



(19) **United States**

(12) **Patent Application Publication**
Paradise et al.

(10) **Pub. No.: US 2024/0185673 A1**

(43) **Pub. Date: Jun. 6, 2024**

(54) **PEER-TO-PEER WAGERING PLATFORM**

Apr. 8, 2013, now Pat. No. 8,715,077, which is a continuation-in-part of application No. 13/569,424, filed on Aug. 8, 2012, now Pat. No. 8,414,387.

(71) Applicant: **Skillz Platform Inc.**, San Francisco, CA (US)

(72) Inventors: **Andrew Paradise**, San Francisco, CA (US); **Casey Chafkin**, Boston, MA (US); **Jason Petralia**, Winchester, MA (US); **Ahmed Abdalla**, North Attleboro, MA (US)

(21) Appl. No.: **18/415,562**

(22) Filed: **Jan. 17, 2024**

Publication Classification

(51) **Int. Cl.**
G07F 17/32 (2006.01)
A63F 13/00 (2006.01)
A63F 13/61 (2006.01)
A63F 13/80 (2006.01)
(52) **U.S. Cl.**
CPC **G07F 17/3223** (2013.01); **A63F 13/00** (2013.01); **A63F 13/61** (2014.09); **A63F 13/80** (2014.09); **G07F 17/323** (2013.01); **G07F 17/3244** (2013.01); **G07F 17/3276** (2013.01)

Related U.S. Application Data

(63) Continuation of application No. 18/058,424, filed on Nov. 23, 2022, now Pat. No. 11,915,548, which is a continuation of application No. 16/992,618, filed on Aug. 13, 2020, now Pat. No. 11,551,515, which is a continuation of application No. 16/382,890, filed on Apr. 12, 2019, now Pat. No. 10,748,373, which is a continuation of application No. 15/595,794, filed on May 15, 2017, now Pat. No. 10,290,179, which is a continuation of application No. 14/976,569, filed on Dec. 21, 2015, now Pat. No. 9,649,564, which is a continuation of application No. 14/212,651, filed on Mar. 14, 2014, now Pat. No. 9,240,101, which is a continuation of application No. 13/858,581, filed on

(57) **ABSTRACT**

Data characterizing historical skills-based gaming metrics for a first user and historical skills-based gaming metrics for at least one second user is accessed. Using the accessed data and a set of rules, a targeted advertisement to present to the first user is determined. The targeted advertisement specifies at least one skills-based game and a characterization of the at least one second user's historical skills-based gaming metrics. The targeted advertisement is generated. Data characterizing the targeted advertisement is provided. Related apparatus, systems, techniques, and articles are also described.

