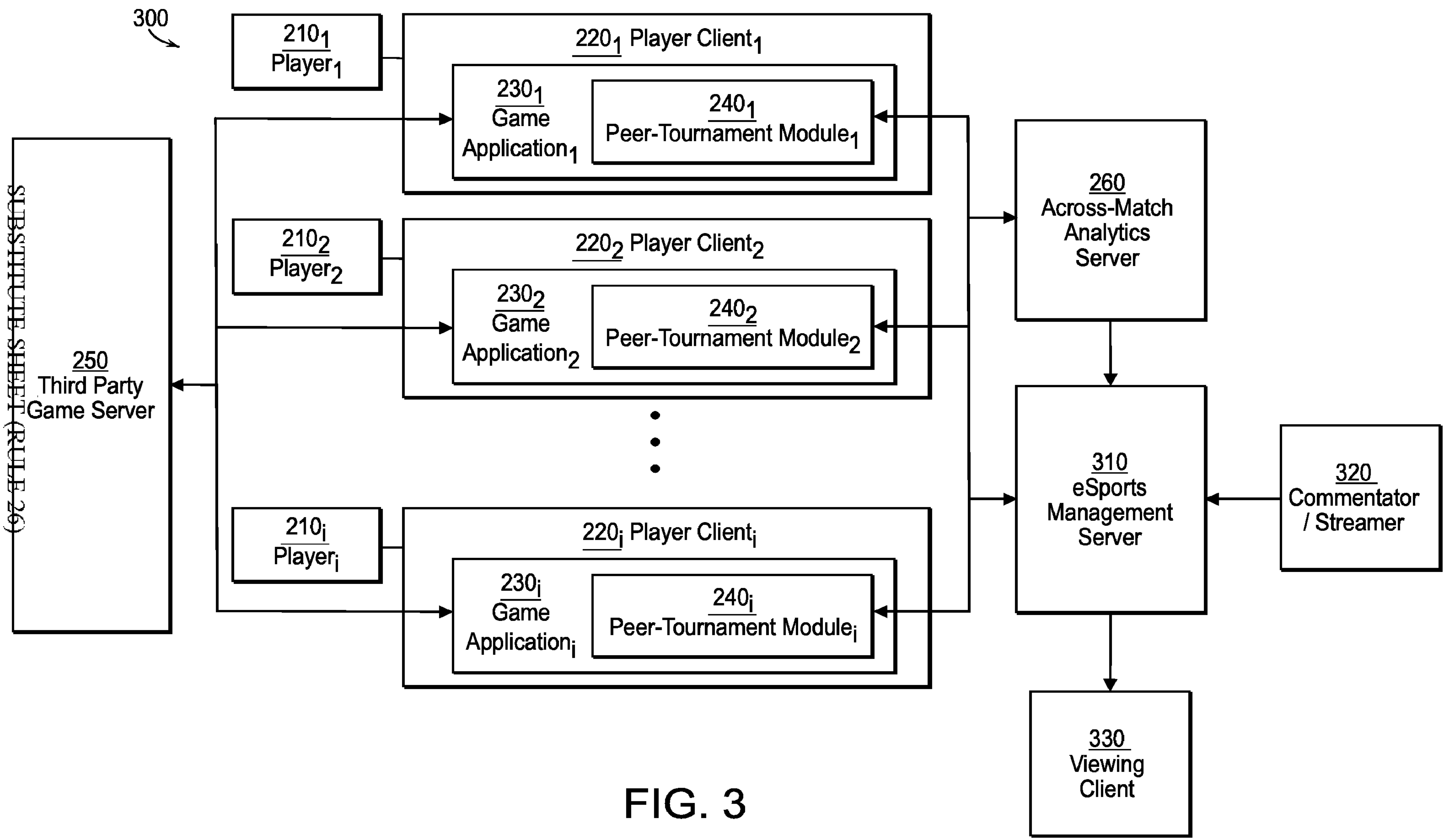
