



(51) International Patent Classification:
G06Q 50/00 (2006.01) **H04W 4/02** (2009.01)

(21) International Application Number:
PCT/US2010/025574

(22) International Filing Date:
26 February 2010 (26.02.2010)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
61/156,322 27 February 2009 (27.02.2009) US

(71) Applicant (for all designated States except US): **SCVN-GR, INC.** [US/US]; 580 Harrison Avenue, 4th Floor, Boston, MA 02118 (US).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **PRIEBATSCH, Seth, M.** [US/US]; 78 Marlborough Street, Boston, MA 02116 (US). **FINKELSTEIN, Adam** [US/US]; 18 Aiken Avenue, Princeton, NJ 08540 (US). **KARPOV, Valeri** [US/US]; 2-10 30th Street, Fair Lawn, NJ 07410 (US).

(74) Agents: **VALLABH, Rajesh** et al.; Foley Hoag LLP, 155 Seaport Boulevard, Boston, MA 02210-2600 (US).

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

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

Published:

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

(54) Title: COMPUTER-IMPLEMENTED METHOD AND SYSTEM FOR GENERATING AND MANAGING CUSTOMIZED INTERACTIVE MULTIPLAYER LOCATION-BASED MOBILE GAMES

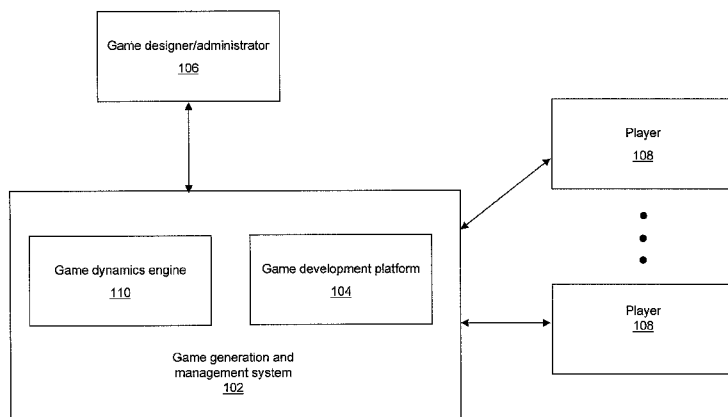


FIG. 1

(57) Abstract: A computer-implemented method and system are provided for generating and managing a customized interactive multiplayer location-based mobile game. The method includes the steps of: (a) receiving at a computer system content and high-level game logic from a game designer to be used in developing a location-based mobile game; (b) generating a playable location-based mobile game based on the content and high-level game logic using the computer system; (c) registering using the computer system a plurality of players desiring to play the multiplayer location-based mobile game using their personal mobile communications devices; and (d) managing execution of the game for each player or group of players by routing the player or group of players to a plurality of locations to perform a location-based task at each location, and interacting with the player or group of players after each of the location-based tasks has been performed, wherein managing execution of the game and interacting with the player or group of players comprises using the computer system to communicate with a personal mobile communications device of the player or group of players.



Computer-Implemented Method And System For Generating And Managing Customized Interactive Multiplayer Location-Based Mobile Games

CROSS REFERENCE TO RELATED APPLICATION

[0001] This application claims priority from U.S. Provisional Patent Application Serial No. 61/156,322, filed on February 27, 2009, entitled The HuntBuilder (A User-Driven Method for Dynamically Generating and Orchestrating Location Based Mobile Games), which is hereby incorporated by reference.

BACKGROUND

[0002] The present application relates to methods and systems for generating and managing customized interactive multiplayer location-based mobile games.

BRIEF SUMMARY OF EMBODIMENTS OF THE INVENTION

[0003] In accordance with one or more embodiments, a computer-implemented method is provided for generating and managing a customized interactive multiplayer location-based mobile game. The method includes the steps of: (a) receiving at a computer system content and high-level game logic from a game designer to be used in developing a location-based mobile game; (b) generating a playable location-based mobile game based on the content and high-level game logic using the computer system; (c) registering using the computer system a plurality of players desiring to play the multiplayer location-based mobile game using their personal mobile communications devices; and (d) managing execution of the game for each player or group of players by routing the player or group of players to a plurality of locations to perform a location-based task at each location, and interacting with the player or group of players after each of the location-based tasks has been performed, wherein managing execution of the game and interacting with the player or group of players comprises using the computer system to communicate with a personal mobile communications device of the player or group of players.

[0004] In accordance with one or more further embodiments, a computer program product is provided. The computer program product resides on a computer readable medium and has a plurality of instructions stored thereon which, when executed by a computer processor, cause that processor to: (a) receive content and high-level game logic from a game designer to be used in developing a location-based mobile game; (b) generate a playable location-based mobile game based on the content and high-level game logic; (c) register a plurality of players desiring to play the multiplayer location-based mobile game using their personal mobile communications devices; and (d) manage execution of the game for each player or group of players by routing the player or group of players to a plurality of locations to perform a location-based task at each location, and interact with the player or group of players after each of the location-based tasks has been performed, wherein managing execution of the game and interacting with the player or group of players comprises communicating with a personal mobile communications device of the player or group of players.