1200



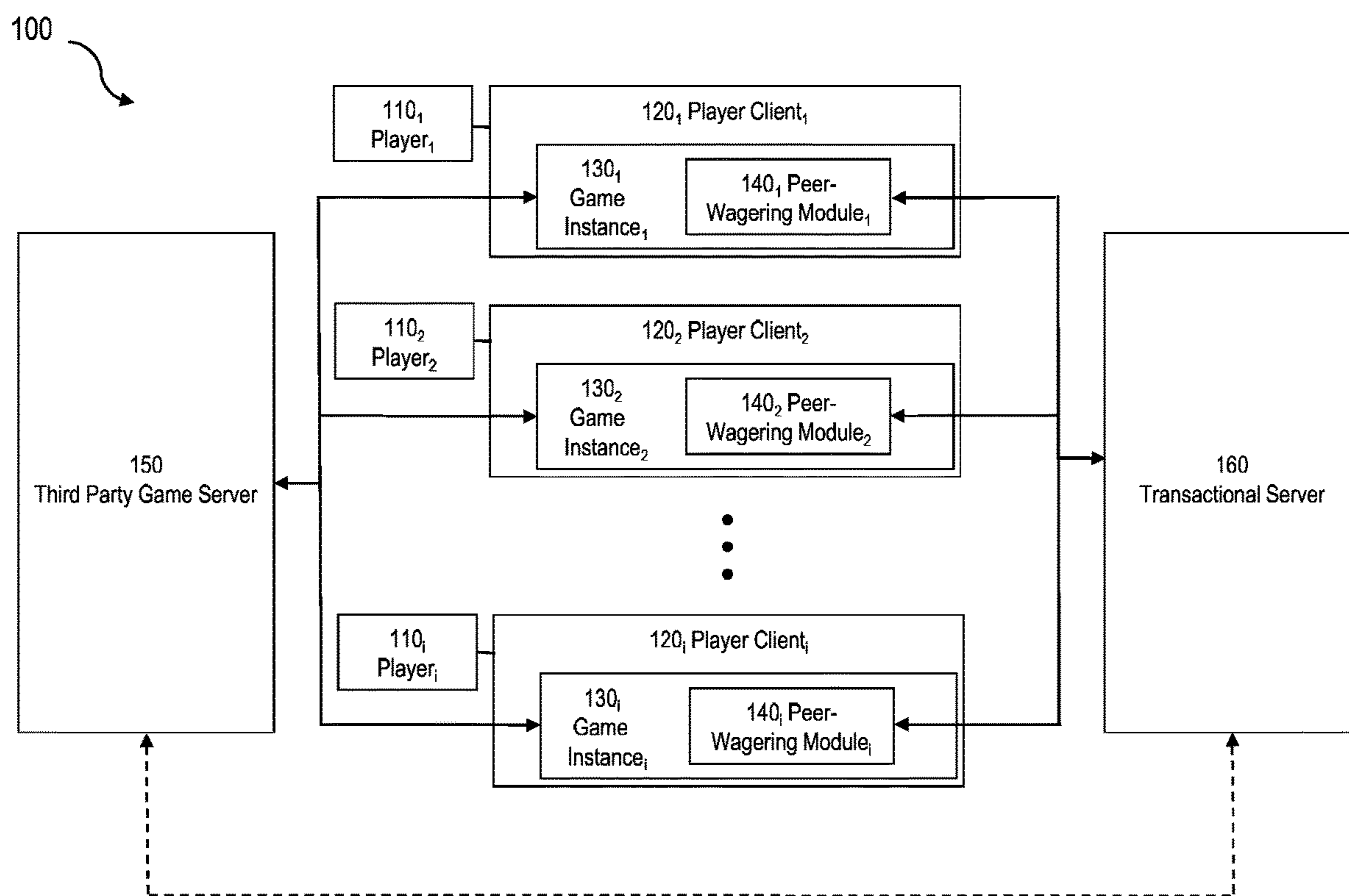


FIG. 1

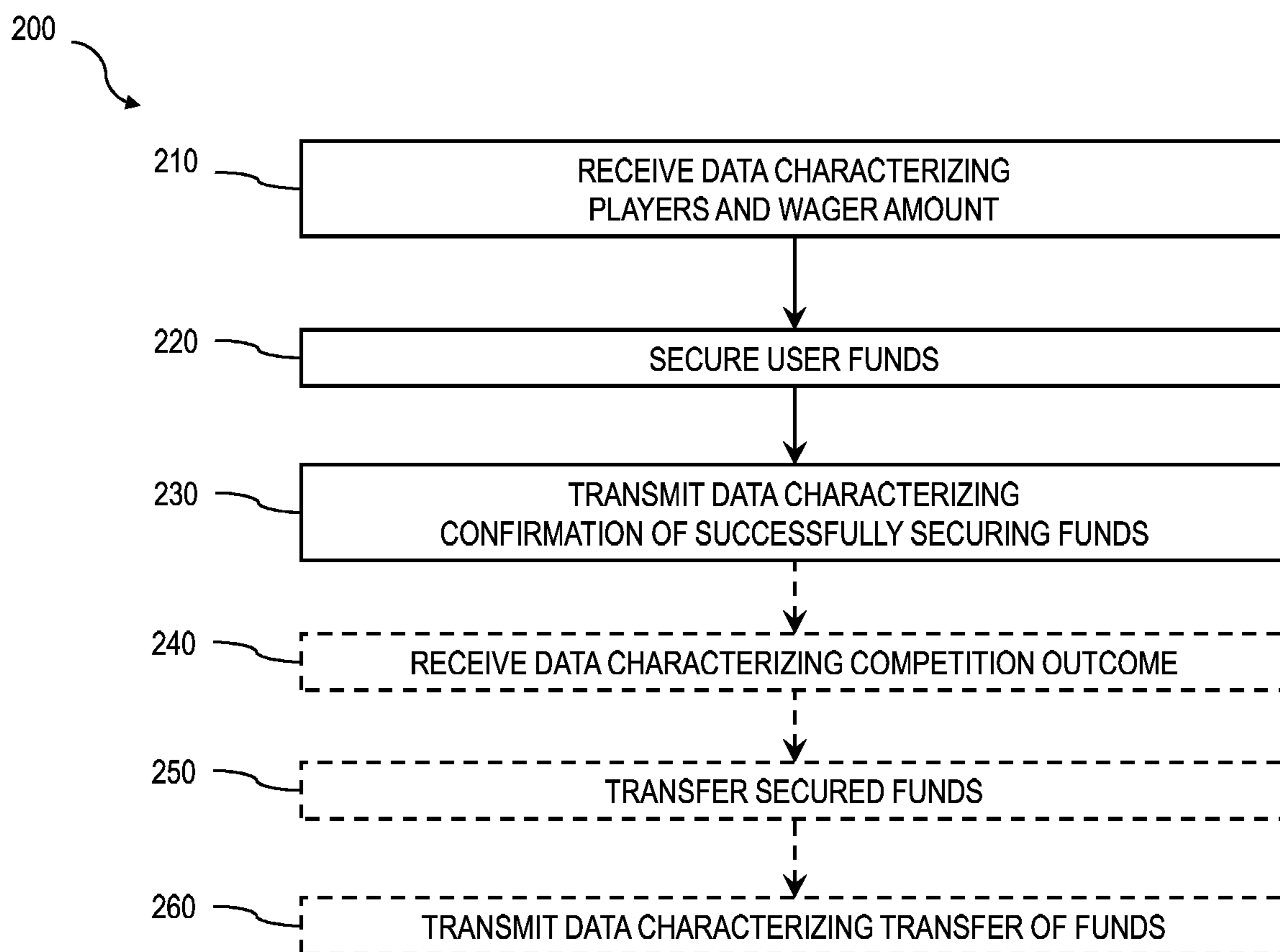


FIG. 2

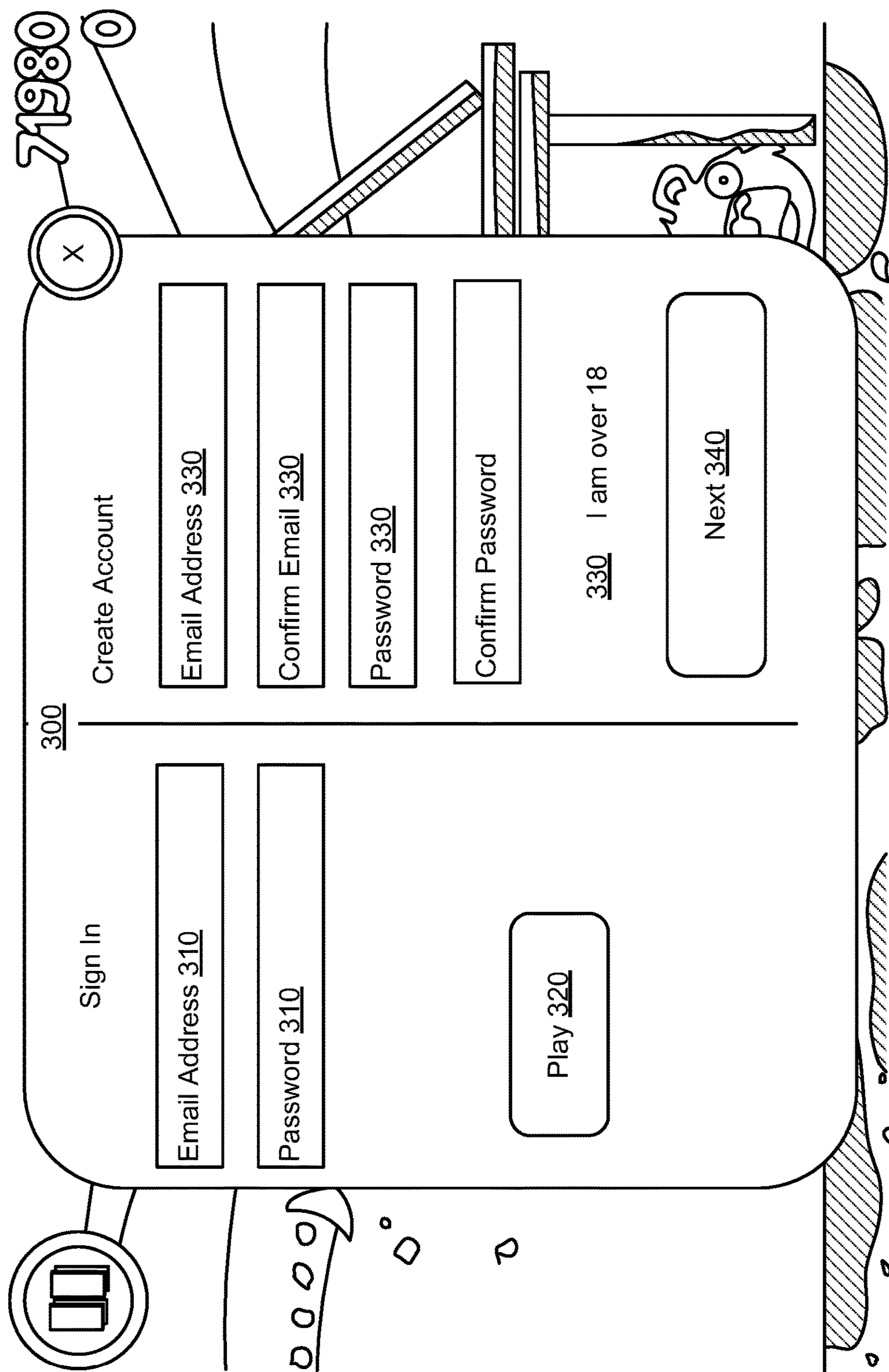


FIG. 3

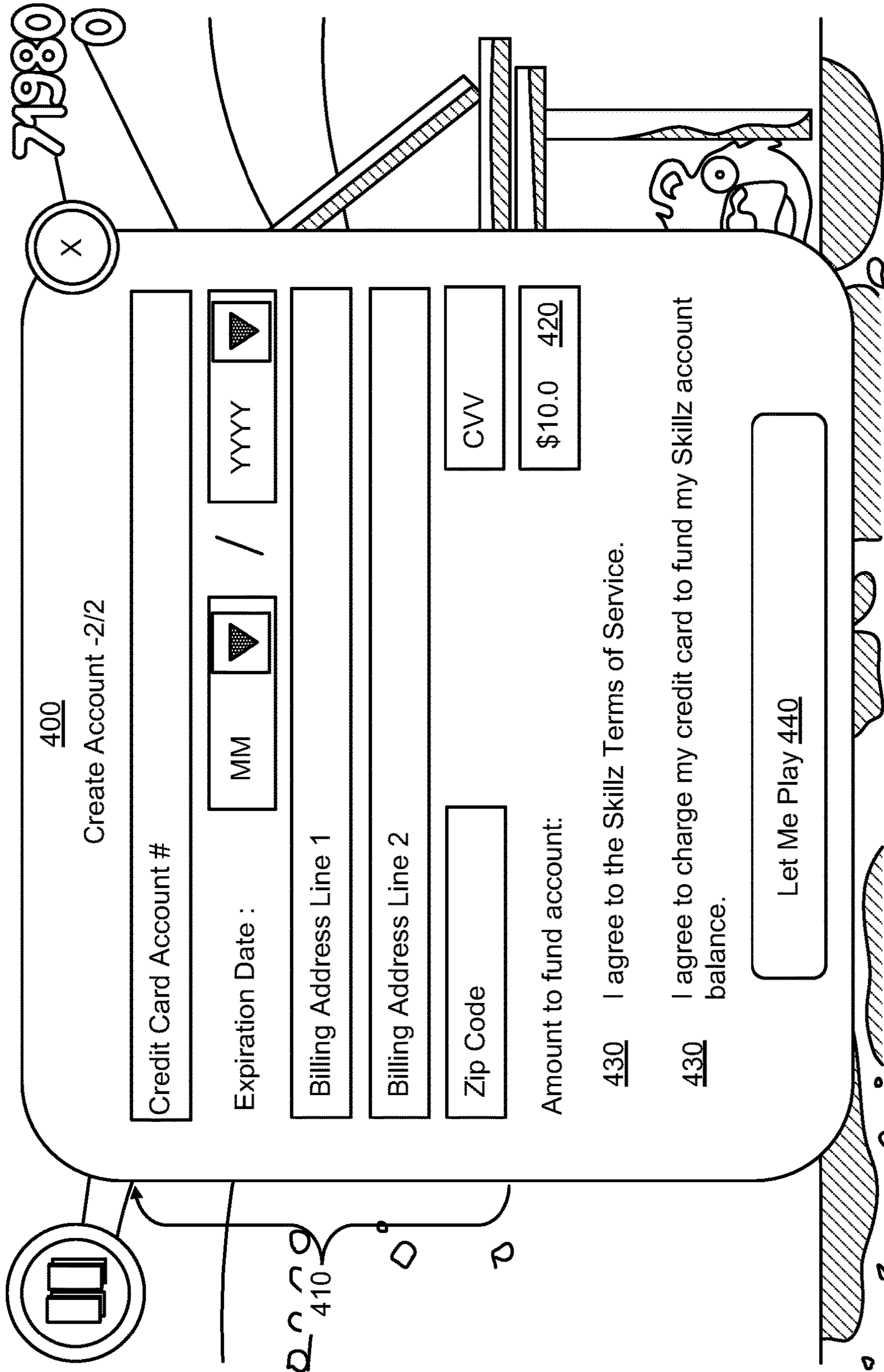


FIG. 4

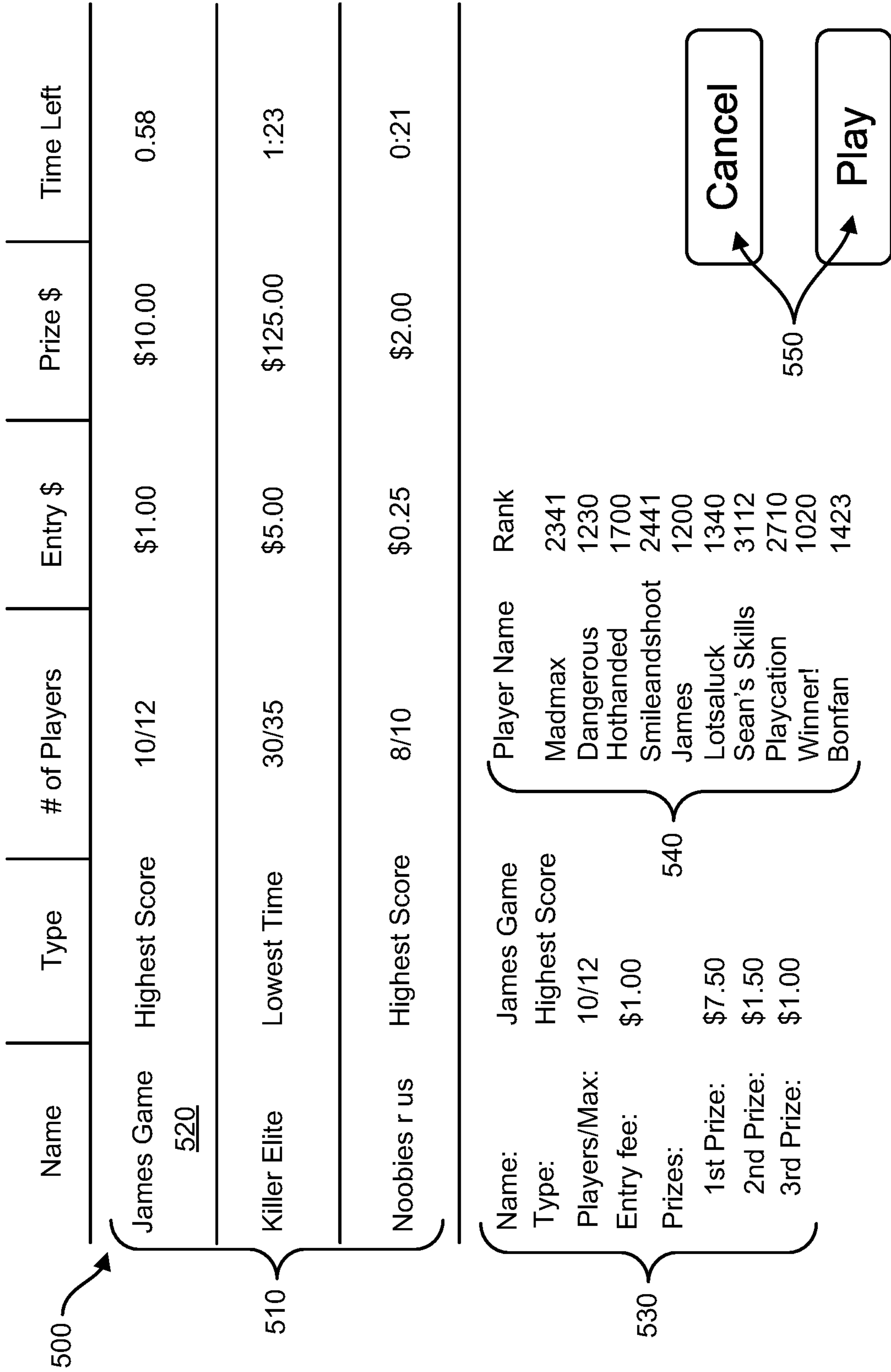


FIG. 5

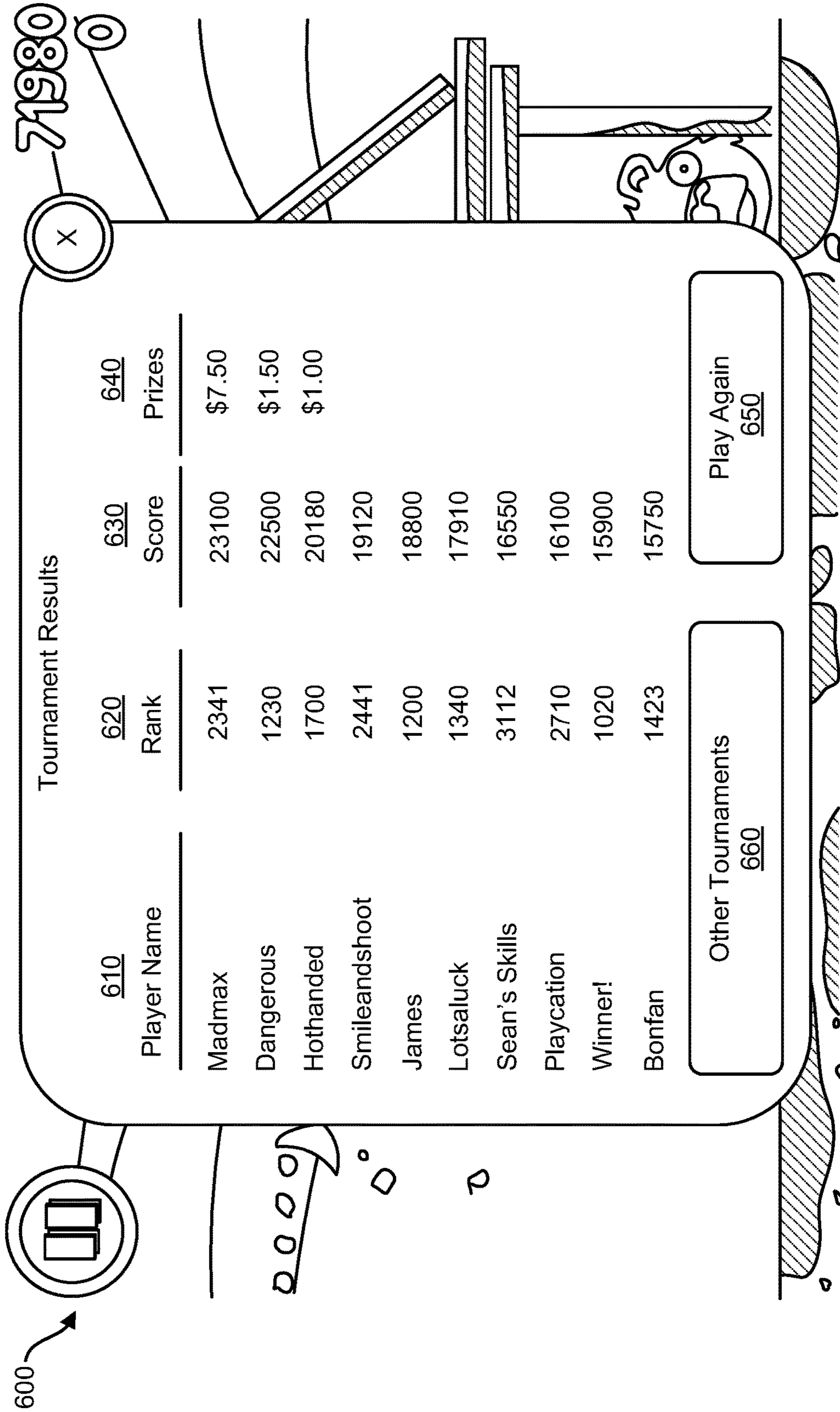


FIG. 6

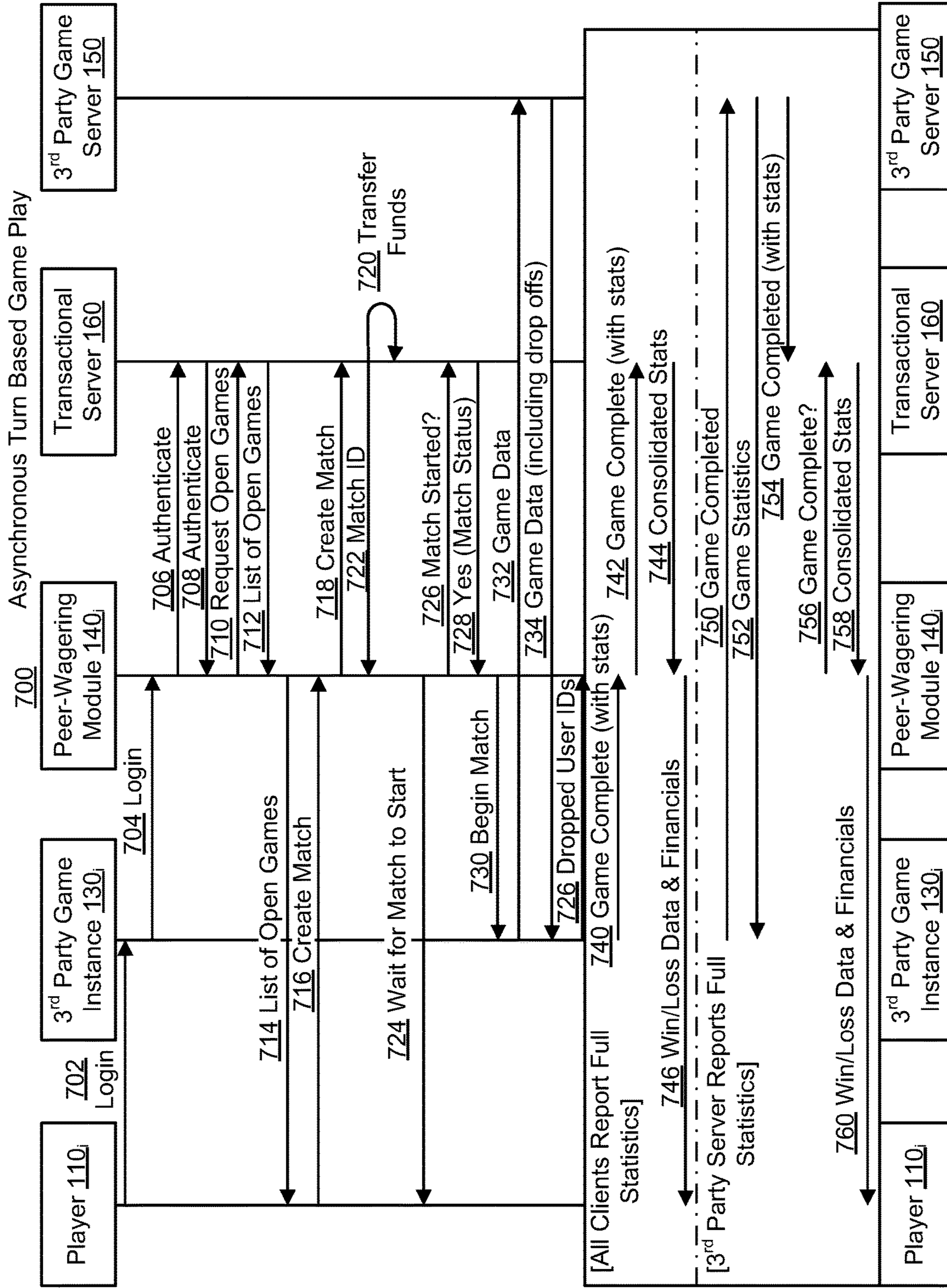


FIG. 7

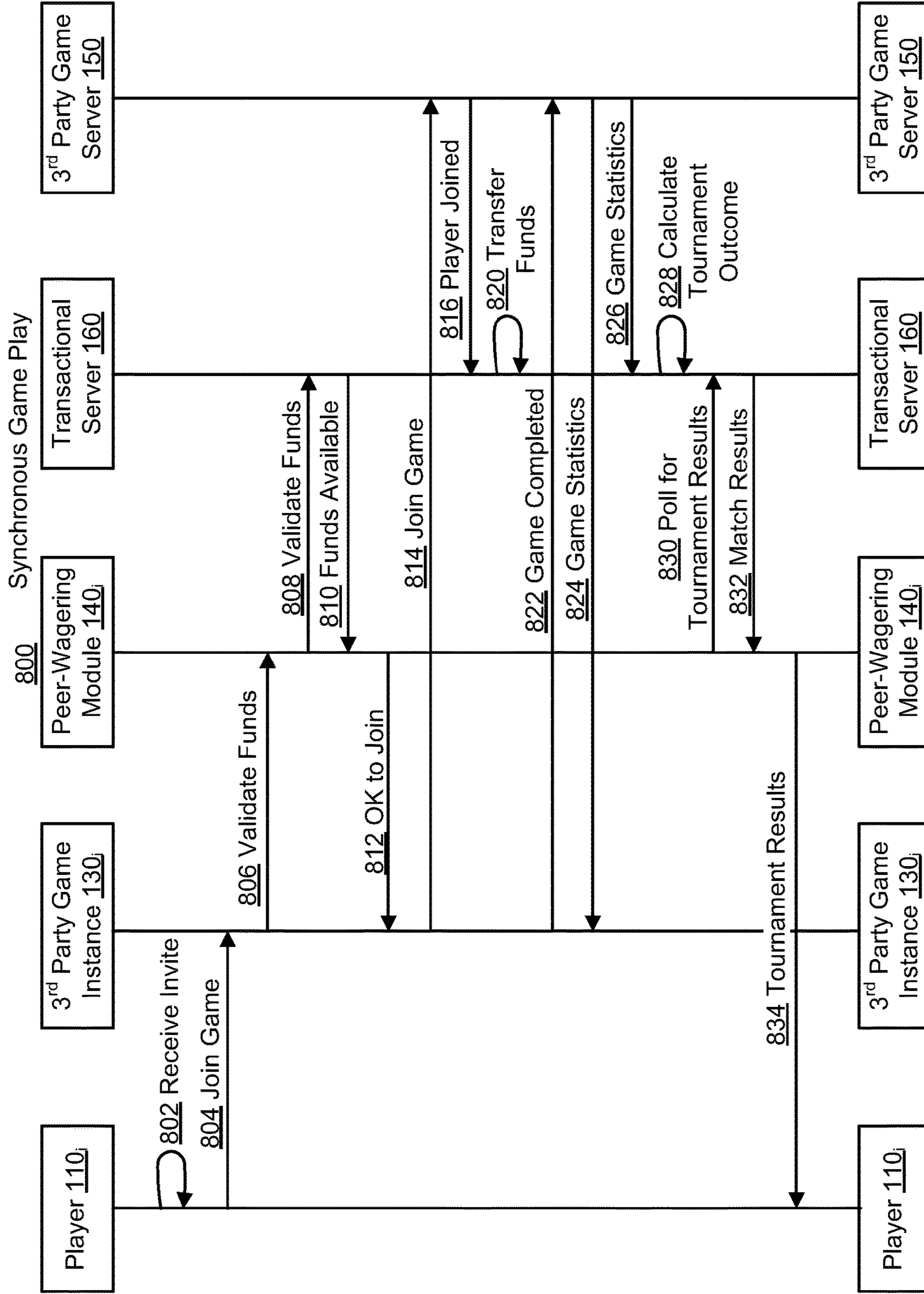


FIG. 8

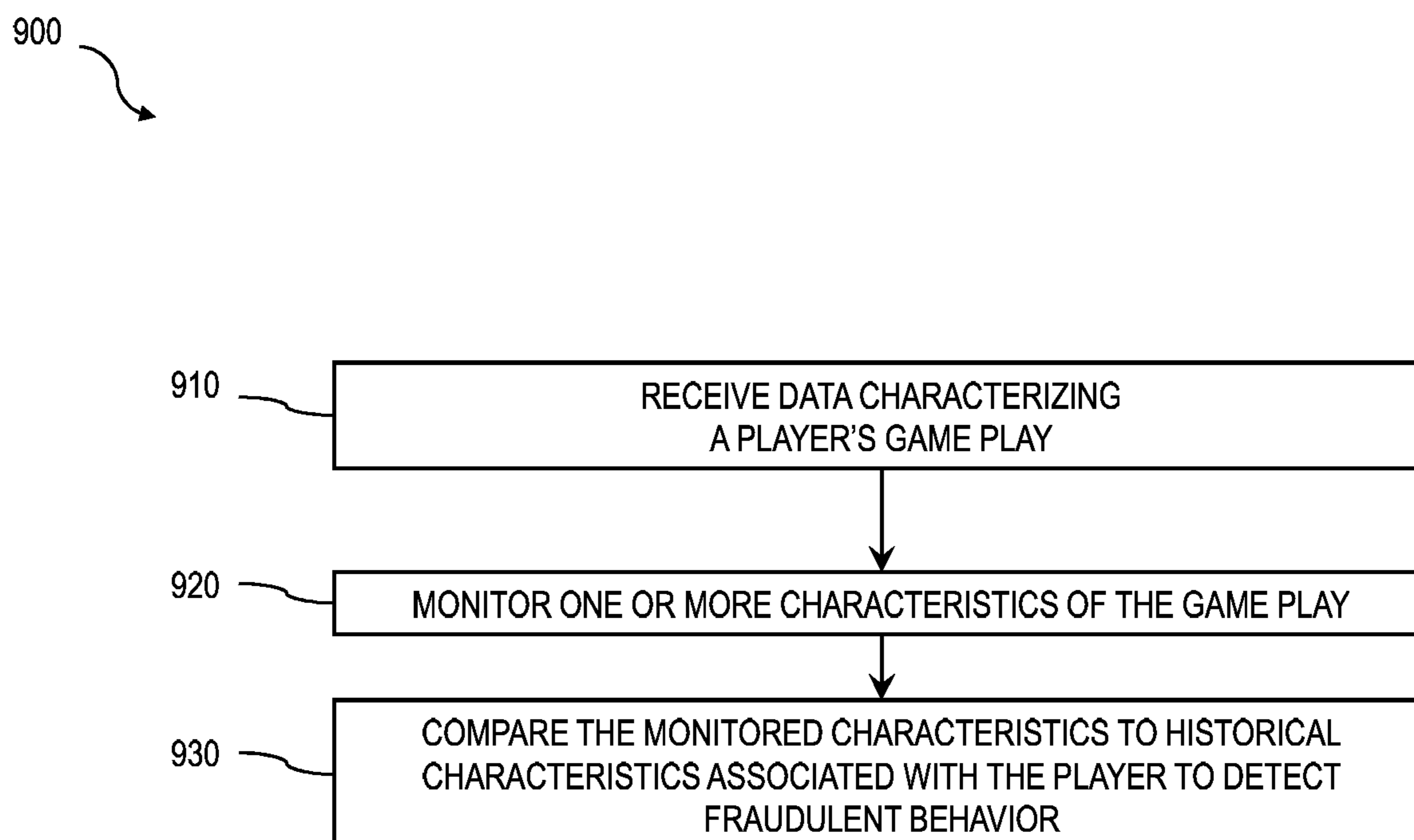


FIG. 9

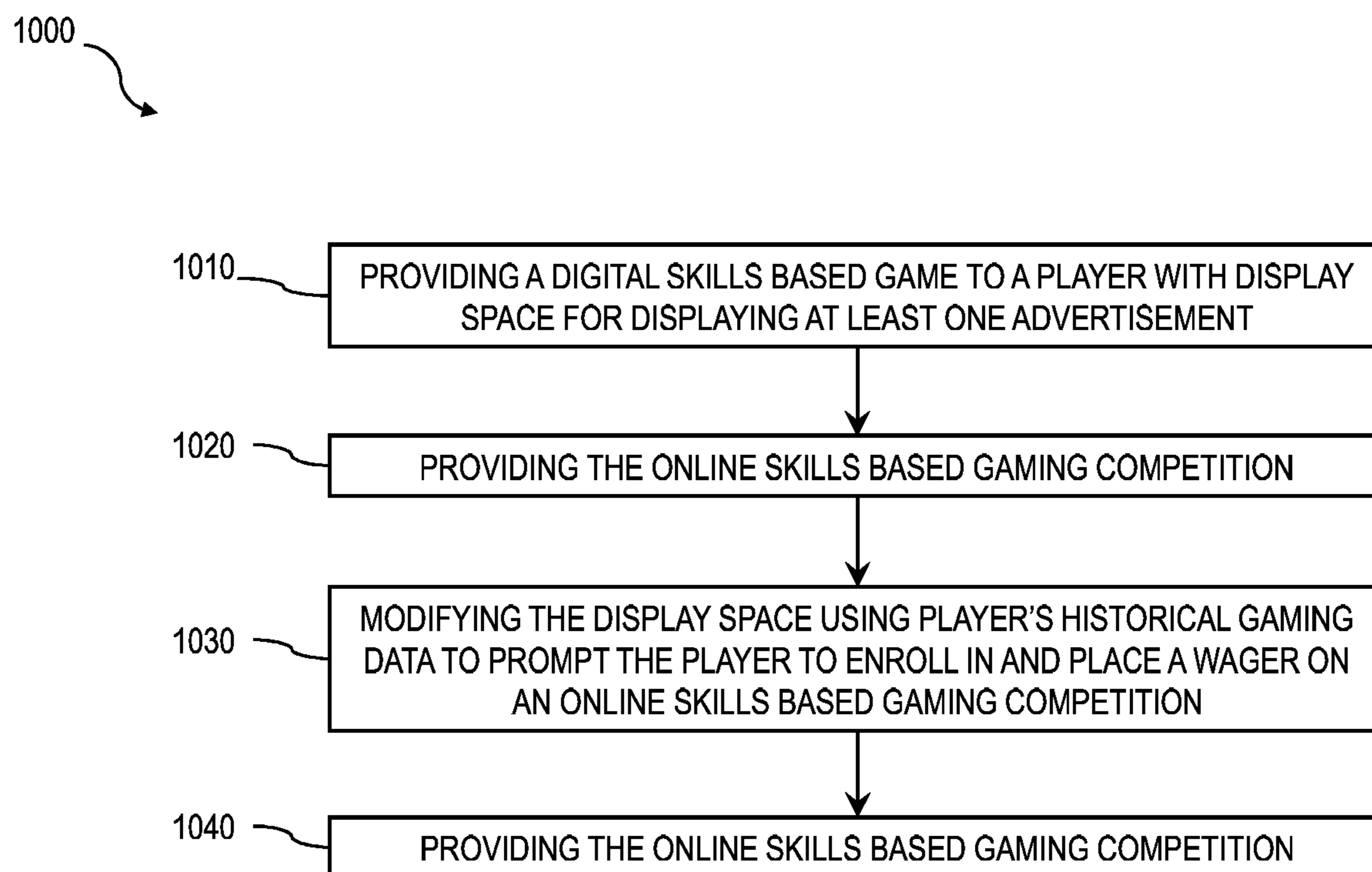


FIG. 10

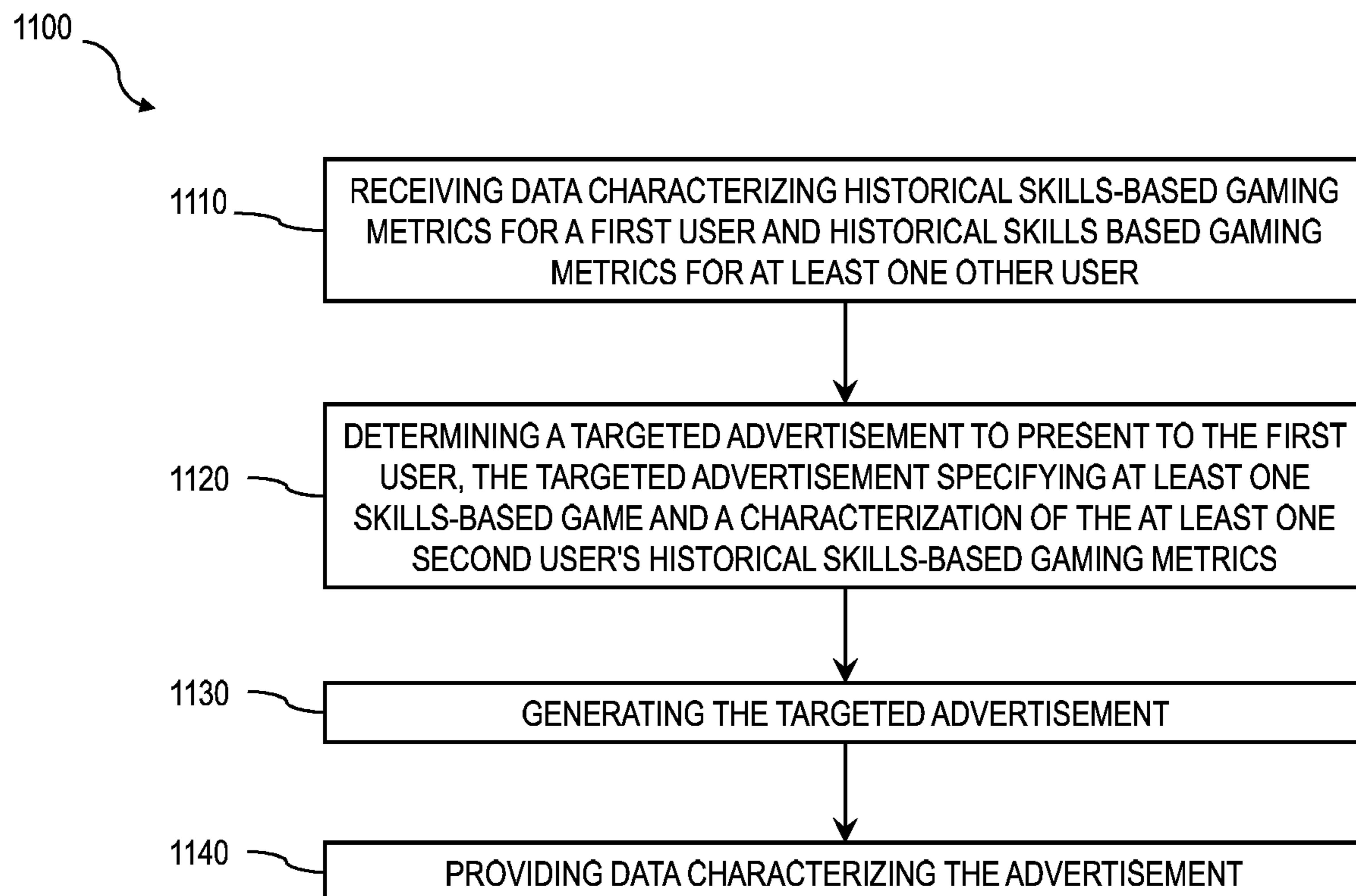


FIG. 11

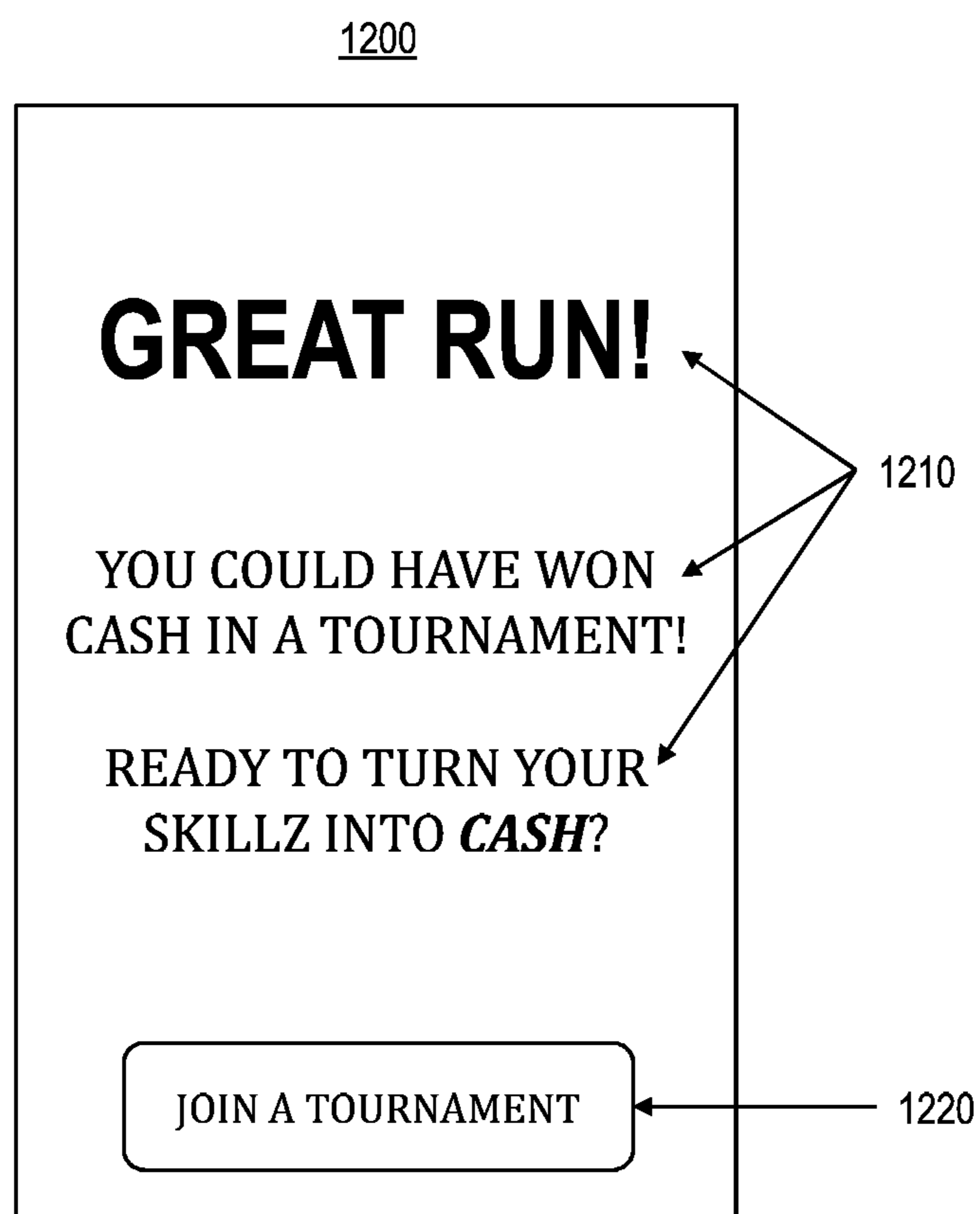


FIG. 12

PEER-TO-PEER WAGERING PLATFORM

RELATED APPLICATIONS

[0001] This application is a continuation of U.S. application Ser. No. 18/058,424, filed Nov. 23, 2022, which is a continuation of U.S. application Ser. No. 16/992,618, filed Aug. 13, 2020, which is a continuation of U.S. application Ser. No. 16/382,890, filed Apr. 12, 2019, and granted as U.S. Pat. No. 10,748,373, which is a continuation of U.S. application Ser. No. 15/595,794, filed May 15, 2017, and granted as U.S. Pat. No. 10,290,179, which is a continuation of U.S. application Ser. No. 14/976,569, filed Dec. 21, 2015, and granted as U.S. Pat. No. 9,649,564, which is a continuation of U.S. application Ser. No. 14/212,651, filed Mar. 14, 2014, and granted as U.S. Pat. No. 9,240,101, which is a continuation of U.S. application Ser. No. 13/858,581, filed Apr. 8, 2013, and granted as U.S. Pat. No. 8,715,077, which is a continuation-in-part of U.S. application Ser. No. 13/569,424, filed on Aug. 8, 2012, and granted as U.S. Pat. No. 8,414,387, the contents of each is hereby incorporated by reference in their entirety.

TECHNICAL FIELD

[0002] The subject matter described herein relates to a peer-to-peer software platform that enables dynamic advertising based on gameplay and wagering on online skill based games.

BACKGROUND

[0003] An online game is a video game played over a network on some form of computer, mobile device, or on a video game console such as the Xbox 360 and PlayStation 3. This usually means the Internet or equivalent technology, but games can use a wide range of technologies. A recent expansion of online gaming has reflected the overall expansion of computer networks from small local networks to the Internet and the growth of Internet access itself.

[0004] Online games can range from simple text based games to mobile games to games incorporating complex graphics and virtual worlds populated by many players simultaneously. Many online games require skill and strategy and have a social aspect beyond single player games. For example, players compete head-to-head, in a tournament, or for the highest score on a leader board. Example categories of online skill-based games include first person shooters, real time strategy games, social games, role-playing games, board games, card games, etc. Due to their social nature, many online games are inherently competitive.

SUMMARY

[0005] In one aspect, data characterizing historical skills-based gaming metrics for a first user and historical skills-based gaming metrics for at least one second user is accessed. Using the accessed data and a set of rules, a targeted advertisement to present to the first user is determined. The targeted advertisement specifies at least one skills-based game and a characterization of the at least one second user's historical skills-based gaming metrics. The targeted advertisement is generated. Data characterizing the targeted advertisement is provided.

[0006] In another aspect, a digital skills based game is provided to a first user. The skills based game includes display space for displaying at least one advertisement to the