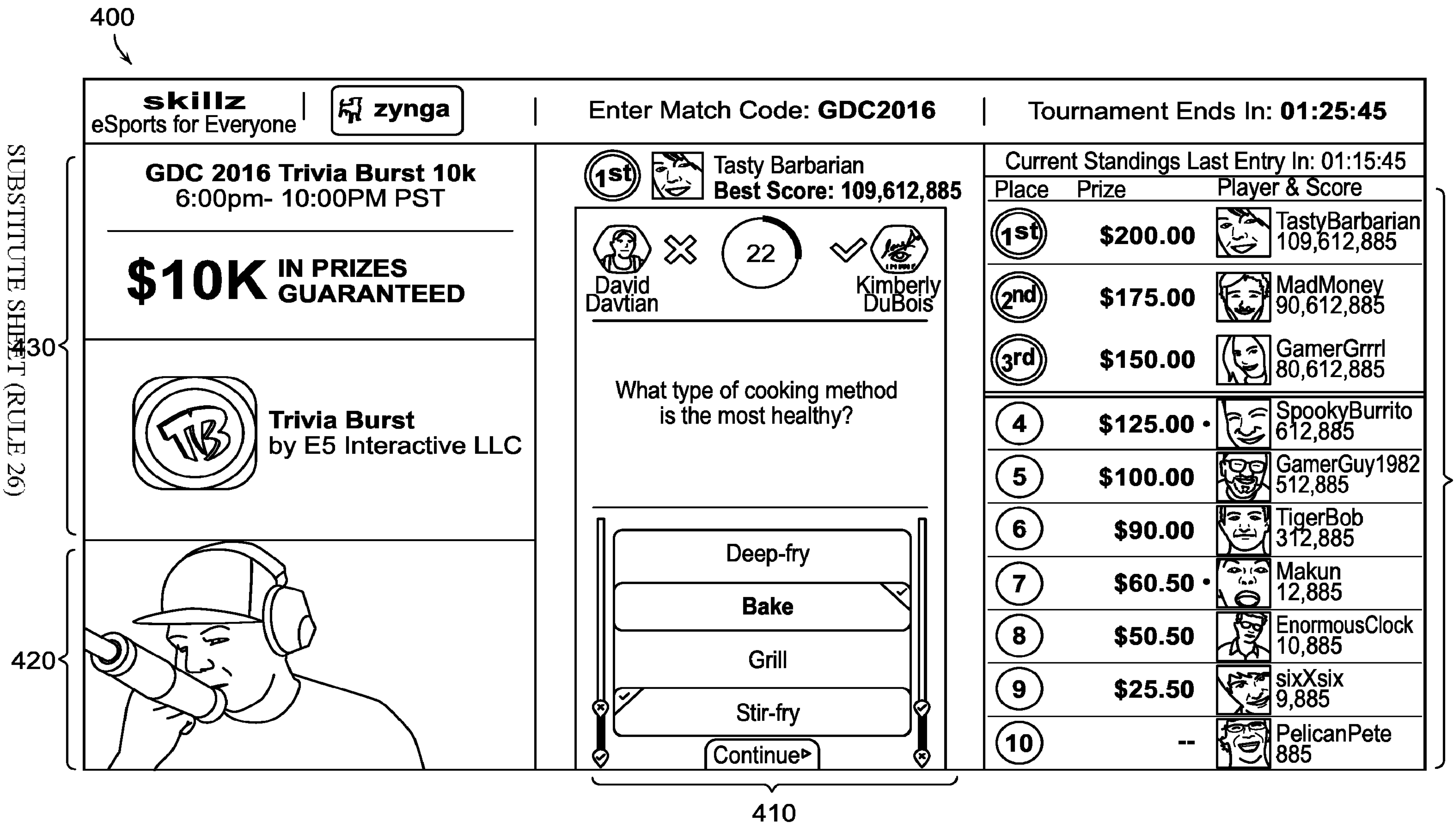