[0005] In accordance with one or more further embodiments, a computer system is provided for generating and managing a customized interactive multiplayer location-based mobile game. The computer system includes a game development platform and a game dynamics engine. The game development platform is configured to receive content and high-level game logic from a game designer to be used in developing a location-based mobile game, and to generate a playable location-based mobile game based on the content and high-level game logic using the computer system. The game dynamics engine is configured to register a plurality of players desiring to play the multiplayer location-based mobile game using their personal mobile communications devices. The game dynamics engine is also configured to manage execution of the game for each player or group of players by routing the player or group of players to a plurality of locations to perform a location-based task at each location, and to interact with the player or group of players after each of the location-based tasks has been performed. Managing execution of the game and interacting with the player or group of players involves using the computer system to communicate with a personal mobile communications device of the player or group of players.

[0006] Various embodiments of the invention are provided in the following detailed description. As will be realized, the invention is capable of other and different embodiments, and its several details may be capable of modifications in various respects, all without departing from the invention. Accordingly, the drawings and description are to be regarded as illustrative in nature and not in a restrictive or limiting sense, with the scope of the application being indicated in the claims.

BRIEF DESCRIPTION OF THE DRAWINGS

[0007] FIG. 1 is a simplified illustration of an exemplary network in which a game generation and management system in accordance with one or more embodiments can be implemented.

[0008] FIG. 2 is a simplified flowchart illustrating an exemplary player registration process in accordance with one or more embodiments.

[0009] FIG. 3 is a simplified flowchart illustrating an exemplary game execution process in accordance with one or more embodiments. (FIG. 3 is comprised of FIGS. 3A and 3B, which are collectively referred to as FIG. 3.)

DETAILED DESCRIPTION

[0010] The present application is directed to a computer implemented method and system for generating and managing or orchestrating the execution of complex customized interactive multiplayer location-based mobile games. By way of example, such games can include so-called scavenger hunts, in which individual players or teams of players are routed to a series of locations to perform location-based tasks (such as collecting items, answering questions, or taking photographs at the location). In accordance with one or more embodiments, a computer system is provided for generating the game and managing execution of the game by interacting with players through their personal mobile devices.

[0011] The system can also be used to generate and manage a variety of games other than scavenger hunt type games. The term “game” is used herein to generally describe a variety of interactive location-based activities involving

players, groups, or teams. These activities can be a variety of fields and can be used for various purposes as described in further detail below.

[0012] FIG. 1 is a simplified illustration of an exemplary network in which computer-implemented game generation and management system 102 in accordance with various embodiments can be implemented. The game system 102 includes a game development platform or game builder 104 that can be used by a game designer or organizer 106 to generate customized interactive location-based games as described in further detail below. The game designer or organizer 106 can, e.g., be any individual desiring to build and/or manage an interactive location-based game for a variety of purposes. For example, the game designer or organizer 106 can be an individual associated with an organization, place, or event (e.g., a university, museum, company, city, or convention) wishing to use the game to provide an interactive experience associated with a place for multiple players.

[0013] For ease of explanation, the term “game designer” is used herein to identify someone building or organizing the game, and the term “game administrator” is used to identify someone managing the execution of the game. It should be understood, however, that in many cases, the game designer and game administrator will be the same person.

[0014] The games can be played by multiple players (individually or in groups or teams) through their personal mobile devices 108. The mobile devices 108 can include a variety of portable communications devices such as, e.g., cellular telephones, smart phones, portable digital assistants, and the like. By way of example, a mobile device 108 can include an SMS compatible cellular phone or a smart phone capable of accessing the game through a mobile web portal.

[0015] The game system 102 can communicate with game designers or administrators 106 and players 108 over a variety of communications networks including, without limitation, wired and wireless networks such as the Internet, intranets, cellular communications networks, mesh networks, wireless local area networks, and other connections.

[0016] The game system 102 can include an enterprise-level server or other computer server system for performing the functions described herein. The

functions of the game system 102, including generating games through the game development system 108 and managing execution of games through a game-dynamics or game-play engine 110, can be distributed across one or more virtual or physical computer systems.

[0017] The game development platform 104 allows non-technical game designers to quickly and easily build sophisticated location based mobile games using software that is accessible online. Users can input their custom game data and location-based content and program in high-level game logic, which the game development platform 104 then interprets. The platform 104 can also implement pre-loaded game-dynamics as will be described in further detail below. In accordance with one or more embodiments, the game can be deployed across virtually all mobile devices 108 and all carriers. The game-dynamics engine 110 manages execution of fast-paced game-play interactions, and can automate scoring and tracking and mapping user-submitted photos to clue/challenge tasks and associated locations.

