

US008414387B1

(12) **United States Patent**
Paradise et al.

(10) **Patent No.:** **US 8,414,387 B1**
(45) **Date of Patent:** **Apr. 9, 2013**

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

(75) Inventors: **Andrew Paradise**, Boston, MA (US);
Casey Chafkin, Boston, MA (US)

(73) Assignee: **Lookout Gaming, LLC**

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **13/569,424**

(22) Filed: **Aug. 8, 2012**

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

(52) **U.S. Cl.** **463/25; 463/29; 463/42**

(58) **Field of Classification Search** **463/25,**
463/29, 42

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

2009/0209350 A1* 8/2009 Kelly et al. 463/42

* cited by examiner

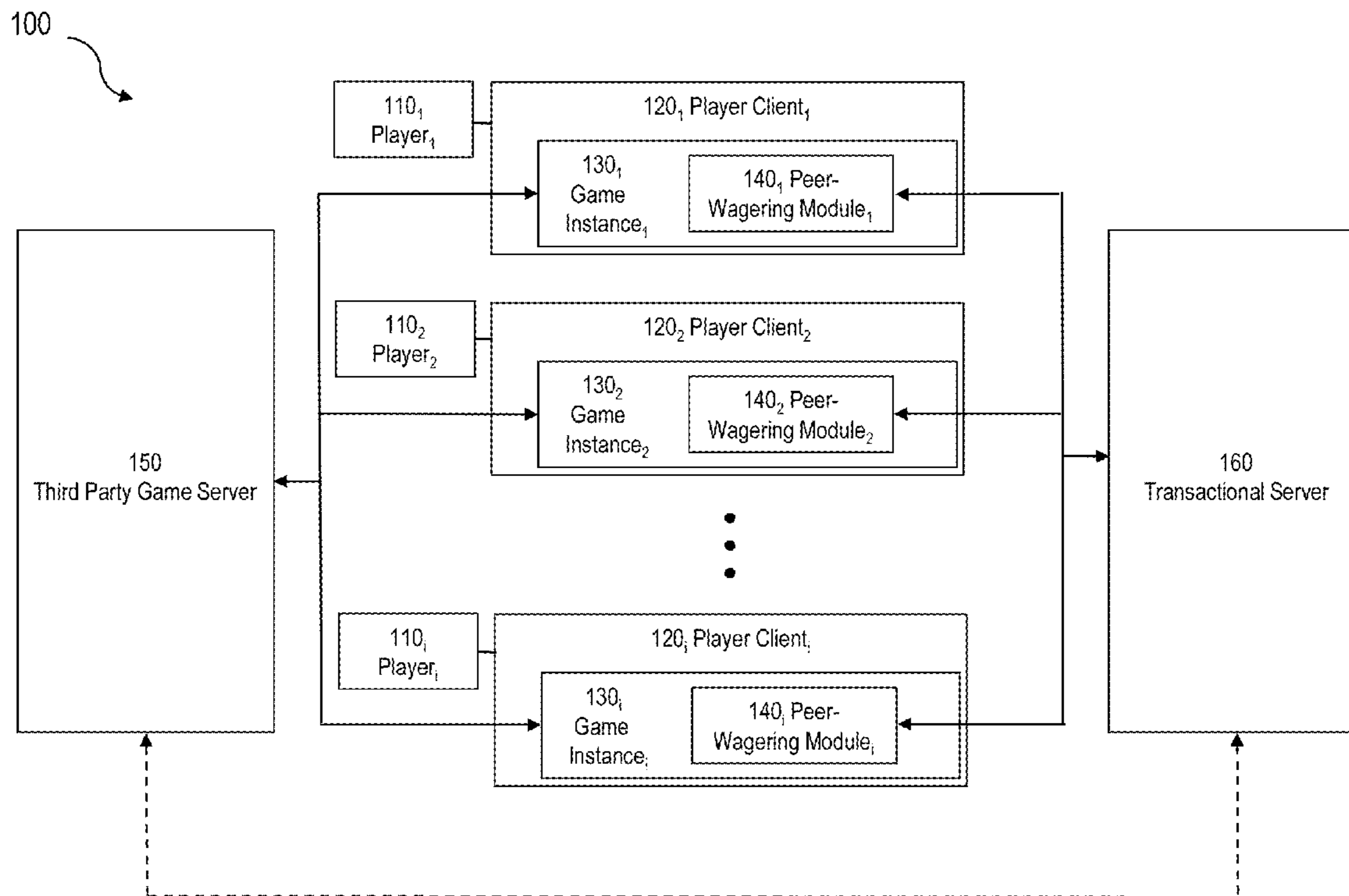
Primary Examiner — Omkar Deodhar

(74) *Attorney, Agent, or Firm* — Mintz Levin Cohn Ferris Glovsky and Popeo, P.C.

(57) **ABSTRACT**

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. Related apparatus, systems, techniques, and articles are also described.