300
210_1 $Player_1$
220_1 Player $Client_1$
230_1 Game $Application_1$
240_1 Peer-Tournament $Module_1$
210_2 $Player_2$
220_2 Player $Client_2$
230_2 Game $Application_2$
240_2 Peer-Tournament $Module_2$
210_i $Player_i$
220_i Player $Client_i$
230_i Game $Application_i$
240_i Peer-Tournament $Module_i$
250 Third Party Game Server
260 Across-Match Analytics Server
310 eSports Management Server
320 Commentator / Streamer
330 Viewing Client

FIG. 3

FIG. 4



Enter Match Code: **GDC2016**

Tournament Ends In: **01:25:45**

GDC 2016 Trivia Burst 10k
6:00pm- 10:00PM PST

$10K IN PRIZES GUARANTEED

Trivia Burst
by E5 Interactive LLC

1st Tasty Barbarian
Best Score: 109,612,885

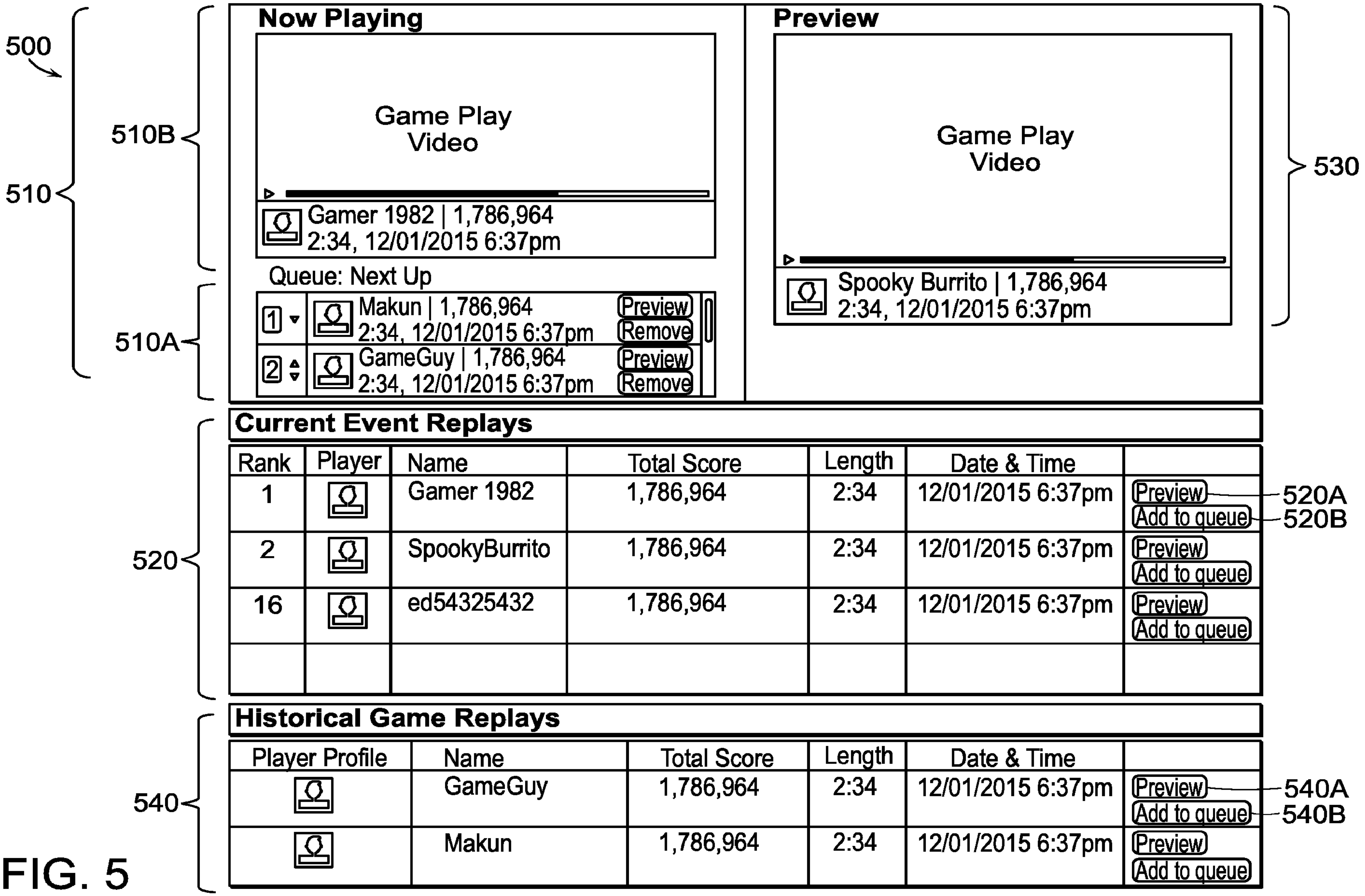
David Davtian ✗ 22 ✓ Kimberly DuBois

What type of cooking method is the most healthy?

- Deep-fry
- **Bake**
- Grill
- Stir-fry

Current Standings Last Entry In: 01:15:45

Place	Prize	Player & Score
1st	$200.00	TastyBarbarian 109,612,885
2nd	$175.00	MadMoney 90,612,885
3rd	$150.00	GamerGrrrl 80,612,885
4	$125.00 •	SpookyBurrito 612,885
5	$100.00	GamerGuy1982 512,885
6	$90.00	TigerBob 312,885
7	$60.50 •	Makun 12,885
8	$50.50	EnormousClock 10,885
9	$25.50	sixXsix 9,885
10	--	PelicanPete 885