user. Using historical skills-based gaming metrics and a set of rules, a targeted advertisement to present to the first user is determined. The targeted advertisement specifies at least one skills-based game and a characterization of at least one second user's historical skills-based gaming metrics. The advertisement display space is modified to include the targeted advertisement. The advertisement prompts the user to, using a peer-to-peer wagering platform, enroll in and place a wager on an online skills based gaming competition. The online skills-based gaming competition is provided to the user.

[0007] In yet another aspect, a system includes a third party gaming server, a plurality of third party game instances, and a transactional server. The third party gaming server provides data related to the third party game. The plurality of third party game instances are in communication with and receiving game data from the third party gaming server. Each third party game instance includes a peer-wagering module. The transactional server determines, using historical skills-based gaming metrics and a set of rules, a targeted advertisement to present to a first user. The targeted advertisement specifies at least one skills-based game, and a characterization of at least one second user's historical skills-based gaming metrics. The transactional server generates the targeted advertisement.

[0008] One or more of the following features can be included. For example, the advertisement can further specify a proposed wager for the first user in connection with the at least one skills-based game. When to prompt the targeted advertisement to the first user can be determined using the accessed data and the set of rules. Historical skills-based gaming metrics can include game-related attributes from a user's past game plays. Historical skills-based gaming metrics can include game-related attributes from a user's in-progress game or most recent game played. Historical skills-based gaming metrics can include one or more of: game outcome, game score, game level completed, game difficulty, player ratings, average game scores, statistical measure of game outcome, game achievements, top scores across a set of other user's results, completion of in-game objectives, user ranking, in-game character level, in-game character possession of game item, in-game character attribute.

[0009] Historical skills-based gaming metrics can be aggregated by tracking a user's game data over a period of time. The first user can provide a peer-to-peer wagering platform that enables the first user to place a wager on an online skills-based gaming competition. Providing can include transmitting, persisting, displaying, and prompting to the user. The set of rules can be customizable by a skills-based game tournament creator. The targeted advertisement can further specify at least one skills-based game tournament. The historical skills-based gaming metrics for the at least one second user can be taken from skills-based game tournaments. The advertisement can further specify a proposed wager for the first user in connection with the at least one skills-based game. When to modify the advertisement display space can be determined using the accessed data and the set of rules.

[0010] The transactional server can be configured to generate a pseudo-random number stream for use by at least some of the plurality of third party game instances.

[0011] In yet another aspect, a system includes a third party game server, a plurality of third party game instances

and a transactional server. The third party game server provides data related to a third party game. Each of the third party game instances is in communication with and receiving game data from the third party gaming server. Additionally, each third party game instances includes a peer-wagering module to receive a wager amount from a player associated with the corresponding third party game instance. Each third party game instance is associated with an online gaming competition and each player is participating in the online gaming competition. The transactional server receives the wager amount from each peer-wagering module. The transactional server secures previously deposited funds associated with each of the associated corresponding players. The fund amount is equal to the respective wager amount. Each server includes at least one computing system having at least one data processor.