53 Claims, 10 Drawing Sheets



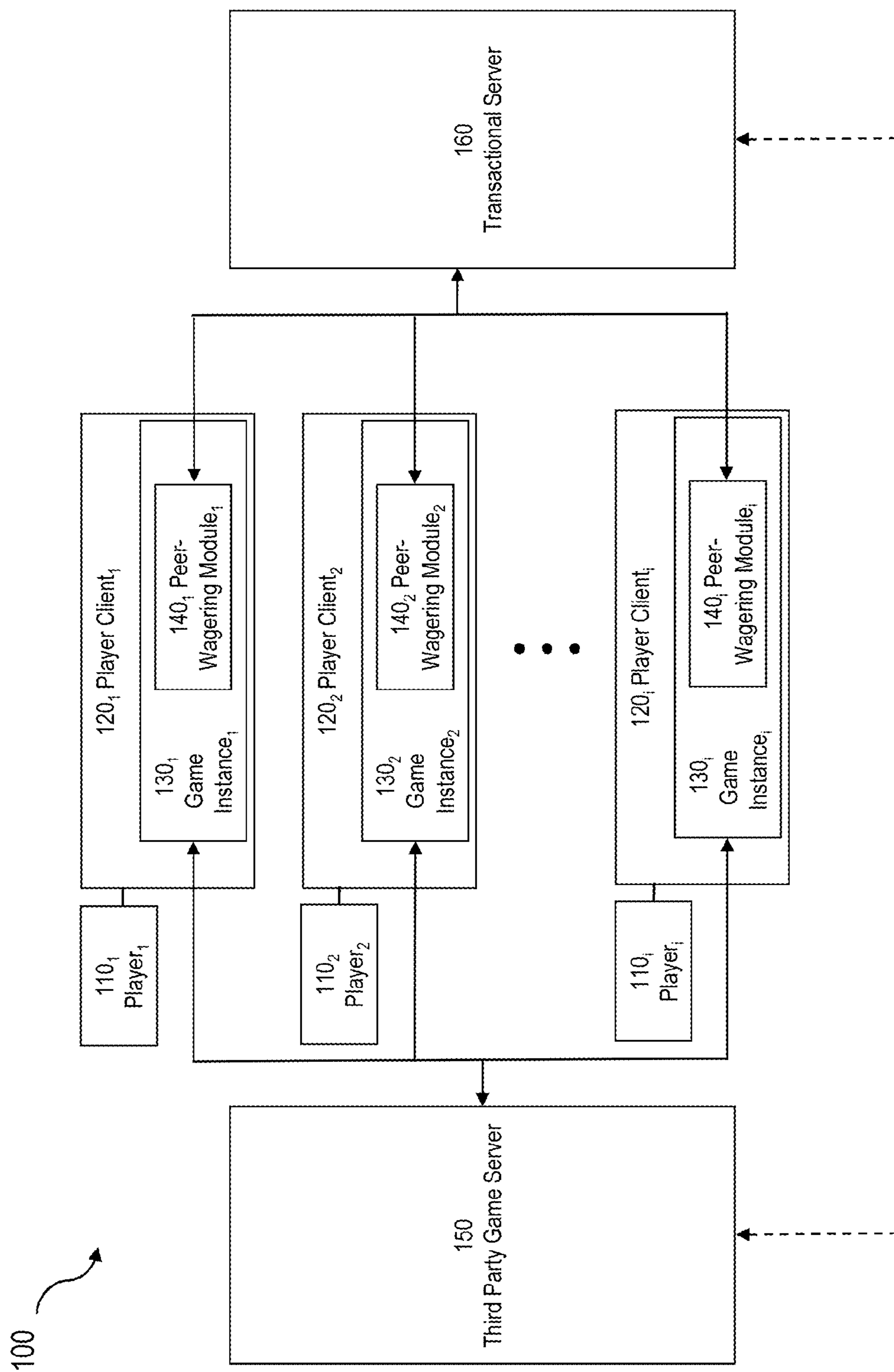


FIG. 1

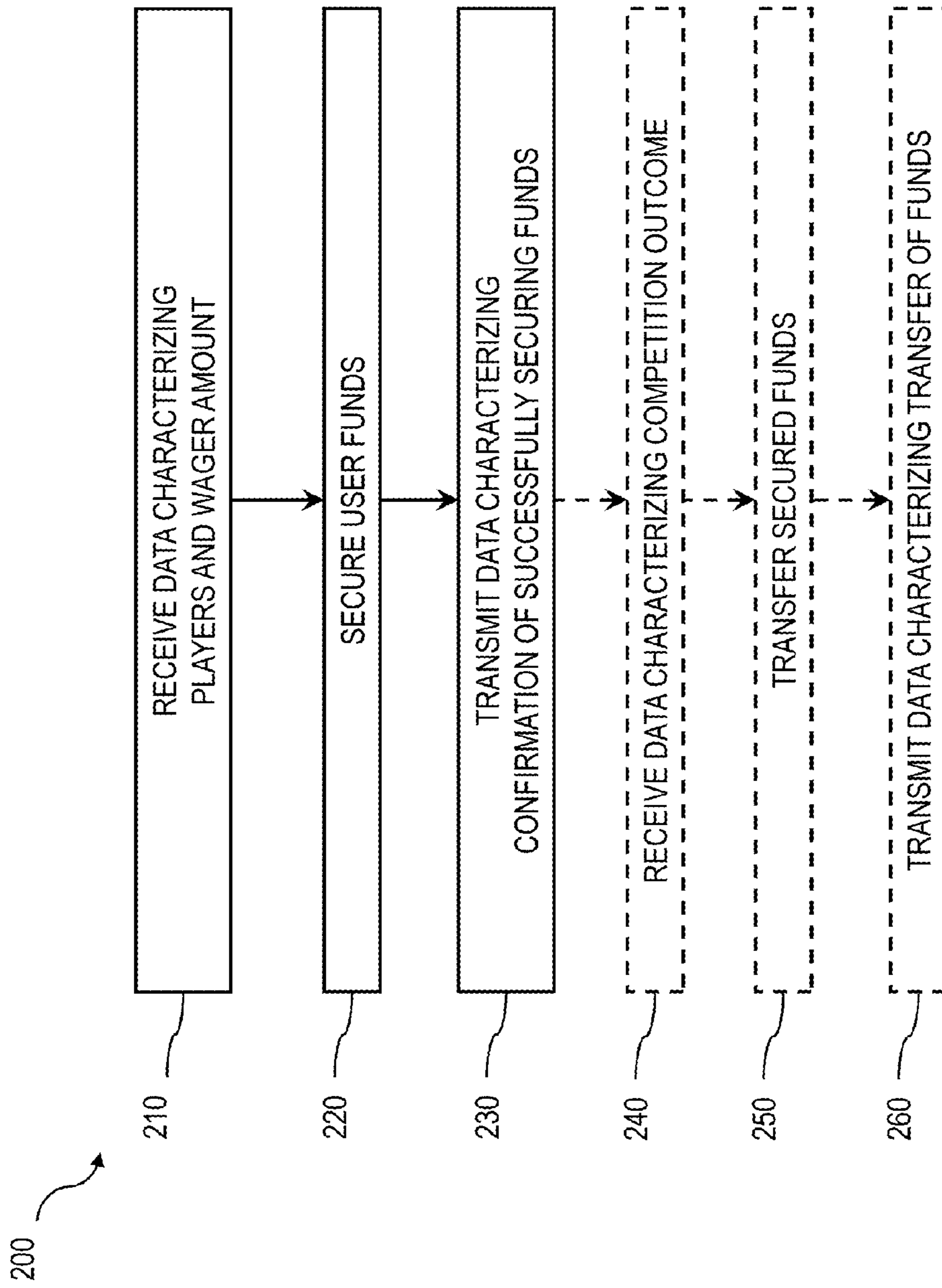


FIG. 2

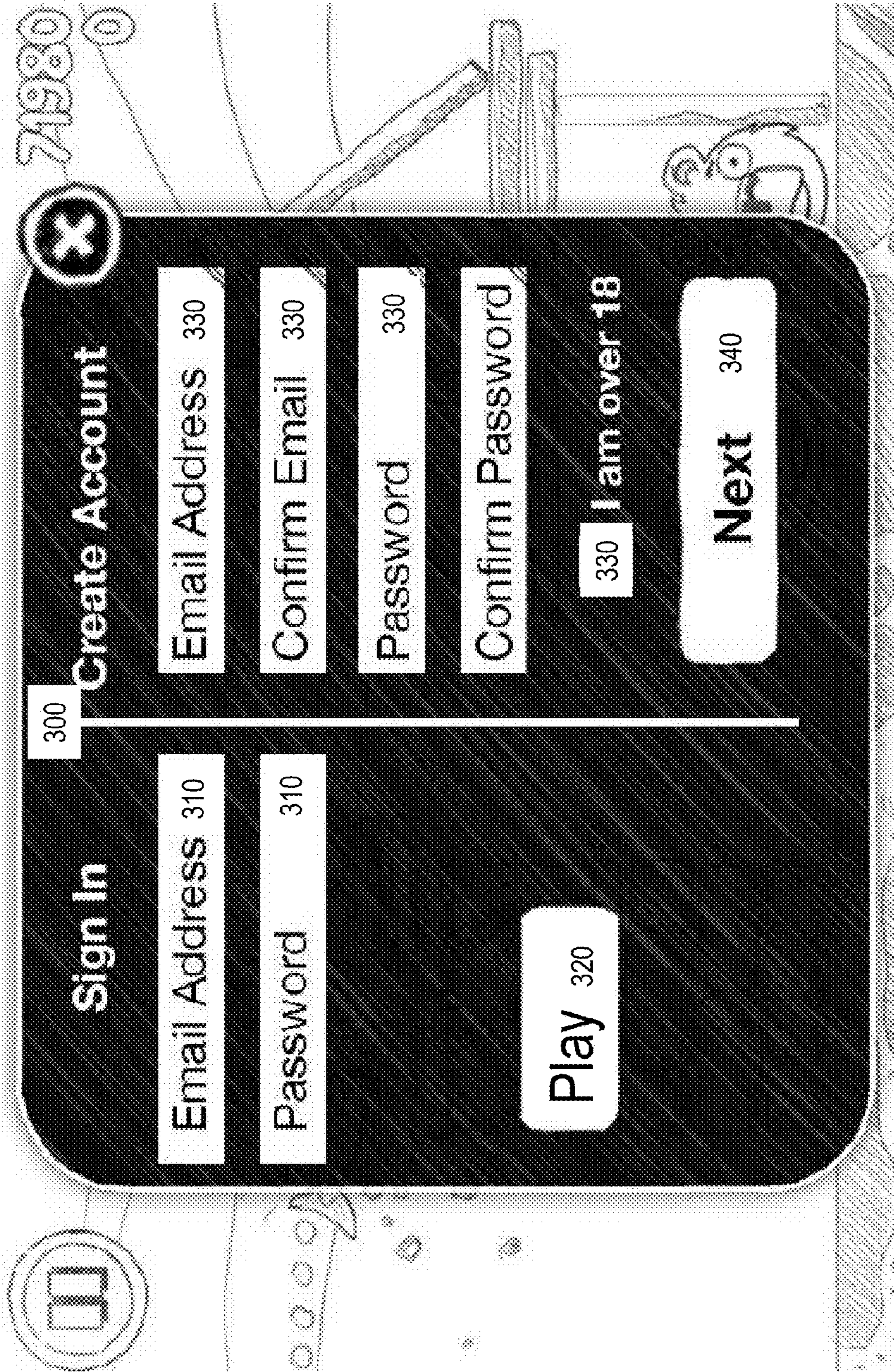


FIG. 3

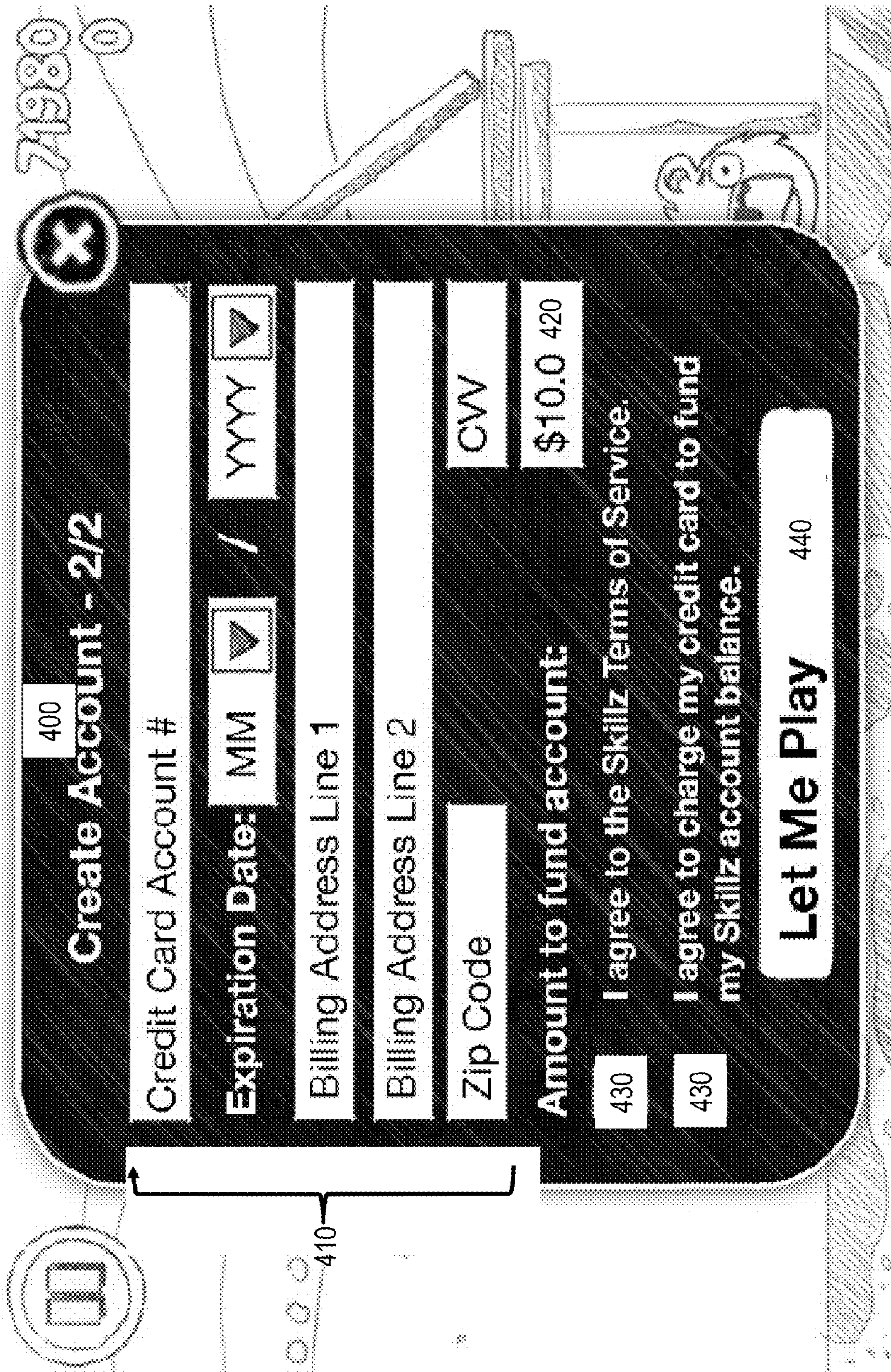


FIG. 4

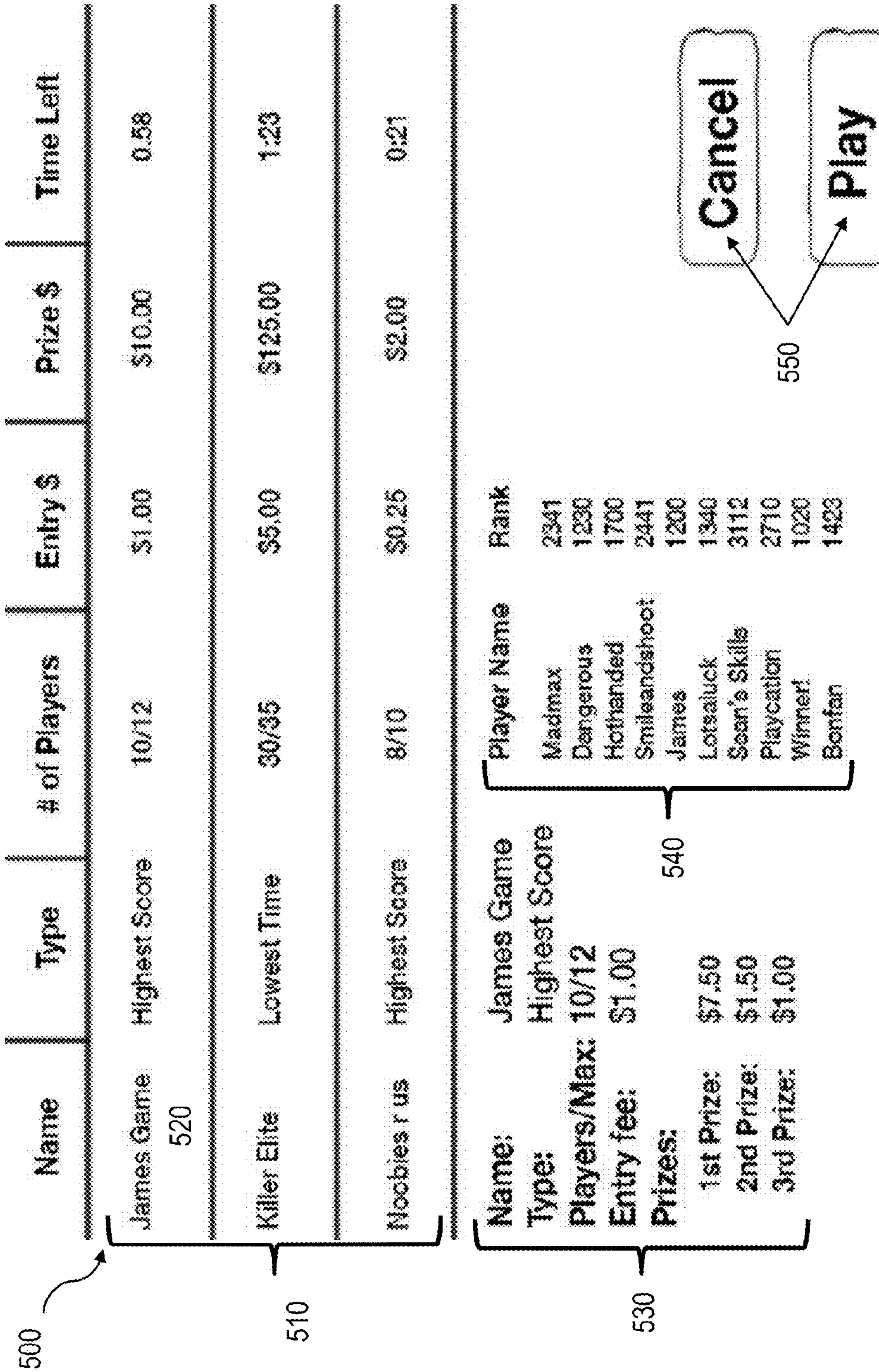


FIG. 5

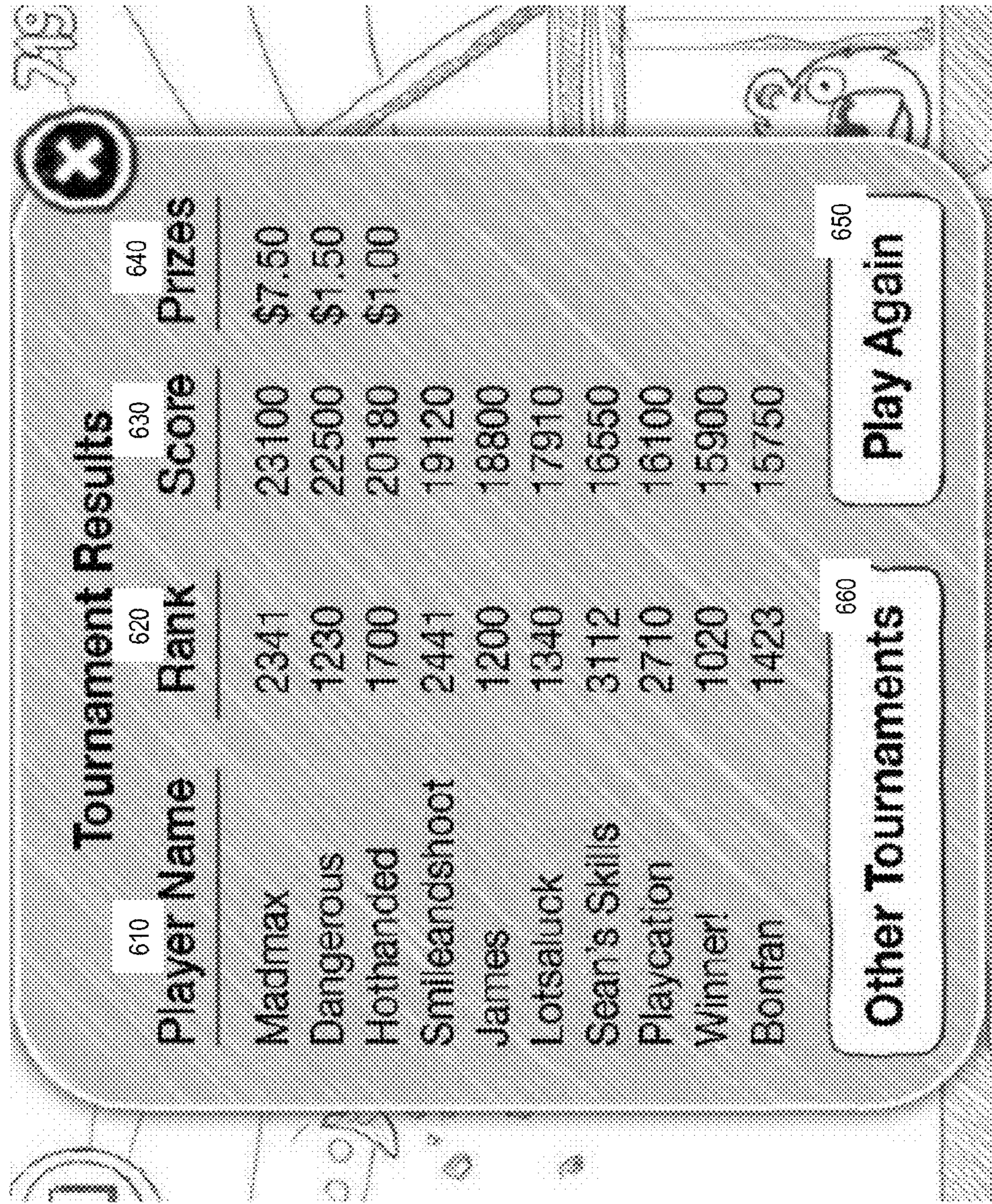


FIG. 6

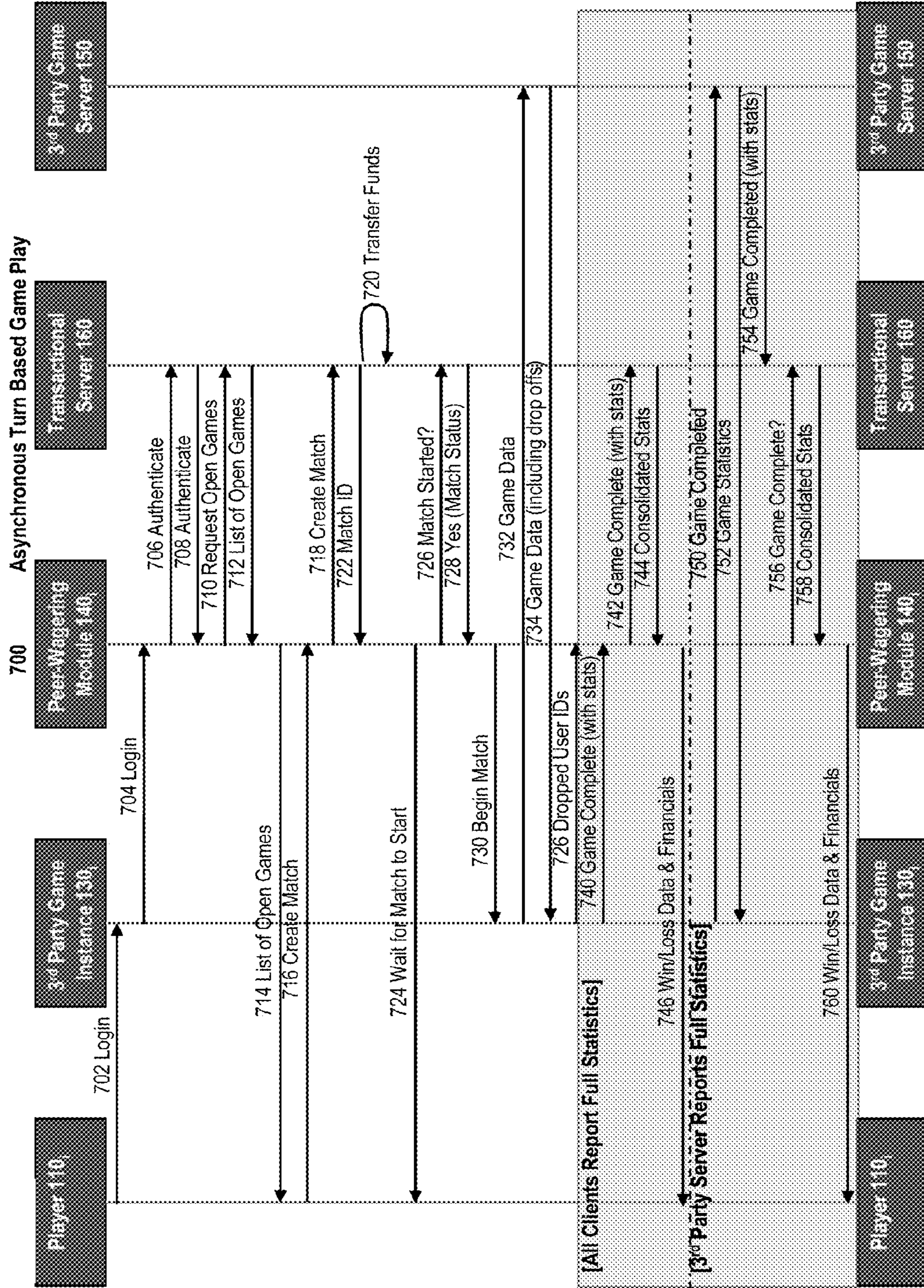


FIG. 7

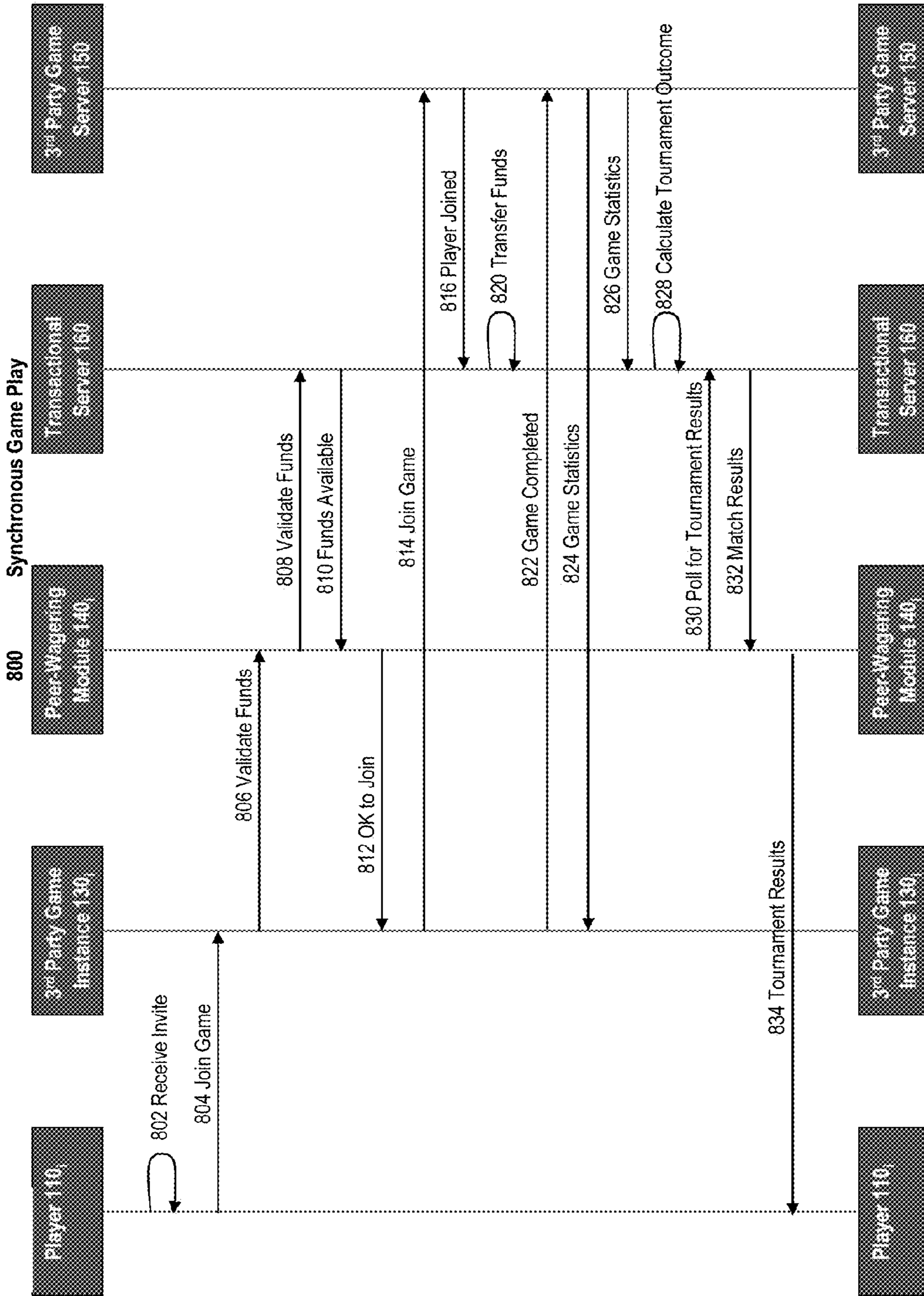


FIG. 8

900

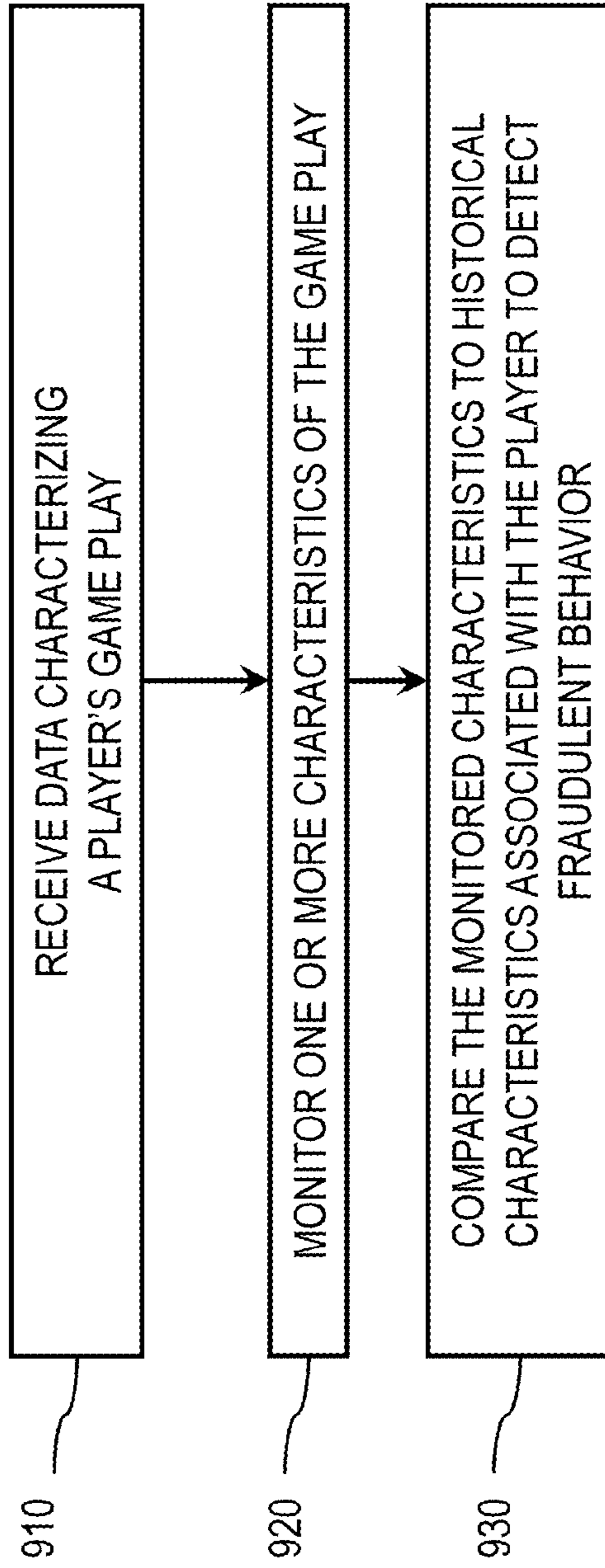


FIG. 9

1000

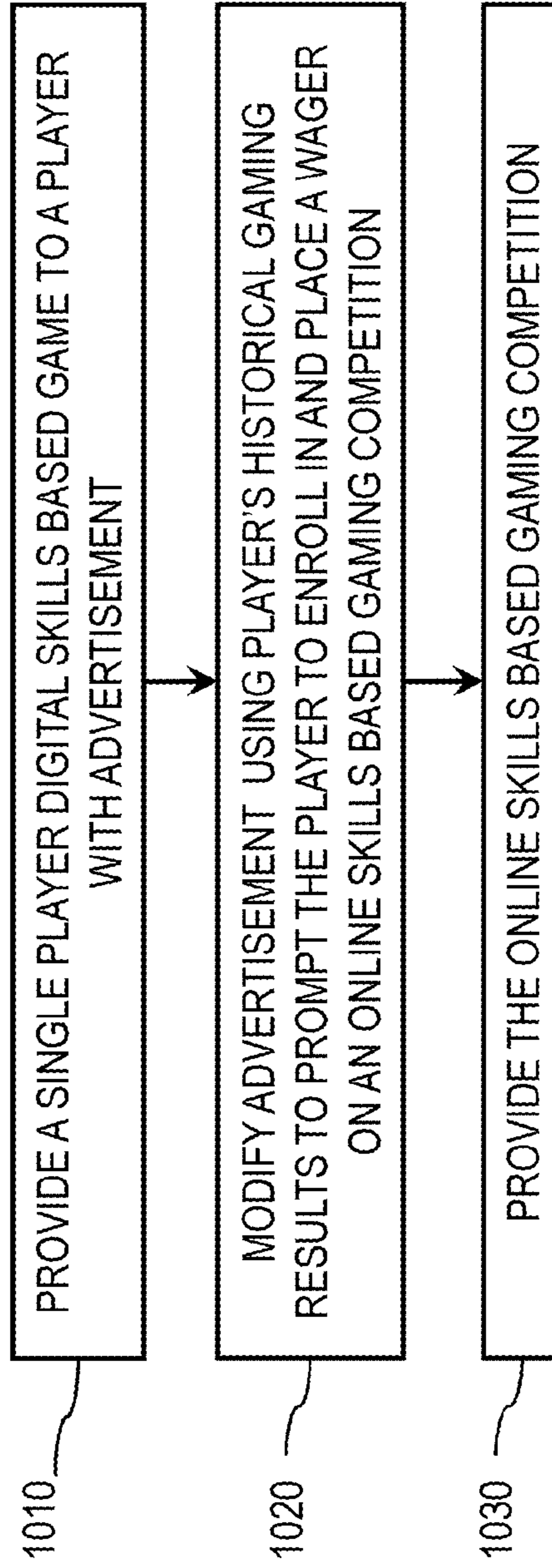


FIG. 10

PEER-TO-PEER WAGERING PLATFORM

TECHNICAL FIELD

The subject matter described herein relates to a peer-to-peer software platform that enables wagering on online skill based games.

BACKGROUND

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.

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

In one 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.

In 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.

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.

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 and losses 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.

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 characterizing 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 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.

The data characterizing the outcome of the third party skills 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.

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.

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.

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.

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

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

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