Now Playing

Gamer 1982 | 1,786,964
2:34, 12/01/2015 6:37pm

Queue: Next Up

1 Makun | 1,786,964
2:34, 12/01/2015 6:37pm

2 GameGuy | 1,786,964
2:34, 12/01/2015 6:37pm

Preview

Spooky Burrito | 1,786,964
2:34, 12/01/2015 6:37pm

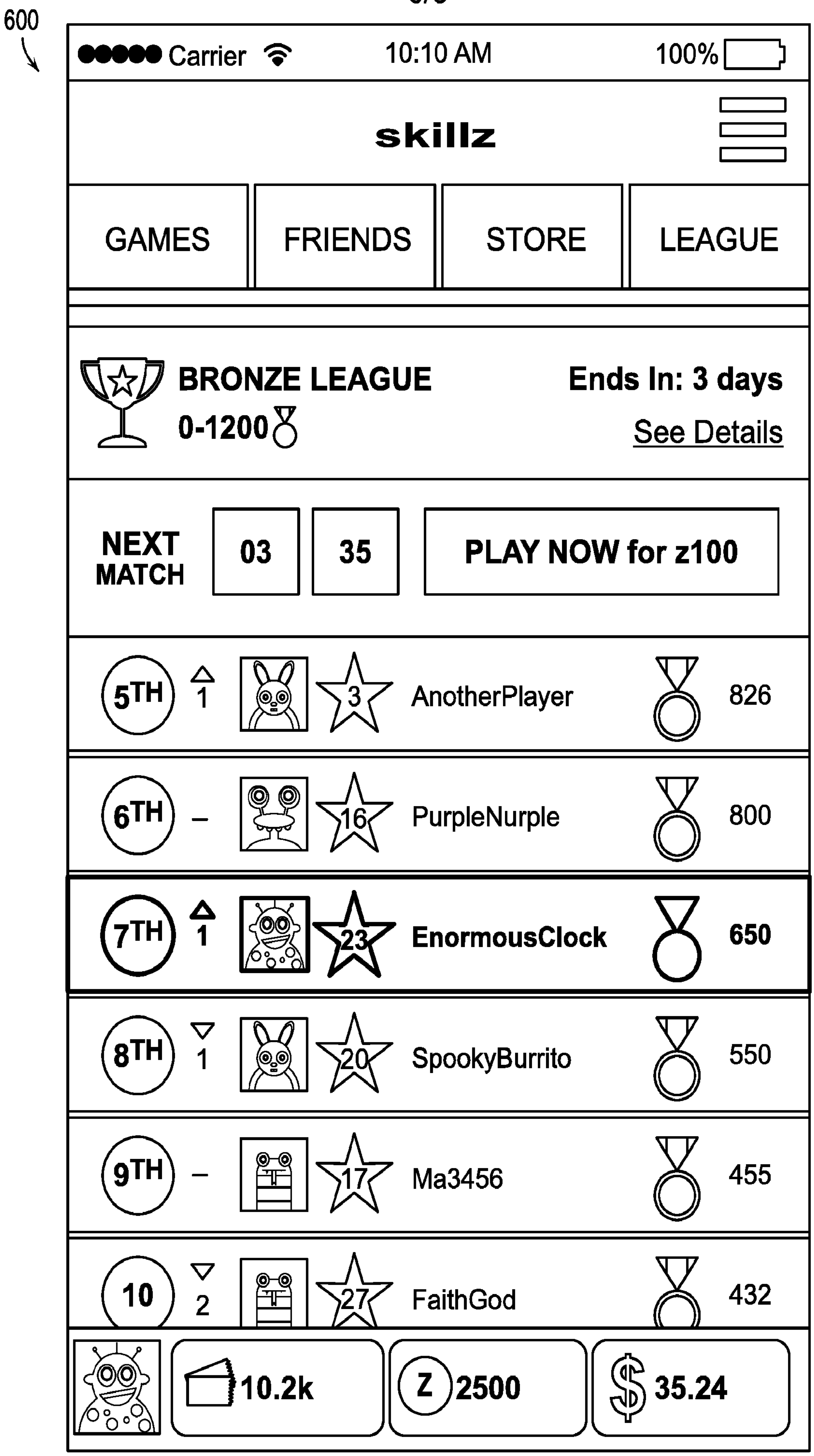
Current Event Replays

Rank	Player	Name	Total Score	Length	Date & Time	
1		Gamer 1982	1,786,964	2:34	12/01/2015 6:37pm	Preview Add to queue
2		SpookyBurrito	1,786,964	2:34	12/01/2015 6:37pm	Preview Add to queue
16		ed54325432	1,786,964	2:34	12/01/2015 6:37pm	Preview Add to queue

Historical Game Replays

Player Profile	Name	Total Score	Length	Date & Time	
	GameGuy	1,786,964	2:34	12/01/2015 6:37pm	Preview Add to queue
	Makun	1,786,964	2:34	12/01/2015 6:37pm	Preview Add to queue

FIG. 5



●●●●● Carrier 10:10 AM 100%

skillz

GAMES FRIENDS STORE LEAGUE

BRONZE LEAGUE
0-1200

Ends In: 3 days
See Details

NEXT MATCH 03 35 PLAY NOW for z100

5TH	△ 1	3	AnotherPlayer	826
6TH	–	16	PurpleNurple	800
7TH	**△ 1**	**23**	**EnormousClock**	**650**
8TH	▽ 1	20	SpookyBurrito	550
9TH	–	17	Ma3456	455
10	▽ 2	27	FaithGod	432