[0012] In yet another aspect, data characterizing a plurality of players and a wager amount for each player can be received. Each player is enrolled in a third party skills based digital gaming competition. An amount of funds previously deposited by the respective player is secured for each player, using an online skills based digital game wagering platform. The secured funds are equal to the wager amount. The funds can be secured such that they cannot be transferred, withdrawn, or secured for a different wager. Data characterizing a confirmation that each of the plurality of players funds were successfully secured is transmitted.

[0013] In yet another aspect, a digital skills based game is provided to a player using a mobile processing device. The skills based game is single player and displays at least one advertisement to the player. The advertisement is modified using historical gaming results of the player. Content of the advertisement includes text, graphics, and linked-destination. The advertisement prompts the player to enroll in and place a wager on an online skills based gaming competition. The online skills based gaming competition is provided to the player.

[0014] One or more of the following features can be included. The transactional server can receive game statistics describing the results of the online gaming competition from the third party gaming server and automatically transfer, based on the received game statistics, at least a portion of the secured funds to at least one account associated with at least one of the players. The third party game can be an asynchronous competitive skill based game played between each player. The third party game can be a synchronous competitive skill based game played between each player. The transactional server can send a notification of each player's winnings, losses, score, time, and other game play information to the peer-wagering module in each of the plurality of game instances. The transactional server can determine if the previously deposited funds associated with a player are less than the wager amount. The online gaming competition can be a tournament. The online game competition can be selected from a group consisting of: Massively Multiplayer Online Role Playing Games ("MMORPG"), First Person Shooters ("FPS"), and Real Time Strategy ("RTS") games.

[0015] Data characterizing the outcome of the third party skills based gaming competition can be received. At least a portion of the secured funds can be transferred, using the online skills based digital game wagering platform, to an account associated with one of the plurality of players. Data characterizing the transfer can be transmitted. Data charac-

terizing the winnings and losses of one or more of the plurality of players can be transmitted. At least one of the plurality of player's can be rewarded with non-monetary compensation based on the outcome of the third party skills based gaming competition. The non-monetary compensation can be used as a substitute to funds to enter a future online skills based gaming competition. The data characterizing the transfer can be transmitted to a plurality of peer-wagering modules, each module included in an instance of a third party game. The data characterizing the confirmation is transmitted to a plurality of peer-wagering modules, each module included in an instance of a third party game. The data characterizing the outcome of the third party skills based gaming competition can be received from a third party game server, the game server providing game data to a plurality of game instances in communication with the game server, each game instance including a peer-wagering module.