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;

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

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

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

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

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

FIG. 9 is a process flow diagram illustrating a method of detecting fraudulent behavior; and

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

Like reference symbols in the various drawings indicate like elements.

DETAILED DESCRIPTION

FIG. 1 is a system diagram 100 illustrating a system that facilitates wagering within electronic multiplayer games of

skill. A plurality of players 110, ($i=1, 2, \dots, N$) 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.

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).

The transactional server 160 can also pass advertising logic to the third party game server 150. This advertising logic can be algorithmically customized based on historical player 110_{*i*} results supplied by the 120_{*i*} player client. Advertising logic includes 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. For example, an advertisement could 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.” FIG. 10 is a process flow diagram 1000 illustrating a method of advertising an online skills based competition. At 1010, a single player digital skills based game is provided to a player. The game contains advertisements. At 1020, the advertisements are modified using the player’s historical gaming results to prompt the player to enroll in and place a wager on an online skills based gaming competition. At 1030, the online skills based gaming competition is provided.

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.

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.

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**.

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.

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).

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

FIG. 2 is a process flow diagram **200** illustrating a method of facilitating wagering within electronic multiplayer 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 transac-

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.

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**.

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.

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.

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.

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**.

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. Tourna-

ments with a variable number of participants can have a fixed starting time and can have a maximum number of allowed entrants.

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.

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.

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.

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.

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 .

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.

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 .

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.

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).

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.

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.

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.

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.