10.2k Z 2500 $ 35.24

FIG. 6

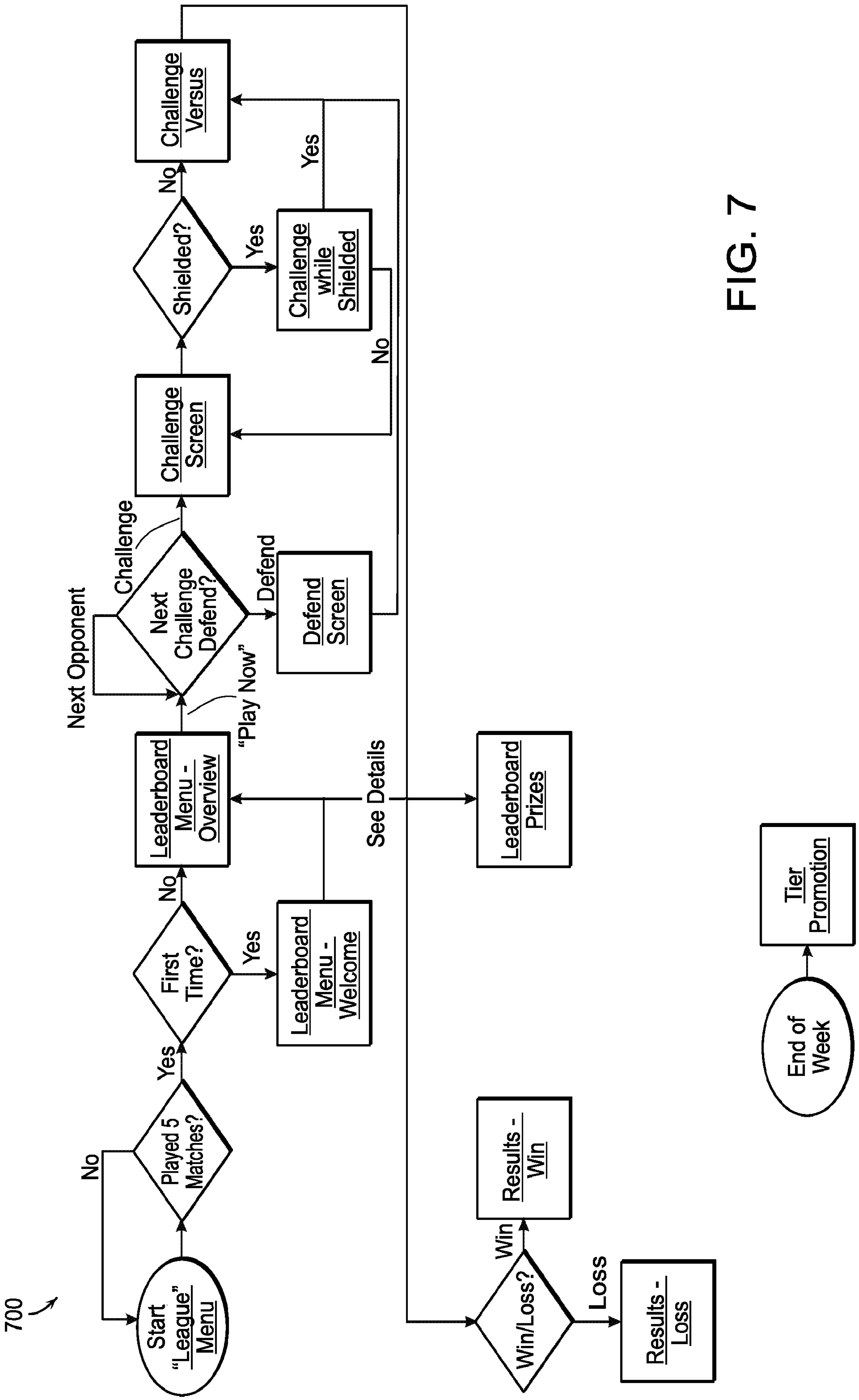
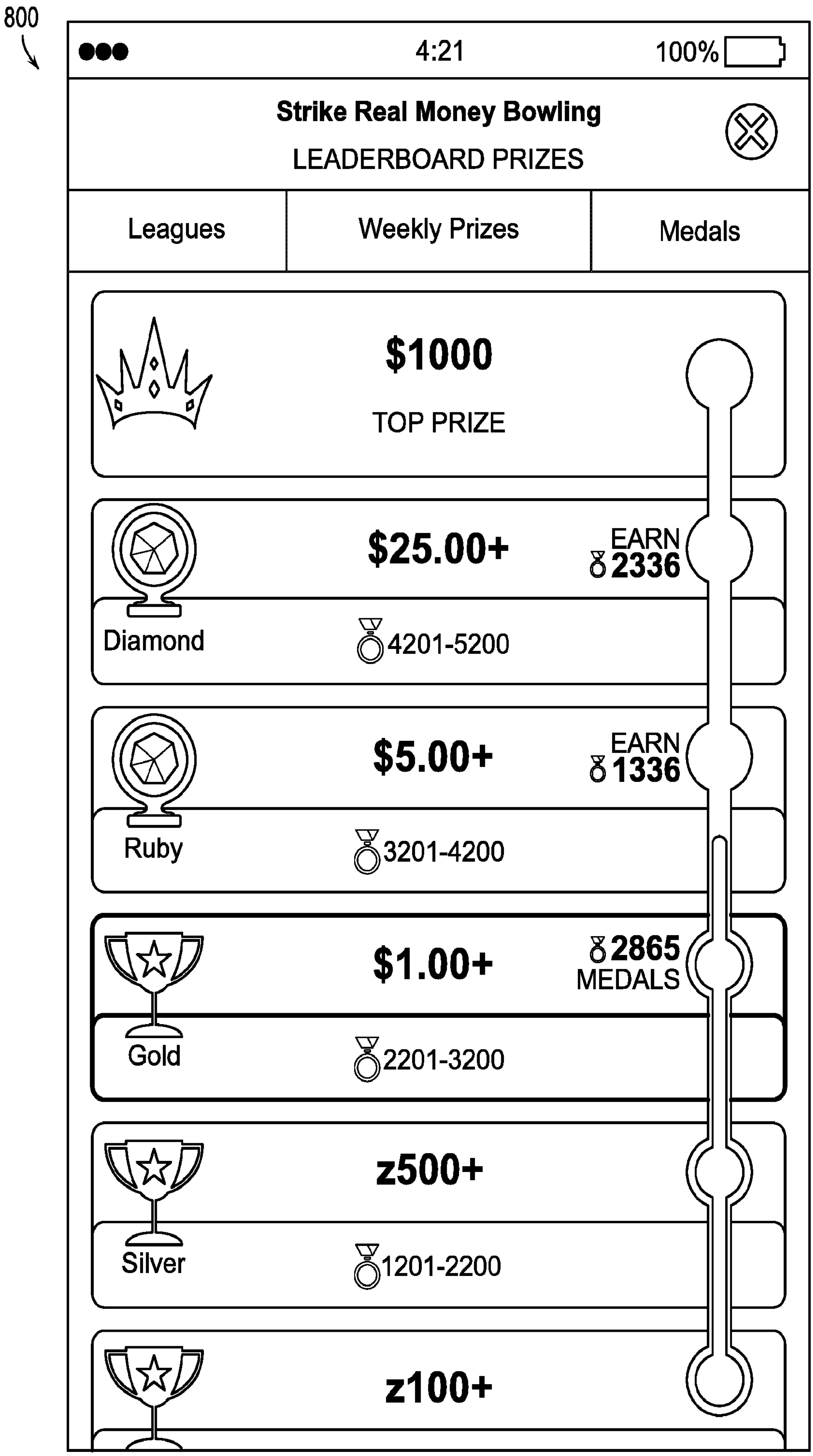

700
Start "League" Menu
Played 5 Matches?
No
Yes
First Time?
Yes
No
Leaderboard Menu - Welcome
Leaderboard Menu - Overview
See Details
Leaderboard Prizes
"Play Now"
Next Opponent
Next Challenge Defend?
Challenge
Defend
Defend Screen
Challenge Screen
Shielded?
Yes
No
Challenge while Shielded
Yes
No
Challenge Versus
Win/Loss?
Win
Loss
Results - Win
Results - Loss
End of Week
Tier Promotion

FIG. 7



Strike Real Money Bowling

LEADERBOARD PRIZES

$1000

TOP PRIZE

Diamond	**$25.00+** 4201-5200	EARN **2336**
Ruby	**$5.00+** 3201-4200	EARN **1336**
Gold	**$1.00+** 2201-3200	**2865** MEDALS
Silver	**z500+** 1201-2200	
	z100+	

FIG. 8