[0016] The data characterizing the outcome of the third party skills based gaming competition can be received from a plurality of peer-wagering modules. Each module can be included in an instance of a third party game in communication with a game server providing game data to the plurality of game instances. The third party skills based gaming competition can be associated with an asynchronous game. The third party skills based gaming competition can be associated with a synchronous game. The wager amount can include a plurality of values. Each value can be associated with one of the plurality of players.

[0017] Data characterizing the player's game play can be received. One or more characteristics of the received data can be monitored. The monitored characteristics can be compared to historical characteristics associated with the player to detect fraudulent behavior by the player. Deviations between the monitored characteristics and historical characteristics can indicate fraud. The monitoring can be performed in real time. The monitoring can be performed after the online skills based game competition is complete.

[0018] Articles of manufacture are also described that comprise computer executable instructions permanently stored (e.g., non-transitorily stored, etc.) on computer readable media, which, when executed by a computer, causes the computer to perform operations herein. Similarly, computer systems are also described that may include a processor and a memory coupled to the processor. The memory may temporarily or permanently store one or more programs that cause the 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.

[0019] The subject matter described herein provides many advantages. The current subject matter provides a fully integrated set of technologies to facilitate wagering within electronic multiplayer or single-player games of skill. Users can complete a wager conveniently and receive any winnings. Users can track individual winnings and losses and, when placing bets, can be comfortable knowing that the current subject matter places wagered funds in escrow to guarantee payment to the winner(s). Additionally, users can easily create competitions using their favorite online game for friends to join and wager real money on the outcome. Users can also compete against strangers. Furthermore, a software development kit provides game developers with a toolset that is easy to integrate into games. The current

subject matter can convert any game of skill into a tournament enabled wagering platform. The current subject matter also enables user-created tournaments.

[0020] 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

[0021] FIG. 1 is a system diagram illustrating a system that facilitates wagering within electronic multiplayer games of skill;

[0022] FIG. 2 is a process flow diagram illustrating a method of facilitating wagering within electronic multiplayer games of skill;

[0023] FIG. 3 is an example user interface integrated into a third party game that can present to the players for signing into or registering their account;

[0024] FIG. 4 is an example user interface integrated into a third party game that presents to players for registering their accounts;

[0025] FIG. 5 is an example user interface integrated into a third party game that can present to players browsing public tournaments;

[0026] FIG. 6 is an example user interface integrated into a third party game that presents to the players and shows game statistics and results;

[0027] FIG. 7 is a data flow diagram illustrating data communication flow for an asynchronous turn based game;

[0028] FIG. 8 is a data flow diagram illustrating data communication flow when a player joins a game tournament;

[0029] FIG. 9 is a process flow diagram illustrating a method of detecting fraudulent behavior;

[0030] FIG. 10 is a process flow diagram illustrating a method of advertising an online skills based competition;

[0031] FIG. 11 is a process flow diagram illustrating a method of providing data characterizing a targeted; and

[0032] FIG. 12 is an example advertisement.

[0033] Like reference symbols in the various drawings indicate like elements.