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.

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**.

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.

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.

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.

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.

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.

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_i** or the third party game server **150** may report **826** game statistics relevant to fraud detection to the transactional server **160**.

11

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_i, 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_i 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_i notifies 834 their respective player 110_i that the tournament has completed and they have either winnings or losses. The players 110_i receiving the completion notification can launch a new game via the notification.

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.

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.

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.

12

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.

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.

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.

What is claimed is:

1. A system comprising:

a third party gaming server providing data related to a third party game;

a plurality of third party game instances each third party game instance in communication with and receiving game data from the third party gaming server, each third party game instance including 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 associated with an online gaming competition, each player participating in the online gaming competition; and

a transactional server to receive from each peer-wagering module the wager amount, and secure an amount of funds previously deposited and associated with each of the associated corresponding players, the secured fund amount equal to the respective wager amount, the funds being secured such that they cannot be transferred, withdrawn, or secured for a different wager;

wherein each server comprises at least one computing system having at least one data processor.

2. The system of claim 1, wherein the transactional server receives game statistics describing the results of the online gaming competition from the third party gaming server and automatically transfers, 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.

3. The system of claim 1, wherein the third party game is an asynchronous competitive skill based game played between each player.

4. The system of claim 1, wherein the third party game is a synchronous competitive skill based game played between each player.

5. The system of claim 1, wherein the transactional server sends a notification of each player winnings and losses to the peer-wagering module in each of the plurality of game instances.

6. The system of claim 1, wherein the transactional server determines if the previously deposited funds associated with a player is less than the wager amount.

13

7. The system of claim 1, wherein the online gaming competition is a tournament.

8. The system of claim 1, wherein the online game competition is selected from a group consisting of: Massively Multiplayer Online Role Playing Games (“MMORPG”),
5 First Person Shooters (“FPS”), and Real Time Strategy (“RTS”) games.

9. A computer implemented method comprising:

receiving data characterizing a plurality of players and a
wager amount for each player, each player being
enrolled in a third party skills based digital gaming com-
petition;

securing for each player, using an online skills based digital
game wagering platform, an amount of funds previously
deposited by the respective player, the secured funds
equal to the wager amount, the funds being secured such
that they cannot be transferred, withdrawn, or secured
for a different wager; and

transmitting, data characterizing a confirmation that each
of the plurality of players funds were successfully
secured.

10. The computer implemented method of claim 9, further
comprising:

receiving data characterizing the outcome of the third party
skills based gaming competition;

transferring, using the online skills based digital game
wagering platform, at least a portion of the secured funds
to an account associated with one of the plurality of
players; and

transmitting data characterizing the transfer.

11. The computer implemented method of claim 10, fur-
ther comprising:

transmitting data characterizing the winnings and losses of
one or more of the plurality of players.

12. The computer implemented method of claim 10,
wherein at least one of the plurality of player’s is rewarded
with non-monetary compensation based on the outcome of
the third party skills based gaming competition.

13. The computer implemented method of claim 12,
wherein the non-monetary compensation can be used as a
substitute to funds to enter a future online skills based gaming
competition.

14. The computer implemented method of claim 10,
wherein the data characterizing the transfer is transmitted to a
plurality of peer-wagering modules, each module included in
an instance of a third party game.