DETAILED DESCRIPTION

[0034] FIG. 1 is a system diagram 100 illustrating a system that facilitates wagering within electronic multiplayer games of skill. A plurality of players 110_i ($i=1, 2, \dots, N$) operate respective player clients 120_i . Each player client 120_i includes a third party game instance 130_i . The game instance 130_i is any online digital game (e.g., video game) in which player skill determines game outcome, not chance, and wherein multiple players 110_i can compete against one another online. Games are consistent across game instances 130_i (e.g., if the players 110_i are playing chess, each game instance 130_i is an instance of an electronic chess game). Each game instance 130_i can be in communication with and receiving game data from a third party game server 150. The game server 150 provides game data necessary to operate the game. Alternatively, game instances 130_i can exchange game data directly.

[0035] Each game instance 130_i includes a peer-wagering module 140_i . The peer-wagering module 140_i integrates into

the game instance 130_i and enables the players 110_i to wager on the outcome of a given game competition. The peer-wagering module 140_i communicates with and works in tandem with a transactional server 160. The transactional server 160 maintains account information for each player 110_i , including financial information, and acts as a trusted party to hold funds in escrow and/or secure funds to enforce the terms of a wager (i.e., ensures winning players receive the winnings).

[0036] The transactional server 160 can also pass data characterizing advertisements (e.g., advertising logic, invitations, and/or messages) to the third party game server 150. This advertising data can be algorithmically customized based on player 110_i historical gaming data or metrics of the gaming data. For example, an advertisement could include or be modified to prompt the player with “you’ve won four out of five of your last games, click here to play in \$5 tournament and take your game play to the next level.”

[0037] FIG. 11 is a process flow diagram 1100 illustrating a method of advertising to (e.g., inviting) a player (e.g., user) to enroll and participate in an online skills based gaming tournament. Data is received at 1110 characterizing historical skills-based gaming metrics for a first user and historical skills based gaming metrics for at least one other user. The player can be currently playing or have recently played the game and the player can be playing the game while not being enrolled in an online skills-based gaming competition. A targeted advertisement to present to the first user is determined, using a set of rules, at 1120. The targeted advertisement specifies at least one skills-based game and a characterization of other user’s historical skills-based gaming metrics. The targeted advertisement can entice the player to take a specific action based upon the player’s gameplay (e.g., the player is prompted/invited to join a competition). The targeted advertisement can be customized to an individual player. The targeted advertisement is generated at 1130 and the advertisement is provided at 1140. Providing can include transmitting, persisting, displaying, and prompting to the player.

[0038] The historical skills-based gaming metrics can be supplied by the player client 120_i , or tracked and/or recorded (e.g., over time) by the transactional server 160, or the third party gaming server 150. Historical skills-based gaming metrics can include game attributes that can be traced (e.g., recorded or monitored) and are associated with the player’s in-progress, most recent, and/or past game plays. For example, historical skills-based gaming data can include: an in-game score, game outcome, time to complete a stage, game level achieved, difficulty, number of enemies, power-ups acquired, player ratings, number of wins or losses, statistical measures of attributes (e.g., average scores, ratio of wins to losses, etc.), top scores across some or all players, completion of any game related task, objective, and/or achievement, in-game acquisition of items, and/or other character attributes. Historical skills-based gaming metrics can include other traceable game attributes.

[0039] When the advertisement is presented to the player and/or the content of the targeted advertisement can be based on configurable messaging rules (i.e., a set of rules). The rules can be used to target players with specific attributes, levels of skill, or relationships. For example, an advertisement or invitation could prompt to a player to join and wager on a tournament based on a player exceeding a certain predetermined value or score for a particular in-game chal-

lenge. The rules could include whether the player has completed a particular level, game, number of games played, time played, or other game element. An advertisement could be presented when a player is among a certain percentile of players (e.g., prompting players to enroll in a tournament with like-skilled players).

[0040] Additional rule examples can include when a player has wagered, lost, and/or won at least a certain amount (e.g., prompting players to enroll in a tournament that requires “high stakes” or large minimum wagers), when one or more other players associated with the first player (e.g., links on a social networking website) are enrolled in a tournament (e.g., prompting a player to enroll in a tournament with their friends), when a player has played a certain number of games, and/or when a player has completed a level or challenge in a tournament. The messaging rules can also be based on a geographic location of the player (e.g., online wagering regulations may vary based on locality), or whether the player is eligible to wager cash or rewards on a tournament. Administrators, and/or tournament creators can configure or customize the messaging rules.

[0041] The targeted advertisement can include graphical, textual, and hyperlink information necessary to populate pre-existing advertising space with a customized invitation for players to wager on a skill based gaming tournament. Targeted advertisements can include customized messages derived or based on the messaging rules and/or other data triggered by a rule, gameplay, or historical skills-based gameplay metrics. The targeted advertisement can be targeted to the player and include a characterization of the player’s historical skills-based gaming metrics and a characterization of one or more other player’s historical skills-based gaming metrics. For example, the targeted advertisement can include information regarding how many active games are being played for a given game or a different game. The advertisement can include a message regarding how many of the player’s friends, contacts, or in-network users are active in the system. The advertisement content can show how a player would fare if they joined a tournament or played a different game by including a comparison between or characterization of the historical gaming data and other players in the system (e.g., showing how much money the player would have won in a real tournament, a comparison of the player’s average score to other player’s average scores, a mock leader board where the game player is compared to all players of the game, etc.). The targeted advertisement could include a proposed wager amount for the first player.

[0042] FIG. 12 is an example advertisement 1200 presented to a player after the player satisfied a message rule (e.g., had a high score). The advertisement includes text 1210, including a characterization of other player’s historical skills-based gaming metrics, and a button 1220 (i.e., link) to redirect the player to enable the player to enroll and wager on a tournament. The advertisement could further include graphics and be embedded in the game user interface.

[0043] FIG. 10 is a process flow diagram 1000 illustrating a method of advertising an online skills based competition. At 1010, a digital skills based game is provided to a player (i.e., user). The game contains display space for displaying advertisements. At 1020, a targeted advertisement to present to the first user is determined using historical skills-based gaming metrics and a set of rules. The targeted advertise-

ment specifies at least one skills-based game, and a characterization of at least one second user’s historical skills-based gaming metrics. The advertisement display space is modified at 1030 using the player’s historical gaming data to prompt the player to use a peer-to-peer wagering platform to enroll in and place a wager on an online skills based gaming competition. At 1040, the online skills based gaming competition is provided to the player.

[0044] By utilizing specific player’s historical gaming data and messaging rules, the advertisements/messages can be customized to the player and each advertisement has a higher likelihood of enticing the player than a non-customized (e.g., a general) advertisement.

[0045] Referring again to FIG. 1, players 110_i can register and fund accounts either directly with the transactional server 160 (for example, accessing the server via a web browser) or through the peer-wagering module 140_i. The players 110_i can fund their account through several means, such as, for example: PayPal, credit cards, Automated Clearing House (ACH) transfer, cash, loyalty cards, pre-paid cards, coupons, system credit, credit earned by watching advertisements, Google checkout, etc. Other methods of funding the account are possible.

[0046] Players 110_i can create tournaments. Prior to game play, the peer-wagering module 140_i can receive credential information from the player 110_i, send the credentials to the transactional server 160, which can authenticate the credentials. Authentication can include age and location controls to ensure local law compliance. Age can be entered by the player 110_i and location can be verified by any of a billing address used to fund an account, the GPS location of a mobile device (if available), and an IP address of the player client 120_i. When each player 110_i enrolls in a tournament, including placing a wager on the tournament (e.g., paying a tournament entry fee), the transactional server 160 can secure the player 110_i funds. Secured funds cannot be withdrawn or used for another wager. Securing the funds can include transferring the funds from the player account to an escrow account as well as placing a “hold” on the funds in the player’s account.

[0047] Once the transactional server 160 secures funds from all participating players 110_i, the tournament can proceed. The tournament proceeds under normal game mechanics (such as each game instance 130_i communicating game data with the game server 150) until game play completes. The transactional server 160 receives completed game statistics from the game server 150 or, alternatively, from each peer-wagering module 140_i. The game statistics can indicate winners and losers based on one or more in-game metrics. A player 110_i can also determine one or more custom in-game metrics to be used in determining winners and losers during tournament initialization. The transactional server 160 transfers the previously secured funds to one or more player 110_i accounts based on the game statistics. For example, a winning player can have the player’s winnings transferred from the other players’ accounts or the secure escrow account into the winning player’s account. The transactional server 160 sends financial data related to winnings and losses to each peer-wagering module 140_i, which provides the financial data to the players 110_i. Additionally, the transactional server 160 can send the game statistics to each peer-wagering module 140_i, which provides the game statistics to the player 110_i.

[0048] Game statistics sent to the transactional server **160** from either the game server **150** or from each peer-wagering module **140_i**, can include summary level statistics such as winners and losers and/or specific in game actions such as player orientation within the gaming environment, player actions (e.g. buttons pressed or character movement), or user display details. The user display details can include, but are not limited to, graphics card-information, in-game screen shots and live action game-play. These statistics can be used to determine system level player-rewards that are independent from tournament outcomes as well as in the detection of fraudulent behavior through any of the following: real-time tournament monitoring, delayed tournament review, or statistical player review for idiosyncratic behavior or behavior characteristic of fraudulent play.

[0049] The peer-wagering module **140_i**, provides necessary user interface components and player **110_i**, to transactional server **160** interaction functionality for the game instance **130_i**. This provides a low barrier for third party game providers to enable the peer-wagering functionality into the game. The peer-wagering module can be implemented with platform specific software development kits (SDKs).

[0050] Communication can occur over any suitable communications network, such as, for example, the internet.

[0051] FIG. 2 is a process flow diagram **200** illustrating a method of facilitating wagering within electronic multi-player games of skill. At **210**, the transactional server **160** receives data characterizing each player and each player's wager amount. The wager amount can be variable among the players or fixed among the players. Optionally the players may be matched in tournaments with players wagering similar amounts. Each player has enrolled in a third party skills based gaming competition (e.g., a tournament). At **220**, the transactional server **160** secures, for each player and using a respective player account, funds equal to the corresponding received wager amount. The players have previously registered the player account and deposited funds in the account. The transactional server **160** can place the funds in an escrow account until the completion of the gaming competition. Alternatively, the transactional server **160** can secure the funds by placing a "hold" on the funds in the players' accounts preventing withdrawal or use of the funds. At **230**, the transactional server **160** transmits data characterizing a confirmation that the transactional server **160** has successfully secured funds from each enrolled player. If funds as to a player are not available in the respective player account, the transactional server **160** can send data to prompt the player to deposit additional funds into the player's account.

[0052] Optionally, at **240**, the transactional server **160** receives data characterizing the outcome of the competition. The data can include in-game statistics and indicate that some of the enrolled players are winners and some are losers. Alternatively, the transactional server **160** can differentiate, based on the in-game statistics, players that are winners and players that are losers. The transactional server **160** can determine based on the in-game statistics the amount of secured funds (if any) to which enrolled players are entitled. Optionally, at **250**, the transactional server **160** can transfer at least a portion of the secured funds to one or more accounts associated with their respective enrolled players. Optionally, at **260**, the transactional server **160** can transmit data characterizing the transfer and/or the in-game statistics. The transactional server **160** can receive the

respective data from and transmit the respective data to one or more of the peer-wagering module **140_i**, and the third party game server **150**.

[0053] The game can be asynchronous. Asynchronous games cover any turn-based game where players **110_i** take turns and real-time game play is not an issue. A player **110_i** may leave the game to perform other tasks on the same device on which the game is running, without forfeiting a tournament. Asynchronous games can include games such as Chess, Checkers, Go, and most board games where timing of player turns is not a consideration.

[0054] The game can be synchronous. Synchronous games cover any game where real time interaction between the game and player or between players is required. For example, first person simulations wherein each player has one or more characters (i.e. avatars) and multiple players' characters are interacting in real time with each other's characters or game environment in a synchronized way. First person shooters, driving and racing simulators as well as real time sports simulations are synchronous. Some turn-based game designs can also include synchronous aspects if all players must be present at some times while a game is in progress.