15. The computer implemented method of claim 9,
wherein the data characterizing the confirmation is transmit-
ted to a plurality of peer-wagering modules, each module
included in an instance of a third party game.

16. The computer implemented method of claim 9,
wherein the data characterizing the outcome of the third party
skills gaming competition is 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.

17. The computer implemented method of claim 9,
wherein the data characterizing the outcome of the third party
skills gaming competition is received from a plurality of
peer-wagering modules, each module included in an instance
of a third party game in communication with a game server
providing game data to the plurality of game instances.

18. The computer implemented method of claim 9,
wherein the third party skills based gaming competition is
associated with an asynchronous game.

14

19. The computer implemented method of claim 9,
wherein the third party skills based gaming competition is
associated with a synchronous game.

20. The computer implemented method of claim 9,
wherein the wager amount includes a plurality of values, each
value associated with one of the plurality of players.

21. The computer implemented method of claim 10, fur-
ther comprising:

receiving data characterizing the player’s game play;

monitoring one or more characteristics of the received
data; and

comparing the monitored characteristics to historical char-
acteristics associated with the player to detect fraudulent
behavior by the player, wherein deviations between the
monitored characteristics and historical characteristics
indicate fraud.

22. The computer implemented method of claim 21,
wherein the monitoring is performed in real time.

23. The computer implemented method of claim 21,
wherein the monitoring is performed after the online skills
based game competition is complete.

24. A system comprising:

at least one data processor;

memory storing instructions which, when executed by the
at least one data processor, causes the at least one data
processor to perform operations comprising:

receiving data characterizing a plurality of players and a
wager amount for each player, each player being
enrolled in a third party skills based digital gaming
competition;

securing for each player, using an online skills based
digital game wagering platform, an amount of funds
previously deposited by the respective player, the
secured funds equal to the wager amount, the funds
being secured such that they cannot be transferred,
withdrawn, or secured for a different wager; and

transmitting data characterizing a confirmation that each
of the plurality of players funds were successfully
secured.

25. The system of claim 24, wherein the operations further
comprise:

receiving data characterizing the outcome of the third party
skills based gaming competition;

transferring, using the online skills based digital game
wagering platform, at least a portion of the secured funds
to an account associated with one of the plurality of
players; and

transmitting data characterizing the transfer.

26. The system of claim 25, wherein the operations further
comprise:

transmitting data characterizing the winnings and losses of
one or more of the plurality of players.

27. The system of claim 25, wherein at least one of the
plurality of players is rewarded with non-monetary compen-
sation based on the outcome of the third party skills based
gaming competition.

28. The system of claim 27, wherein the non-monetary
compensation can be used as a substitute to funds to enter a
future online skills based gaming competition.

29. The system of claim 25, wherein the data characterizing
the transfer is transmitted to a plurality of peer-wagering
modules, each module included in an instance of a third party
game.

30. The system of claim 24, wherein 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.

31. The system of claim 24, wherein the data characterizing the outcome of the third party skills gaming competition is 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.

32. The system of claim 24, wherein the data characterizing the outcome of the third party skills gaming competition is received from a plurality of peer-wagering modules, each module included in an instance of a third party game in communication with a game server providing game data to the plurality of game instances.

33. The system of claim 24, wherein the third party skills based gaming competition is associated with an asynchronous game.

34. The system of claim 24, wherein the third party skills based gaming competition is associated with a synchronous game.

35. The system of claim 24, wherein the wager amount includes a plurality of values, each value associated with one of the plurality of players.

36. The system of claim 25, wherein the operations further comprise:

- receiving data characterizing the player's game play;
- monitoring one or more characteristics of the received data; and
- comparing the monitored characteristics to historical characteristics associated with the player to detect fraudulent behavior by the player, wherein deviations between the monitored characteristics and historical characteristics indicate fraud.

37. The system of claim 36, wherein the monitoring is performed in real time.

38. The system of claim 36, wherein the monitoring is performed after the online skills based game competition is complete.

39. A non-transitory computer program product storing instructions, which when executed by at least one data processor of at least one computing system, implement operations comprising:

- receiving data characterizing a plurality of players and a wager amount for each player, each player being enrolled in a third party skills based digital gaming competition;
- securing for each player, using an online skills based digital game wagering platform, an amount of funds previously deposited by the respective player, the secured funds equal to the wager amount, the funds being secured such that they cannot be transferred, withdrawn, or secured for a different wager; and
- transmitting data characterizing a confirmation that each of the plurality of players funds were successfully secured.

40. The non-transitory computer program product of claim 39, wherein the operations further comprise:

- receiving data characterizing the outcome of the third party skills based gaming competition;
- transferring, using the online skills based digital game wagering platform, at least a portion of the secured funds

to an account associated with one of the plurality of players; and

transmitting data characterizing the transfer.

41. The non-transitory computer program product of claim 40, wherein the operations further comprise: transmitting data characterizing the winnings and losses of one or more of the plurality of players.

42. The non-transitory computer program product of claim 40, wherein at least one of the plurality of player's is rewarded with non-monetary compensation based on the outcome of the third party skills based gaming competition.

43. The non-transitory computer program product of claim 42, wherein the non-monetary compensation can be used as a substitute to funds to enter a future online skills based gaming competition.

44. The non-transitory computer program product of claim 40, wherein the data characterizing the transfer is transmitted to a plurality of peer-wagering modules, each module included in an instance of a third party game.

45. The non-transitory computer program product of claim 39, wherein 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.

46. The non-transitory computer program product of claim 39, wherein the data characterizing the outcome of the third party skills gaming competition is 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.

47. The non-transitory computer program product of claim 39, wherein the data characterizing the outcome of the third party skills gaming competition is received from a plurality of peer-wagering modules, each module included in an instance of a third party game in communication with a game server providing game data to the plurality of game instances.

48. The non-transitory computer program product of claim 39, wherein the third party skills based gaming competition is associated with an asynchronous game.

49. The non-transitory computer program product of claim 39, wherein the third party skills based gaming competition is associated with a synchronous game.

50. The non-transitory computer program product of claim 39, wherein the wager amount includes a plurality of values, each value associated with one of the plurality of players.

51. The non-transitory computer program product of claim 40, wherein the operations further comprise: receiving data characterizing the player's game play; monitoring one or more characteristics of the received data; and comparing the monitored characteristics to historical characteristics associated with the player to detect fraudulent behavior by the player, wherein deviations between the monitored characteristics and historical characteristics indicate fraud.

52. The non-transitory computer program product of claim 51, wherein the monitoring is performed in real time.

53. The non-transitory computer program product of claim 51, wherein the monitoring is performed after the online skills based game competition is complete.