[0055] Whether synchronous or asynchronous, an entire level of a can game constitute a turn, and the players can take turns independently. For example, games such as Angry Birds, where two or more players can complete a level, independently but potentially at the same time, and the winner can be determined based on some metric when all players complete the level. In this manner, an entire level constitutes a player turn.

[0056] The transactional server **160** can provide an application-programming interface (API) for the third party game instances **130_i**, or the third party game server **150** to communicate with the transactional server **160**.

[0057] Establishing tournaments can allow players **110_i** to compete with one another within skill-based games in a single or series of contests. Tournaments can work with synchronous and asynchronous play modes, and tournaments can be user or system generated. Tournaments can be either public or private. Public tournaments can be open to any registered player while private tournaments can be open only to invited players. Any tournament must necessarily have at least two participants. Player-created tournaments can require a specific number of entrants in order to begin the competition, whereas system-created tournaments can have a fixed or variable number of permitted and/or required entrants. Tournaments with a variable number of participants can have a fixed starting time and can have a maximum number of allowed entrants.

[0058] Tournaments can comprise a single match or a series of matches (i.e., multi-round tournaments). The structure can be determined at the time of tournament creation. Each match can have a specified number of participants and winners. A specific win-metric can determine the winners of each match. Individual tournament rounds (e.g., matches) can begin at a pre-determined time set up by the creator of the tournament or they can proceed in immediate sequence. Matches not completed by the next designated match time slot can be terminated, and the top contenders from each non-concluded match can be rewarded the win for that match.

[0059] Each tournament or tournament round can have defined criteria by which the winner(s) are determined.

Possible win-metrics can be dependent on the type of game, but can include (for example): highest score, first to complete a level, least moves to complete a level, etc. Additionally, each tournament can have an entry fee which is a dollar amount required to enter the tournament (i.e., a wager amount). However, some system-created tournaments can waive this fee for some or all players **110_i**. The tournament creator can determine the amount of the entry fee.

[0060] Prizes offered to tournament winner(s) can be determined at the time of tournament creation. Prize information can be visible to all prospective tournament entrants. In the case of user-created tournaments, the player creating the tournament can set the total prize pool automatically. For example, the creating player can set the prize pool based on the number of entrants and the entry fee that the creating player has specified. The creating player can allocate prizes in a variety of ways such as awarding prizes to more than one participant in a given tournament. A tournament creator can specify the number of winners and the percentage allocation of prizes to each of those winners. A public tournament creator can base prize distribution on individual performance relative to the defined win-metric; however, a private tournament creator can incorporate team scoring relative to the win-metric in determining prize allocation.

[0061] An example player **110_i** interaction with an asynchronous multiplayer game including the current subject matter includes creating a tournament, joining a tournament, and concluding a tournament. To create a tournament, the player **110_i**, using the player client **120_i**, launches a third party game instance **130_i**. The player can choose within the game to compete using the peer-wagering module **140_i**. The user can log into their transactional server **160** account, optionally electing to remain logged in to the account within this session and future session of this game.

[0062] FIG. 3 is an example user interface **300** integrated into a third party game that can present to the player for signing into or registering their account. At **310**, the player **110_i** can enter credential information such as an email address and password. At **320**, the player **110_i** can sign in using the play button **320**.

[0063] Once signed into their account, the player **110_i** can create a public or private tournament and set parameters for the tournament. The transactional server **160** can prompt the player to add funds to their account if the player **110_i** has inadequate funding. The player **110_i** can invite several known players using their account names (e.g., email addresses, user names, etc.). If the tournament is a public tournament, uninvited participants can join. Public tournaments can start on a rolling basis with players taking their first turn as soon as they join the tournament or as soon as the player before them has played (depending on game mechanics). Private tournaments start when the prescribed number of participants has entered the tournament. The game proceeds according to the game developer's prescribed game mechanics.

[0064] If the player **110_i** does not yet have an account, the player **110_i** can register for an account by entering information such as email address and password in data fields **330** and pressing the next button **340**. FIG. 4 is an example user interface **400** integrated into a third party game that presents to the player **110_i** for registering their account. The player **110_i** can enter at **410** credit card information and billing address and specify at **420** an amount to charge the credit card. The player **110_i** can select checkboxes **430** to agree to

the terms of service and confirm payment. The player **110_i** can complete the registration using push button **440**. The player **110_i** can enter additional account information (not shown) such as full name, age, phone/mobile number (for SMS notification of game events), and an account name that will be visible to other players **110_i**.

[0065] To join a tournament, a player **110_i** can receive a notification inviting them to a tournament. Notifications can arrive via any one of several means. For example, push notifications, SMS, email and in-game notifications are all options for notification. Alternatively, the user browses public tournaments that are seeking players and selects one. The player **110_i** launches the game via the notification they have received or proceeds into the game after selecting a public tournament. The player **110_i** accepts the terms of the tournament, including the funding requirements of the tournament in response to a prompt. If the player **110_i** has inadequate funding in their account, they can add funds. The player **110_i** can accept the terms of the tournament and enter the game. The game proceeds according to the game developer's prescribed game mechanics.

[0066] In the case when the multiplayer game is synchronous, all players **110_i** must start the game simultaneously. To provide this functionality, the transactional server **160** presents the players **110_i** with a tournament lobby while they wait for the synchronous game tournament to begin. The transactional server **160** provides the lobby for both the player **110_i** who created the tournament as well as the players **110_i** who join the tournament. The players can see the other players as they join the tournament and can have the opportunity to withdraw from the tournament before the tournament begins. The game can begin a predetermine length of time (e.g., 60 seconds) after all players have joined the tournament. Any withdrawal by the players **110_i** after this point constitutes forfeiture of the tournament entry fee (i.e. the wager).

[0067] The tournament can also start at a predetermined time independent of the number of players **110_i** in the lobby. Players **110_i** can sign-up for the tournament in advance of the tournament start time. Once players **110_i** have signed-up for a tournament, the transactional server **160** can alert the players **110_i** that the tournament is starting soon through a variety of methods such as email, SMS, in-game alert, etc.

[0068] FIG. 5 is an example user interface **500** integrated into a third party game that can present to a player **110_i** browsing public tournaments that are seeking players. Pending tournament list **510** presents several public tournaments and details regarding those tournaments. For example, the first tournament on the list **520** shows a summary of the tournament settings. The tournament **520** name is "James Game," the type of tournament is "highest score," there are currently ten entrants with a maximum of twelve as shown under number of players. For this example tournament, the wager or entry amount is one dollar and the total prize or payout is ten dollars. The time left shows when the example tournament **520** will start.

[0069] Detailed information **530** provides tournament information for a selected tournament and shows the name, type, number of players and maximum players, entry fee, and prize breakdown. In this example tournament, the top three players **110_i** will receive winnings. Player list **540** provides each player name and overall score currently entered in the tournament. Push buttons **550** enables the player **110_i** to enter or not enter the tournament.

[0070] At the conclusion of a tournament, the normal game mechanics are complete and the third party game server 150 posts data to the transactional server 160 indicating game results. Each player 110_i who completes the game should see game results immediately, including data from the transactional server 160 detailing their winnings or losses for the tournament. Other players 110_i in the tournament can also receive a notification detailing their winnings or losses from the transactional server 160 and indicating that their tournament is complete. For games where a continuous connection to the service is important to game-play or game integrity, any player 110_i who drops off the service and does not re-connect within a certain predetermined period forfeits the game. Third party game developers can determine the period or can optionally include reconnection logic to re-establish a lost connection between players when such a loss of connection cannot be used to gain an advantage in either game-play or wagering. Additionally, players may be required to take their turn or take another specified in-game action within a predetermined amount of time. Players who do not act within this set amount of time, as determined by third party game developers, will forfeit the game. The peer-wagering module 140_i can report to the transactional server 160 when the player 110_i disconnects from the competition or when the player 110_i has not taken his turn within the allotted time.

[0071] FIG. 6 is an example user interface 600 integrated into a third party game that presents to the player 110_i and shows the game statistics and results (e.g., a leader board). Column 610 lists each tournament participants account name. Column 620 displays each participating player's in-game rank and column 630 displays each participating player's in game score. Column 640 displays any prizes won by the corresponding player. In this example result, the top three players received funds. At 650 the player 110_i can launch the game again and re-enter a new tournament with the same settings (e.g., same wager amount, win metric, invite list, etc.). At 660, the player 110_i can go to a tournament selection screen.

[0072] FIG. 7 is a data flow diagram 700 illustrating data communication flows between an example implementation of the elements of FIG. 1 for an asynchronous turn based game. Elements of FIG. 1 are shown across the top and bottom of FIG. 7, specifically, player 110_i (operating player client 120_i), third party game instance 130_i, peer-wagering module 140_i, transactional server 160, and third party gaming server 150.

[0073] At 702 through 708 the player 110_i logs into the transactional server 160 using the interface provided by the peer-wagering module 140_i embedded in the game instance 130_i. The peer-wagering module 140_i authenticates 706 with the transaction server 160 allowing for further messaging between the peer-wagering module 140_i and the transaction server 160, via authentication tokens, security certificates, or other user/password exchanges 708. All future requests are set with the valid authentication method. Peer-wagering module 140_i requests 710 a list of potential gaming opponents for the logged in player for the currently running game. The transactional server 160 and peer-wagering module 140_i returns (712 and 714) a list of the players available for play as well as a list of current tournaments requiring additional players. The creating player 110_i creates a tournament 716 and invites other players 110_i. The peer-wagering module 140_i sends 718 the transactional server 160 data

regarding invited players 110_i and the tournament name. When a player 110_i joins the tournament, the player's information is communicated to the peer-wagering module 140_i as well. Funds are transferred 720 from the joining/creating player's transactional server 160 account. The transactional server 160 places these funds in escrow or places a "hold" on wagered funds in player accounts. The transactional server 160 returns 722 a unique tournament ID and all entered players 110_i wait 724 for the tournament to start. Each peer-wagering module 140_i communicates with 726 the transactional server 160 with the tournament ID until the server indicates 728 the tournament has started. The peer-wagering module 140_i can signal the transactional server 160 to force the tournament to start for those players who have joined.

[0074] During normal game play, all communication (732 and 734) occurs between the third party game instance 130_i and its own third party game server 150. All active gaming statistics including rank of each player is communicated and coordinated between the game instance 130_i and the gaming server 150. The third party game instance 130_i communicates 736 data regarding any players 110_i who have left the game or any data relevant to the detection of fraudulent behavior.

[0075] On completion of the game, each connected peer-wagering modules 140_i send (740 and 742) tournament statistics to the transactional server 160. The transactional server 160 calculates winnings and losses based on the tournament statistics. The transactional server 160 sends (744 and 746) notifications to all players in the completed tournament indicating their tournament has completed, and they have winnings or losses, as well as a leader board for the tournament. The players 110_i receiving the completion notification can launch a new game via the notification.

[0076] Alternatively, on completion of the game 750, the third party game server 150 sends 752 games statistics to the third party game 130_i and sends 754 tournament statistics to the transactional server 160. Each peer-wagering module 140_i polls 756 the transactional server 160. The transactional server 160 sends (758 and 760) a leader board for the tournament, as well as notifications to all players in the completed tournament indicating their tournament has completed, and indicating whether they have winnings or losses. The players 110_i receiving the completion notification can launch a new game via the notification.

[0077] The data flow diagram of FIG. 7 corresponds to an asynchronous turn based game; however, appropriate modification can be performed for synchronous based game play.

[0078] FIG. 8 is a data flow diagram 800 illustrating the data communication flows between an example implementation of the elements of FIG. 1 for a player 110_i/joining a game tournament. At 802, player 110_i receives a notification or in-game message and joins 804 the tournaments. If the player joins an existing tournament through the lobby system, there will be a standard chat window enabled. The transactional server 160 will validate (806 and 808) the funds and inform (810 and 812) the third party game instance 130_i that it is ok for the player 110_i to join the tournament. At 814 and 816, the player joins the tournament. The transactional server 160 transfers 820 wagered funds to an escrow account or places a "hold" on wagered funds in player accounts. During normal game play, all game-related communication occurs between the third party game instance 130_i and the third party game server 150. In some

instances during game play, the third party game instance **130**, or the third party game server **150** may report **826** game statistics relevant to fraud detection to the transactional server **150**.

[0079] FIG. 9 is a process flow diagram **900** illustrating a method of detecting fraudulent behavior. At **910**, data is received characterizing a player's game play. At **920**, characteristics or attributes of the game play can be monitored and, at **930**, the monitored characteristics can be compared to historical characteristics associated with the player to detect fraudulent behavior. Deviations in the game play characteristics from the historical characteristics can indicate fraudulent behavior. Fraudulent behavior can include cheating (i.e., not playing by the game rules and thereby gaining an unfair advantage). These statistics may include specific in-game actions such as player orientation within the gaming environment, player actions (e.g. buttons pressed or character movement), or user display details including, but not limited to, graphics card-information, in-game screen shots and live action game-play. Referring again to FIG. 8, on completion of the game, the third party game server **150** posts **824** game statistics to the third party game instance **130**, and posts **826** game statistics to the transactional server **160**. The transactional server **160** calculates **828** winnings and losses based on the game statistics. Each peer-wagering module **140**, polls **830** the transactional server **160** for match results and receives **832** the match results including a list of winnings and losses, as well as a leader board for the tournament. Each peer-wagering module **140**, notifies **834** their respective player **110**, that the tournament has completed and they have either winnings or losses. The players **110**, receiving the completion notification can launch a new game via the notification.

[0080] The transactional server **160** can further be configured to generate a stream of pseudo-random numbers for use by the third party game instances. Since random numbers are typically used in gameplay engines to decide game elements and properties (e.g., what obstacles are present can be decided based on the value that a random number generator returns), the use of common random numbers can provide a common gameplay experience to a subset of users (e.g., the players involved in a third party game tournament). The common gameplay experience can be used to standardize (or level the playing field) for a game of skill that still has random elements.

[0081] For example, while Tetris is typically considered a game of skill, the order in which the Tetris pieces present to a player is normally random. By providing a common stream of pseudo-random numbers to each game instance participating in an online tournament, the order that Tetris pieces appear to each player can be common across all participating game instances. Thus, the results of a competitive tournament would be based entirely on skill, and not on a random chance of getting an easier order of Tetris pieces presented.

[0082] In one example, the transactional server **160** can use a tournament identification number as a seed to generate pseudo-random numbers, thus causing gameplay to be different between tournaments, but not between game instances involved in a given tournament.

[0083] Various implementations of the subject matter described herein may be realized in digital electronic circuitry, integrated circuitry, specially designed ASICs (application specific integrated circuits), computer hardware, firmware, software, and/or combinations thereof. These

various implementations may 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 may 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.

[0084] These computer programs (also known as programs, software, software applications or code) include machine instructions for a programmable processor, and may be implemented in a high-level procedural and/or object-oriented 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 (e.g., magnetic discs, optical disks, memory, Programmable Logic Devices (PLDs)) 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.

[0085] To provide for interaction with a user, the subject matter described herein may be implemented on a computer having a display device (e.g., a CRT (cathode ray tube) or LCD (liquid crystal display) monitor) for displaying information to the user and a keyboard and a pointing device (e.g., a mouse or a trackball) by which the user may provide input to the computer. Other kinds of devices may be used to provide for interaction with a user as well; for example, feedback provided to the user may be any form of sensory feedback (e.g., visual feedback, auditory feedback, or tactile feedback); and input from the user may be received in any form, including acoustic, speech, or tactile input.

[0086] The subject matter described herein may be implemented in a computing system that includes a back-end component (e.g., as a data server), or that includes a middleware component (e.g., an application server), or that includes a front-end component (e.g., a client computer having a graphical user interface or a Web browser through which a user may interact with an implementation of the subject matter described herein), or any combination of such back-end, middleware, or front-end components. The components of the system may be interconnected by any form or medium of digital data communication (e.g., a communication network). Examples of communication networks include a local area network ("LAN"), a wide area network ("WAN"), and the Internet.

[0087] The 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.

[0088] Although a few variations have been described in detail above, other modifications are possible. For example, the logic flow depicted in the accompanying figures and described herein do not require the particular order shown, or sequential order, to achieve desirable results. Other embodiments may be within the scope of the following claims.

1-20. (canceled)

21. A computer implemented method comprising:

determining, by an application programming interface, software widgets executing on respective mobile applications on a mobile device;

receiving, by the application programming interface executing on the mobile device, a plurality of queries, from the software widgets, for data from a database stored on the mobile device;

determining, by the application programming interface, a level of priority for each query of the plurality of queries;

adjusting, by the application programming interface, the level of priority, as an adjusted level of priority that is associated with at least one query that is stored in a data buffer

providing, by the application programming interface, each of the plurality of queries for storage in the data buffer until processing resources of the mobile device are available to process one or more of the plurality of queries; and

processing each query of the plurality of queries against the database stored on the mobile device based on the adjusted level of priority.

22. The computer implemented method of claim 21, wherein determining, by the application programming interface, a level of priority for each query of the plurality of queries comprises:

determining a level of priority for each query of the plurality of queries based on a timestamp associated with each query of the plurality of queries.

23. The computer implemented method of claim 22, wherein the timestamp associated with each query of the plurality of queries indicates a time that the query was originated by a particular software widget of the software widgets.

24. The computer implemented method of claim 21, wherein determining, by the application programming interface, a level of priority for each query of the plurality of queries comprises:

determining a level of priority for each query of the plurality of queries based on (i) a time stamp associated with the query and (ii) whether or not a particular software widget of the software widgets that originated the query is visible on a display of the mobile device.

25. The computer implemented method of claim 24, wherein determining a level of priority for each query of the plurality of queries based on (i) a time stamp associated with the query and (ii) whether or not a particular software widget of the software widgets that originated the query is visible on the display of the mobile device comprises:

determining a first level of priority for one or more first queries that originate from a software widget that is currently visible on the display of the mobile device, wherein the first level of priority prioritizes the one or more first queries for execution over one or more second queries that originate from a software widget that is not visible on the display of the mobile device.

26. The computer implemented method of claim 21, further comprising:

adjusting, by the application programming interface, a level of priority that is associated with at least one query that is stored in the data buffer.

27. The computer implemented method of claim 26, wherein adjusting, by the application programming interface, a level of priority that is associated with at least one query that is stored in the data buffer comprises:

determining, by the application programming interface, that a first software widget that was previously not visible on a display of the mobile device is now visible on the display of the mobile device;

determining, by the application programming interface, that a particular query that originated from the first software widget is stored in the data buffer; and

adjusting the level of priority associated with the particular query in a manner that promotes the particular query for execution over one or more other queries of the plurality of queries that originated from a widget that is not visible on the display of the mobile device.

28. The computer implemented method of claim 26, wherein adjusting, by the application programming interface, a level of priority that is associated with at least one query that is stored in the data buffer comprises:

determining, by the application programming interface, that a second software widget that was visible on a display of the mobile device is no longer visible on the display of the mobile device;

determining, by the application programming interface, that a particular query that originated from the second software widget is stored in the data buffer; and

adjusting the level of priority associated with the particular query in a manner that demotes the particular query for execution after each other query of the plurality of queries stored in the data buffer that originated from a software widget that is visible on the display of the mobile device.

29. A system comprising:

one or more computers and one or more storage devices storing instructions that are operable, when executed by the one or more computers, to cause the one or more computers to perform operations comprising:

determining, by an application programming interface, software widgets executing on respective mobile applications on a mobile device;

receiving, by the application programming interface executing on the mobile device, a plurality of queries, from the software widgets, for data from a database stored on the mobile device;

determining, by the application programming interface, a level of priority for each query of the plurality of queries;

adjusting, by the application programming interface, the level of priority, as an adjusted level of priority that is associated with at least one query that is stored in a data buffer

providing, by the application programming interface, each of the plurality of queries for storage in the data buffer until processing resources of the mobile device are available to process one or more of the plurality of queries; and

processing each query of the plurality of queries against the database stored on the mobile device based on the adjusted level of priority.

30. The system of claim 29, wherein determining, by the application programming interface, a level of priority for each query of the plurality of queries comprises:

determining a level of priority for each query of the plurality of queries based on a timestamp associated with each query of the plurality of queries.

31. The system of claim **30**, wherein the timestamp associated with each query of the plurality of queries indicates a time that the query was originated by a particular software widget of the software widgets.

32. The system of claim **29**, wherein determining, by the application programming interface, a level of priority for each query of the plurality of queries comprises:

determining a level of priority for each query of the plurality of queries based on (i) a time stamp associated with the query and (ii) whether or not a particular software widget of the software widgets that originated the query is visible on a display of the mobile device.

33. The system of claim **32**, wherein determining a level of priority for each query of the plurality of queries based on (i) a time stamp associated with the query and (ii) whether or not a particular software widget of the software widgets that originated the query is visible on the display of the mobile device comprises:

determining a first level of priority for one or more first queries that originate from a software widget that is currently visible on the display of the mobile device, wherein the first level of priority prioritizes the one or more first queries for execution over one or more second queries that originate from a software widget that is not visible on the display of the mobile device.

34. The system of claim **29**, the operations further comprising:

adjusting, by the application programming interface, a level of priority that is associated with at least one query that is stored in the data buffer.

35. The system of claim **34**, wherein adjusting, by the application programming interface, a level of priority that is associated with at least one query that is stored in the data buffer comprises:

determining, by the application programming interface, that a first software widget that was previously not visible on a display of the mobile device is now visible on the display of the mobile device;

determining, by the application programming interface, that a particular query that originated from the first software widget is stored in the data buffer; and

adjusting the level of priority associated with the particular query in a manner that promotes the particular query for execution over one or more other queries of the plurality of queries that originated from a widget that is not visible on the display of the mobile device.

36. The system of claim **34**, wherein adjusting, by the application programming interface, a level of priority that is associated with at least one query that is stored in the data buffer comprises:

determining, by the application programming interface, that a second software widget that was visible on a display of the mobile device is no longer visible on the display of the mobile device;

determining, by the application programming interface, that a particular query that originated from the second software widget is stored in the data buffer; and

adjusting the level of priority associated with the particular query in a manner that demotes the particular query for execution after each other query of the plurality of queries stored in the data buffer that originated from a software widget that is visible on the display of the mobile device.

37. A non-transitory computer-readable medium storing software comprising instructions executable by one or more computers which, upon such execution, cause the one or more computers to perform operations comprising:

determining, by an application programming interface, software widgets executing on respective mobile applications on a mobile device;

receiving, by the application programming interface executing on the mobile device, a plurality of queries, from the software widgets, for data from a database stored on the mobile device;

determining, by the application programming interface, a level of priority for each query of the plurality of queries;

adjusting, by the application programming interface, the level of priority, as an adjusted level of priority that is associated with at least one query that is stored in a data buffer providing, by the application programming interface, each of the plurality of queries for storage in the data buffer until processing resources of the mobile device are available to process one or more of the plurality of queries; and

processing each query of the plurality of queries against the database stored on the mobile device based on the adjusted level of priority.

38. The non-transitory computer-readable medium of claim **37**, wherein determining, by the application programming interface, a level of priority for each query of the plurality of queries comprises:

determining a level of priority for each query of the plurality of queries based on a timestamp associated with each query of the plurality of queries.

39. The non-transitory computer-readable medium of claim **38**, wherein the timestamp associated with each query of the plurality of queries indicates a time that the query was originated by a particular software widget of the software widgets.

40. The non-transitory computer-readable medium of claim **37**, wherein determining, by the application programming interface, a level of priority for each query of the plurality of queries comprises:

determining a level of priority for each query of the plurality of queries based on (i) a time stamp associated with the query and (ii) whether or not a particular software widget of the software widgets that originated the query is visible on a display of the mobile device.

* * * * *