[0018] The platform 104 allows non-technical users to take advantage of powerful pre-built game components and lets them program in high-level game logic, thereby allowing dynamic interactive games to be created with ease. By allowing users to customize location based content using the game development platform software, location based games can be built without the expense and complexity of doing it by hand. By automatically deploying games across all carriers and all mobile devices, the system 102 removes the time and effort needed to code for various mobile devices or achieve carrier provisioning for mobile messaging. By utilizing personal devices of individual players, the system 102 avoids the expense of purchasing and distributing (and perhaps even producing) proprietary mobile devices. Additionally, the game engine 110 can handle dynamic two-way interactions involving game-logic and user generated response on the fly, removing the need for vast manpower resources to monitor and manage the game and enabling easy scaling to increase the number of participants. The system 102 thereby makes building location-based mobile games faster, less expensive, and easier for the game organizers or designers, and the fast-paced dynamic content makes the game more enjoyable to play for the participants.

[0019] By way of example, the game development platform 104 in accordance with one or more embodiments can be used to build games as follows:

[0020] The user, i.e., game designer or organizer, is prompted to input their game content and to select one of several built-in game dynamics. For example, the content requested is delineated below and is preferably, though not necessarily, inputted in this order:

[0021] The game designer can select from one of the pre-packaged game dynamics. The game designer can then continue through the process and customize it as desired. Four exemplary prepackaged game dynamics, which are described in greater detail below, are identified as Hunt, SCRMBL, Adventure, and ToDo.

[0022] The game designer is then prompted to input game information, including, e.g., a keyword that players will send to the system 102 to enroll in the game, a game title to provide a reference to the game administrator, a game location to help the system geo-code and center the map's game area.

[0023] The game designer is then prompted to enter information on the duration of the game including, how long the game will last for each team. The game designer also indicates when the game is active to start being played by participants. The game designer indicates whether the game has a defined start and end-time or is an ongoing event.

[0024] In addition, for SCRMBL games (which are described in further detail below), the game designer is asked to enter the following information: the point threshold to trigger the scramble, the time threshold to trigger the scramble, the initial time offset to award to the first team, the penalty per point in seconds to deduct from other teams, and the number of minutes (after the first team finishes the final riddle) the game ends.

[0025] The game designer can input a message that goes out to all participants to start off the game, as well as a message that goes out to all participants to close the game.

[0026] A warning message can be input that goes out to all participants a certain number of minutes before the game ends. This applies only to non-SCRM BL game dynamics.

[0027] For SCRM BL games, winning and losing messages can be input by the game designer.

[0028] The game designer can also specify whether the game is open or private. This defines whether or not players need a password to participate in the game. If the game designer selects private, then he or she is prompted to enter an access password, which participants will need to register for the game.

[0029] The game designer can specify a particular routing algorithm, which will be described in further detail below, to be used for the game. If the game designer is building an Adventure game, he or she will be building their own routing algorithm and therefore this field will not show up.

[0030] The game designer can specify whether or not the participants start the game by sending in a start message, or when the administrator starts the game by sending in special pass-code. The game can also be auto-started, in which case the game starts for the player as soon as he or she registers.

[0031] The game development platform 104 includes a Clue/Challenge Sequence Creator in accordance with one or more embodiments. A clue is a riddle that hints at a location and requests, as its answer, the name of the location. A challenge is typically a more difficult riddle that is intended to be solved at that specific location, thus impelling the participant to actually travel to the location. Clue/Challenge sequences can contain multiple sub-elements. The clue/challenge sequences are translated into loosely connected sub-graphs, which players travel through and then (using the preselected routing dynamic are transported to another clue/challenge sequence to continue their game).

[0032] The game designer inputs the title of each clue/challenge sequence for reference, the text of the clue, and an optional hint. The designer also inputs answers, which can be comma delimited for each clue. The system 102 can be configured to handle misspelling.

[0033] The game designer also inputs the text for the first challenge, and can choose to add additional challenges. The game designer can also input an optional hint for each challenge. The game designer also inputs the answers for the challenge. If the challenge is a photo based challenge (i.e., one that requires a player to send in a photo), then any answer sent in can be deemed acceptable. In an Adventure game, the game designer also specifies to what clue a correct answer sends the players and to what clue an incorrect answer would send the players. The game designer can also choose to break their answers into different answer sets and associate a next clue location with each given answer set.

[0034] The game designer can specify the number of points each component of the clue/challenge is worth or the number of points to be deducted for using hints and providing wrong answers. The game designer then geo-tags the location of that clue/challenge sequence.

[0035] In accordance with one or more embodiments, a landing page can be provided for game designers that can be used by participants to register for the game using a website instead of registering from their cell phones. The landing page can include text and images input by the game designer.

[0036] In accordance with one or more embodiments, the game development platform 104 can run a simulation of a game designed by game designer, and provide statistics about the game and offer suggestions based on the statistics analyzed against past games that have been run. The platform 104 can use a system of metrics culled from a large number of anonymized past interactions to come up with these suggestions.

[0037] Using either a cellular phone or an iPhone (or other smart phone) emulator on the screen, the game designer can demo their game and then edit it, if needed, before deploying it.

[0038] The game development platform 104 takes the game content and high level game logic entered in by game designers and runs it through a provisioning process to compile it into a playable game that can be played from virtually any cell phone or mobile device 108. The game can be deployed across multiple platforms (e.g., SMS, Mobile Web, Native Phone Applications). The games are provisioned

with generally all the major carriers to be playable from any SMS-compatible cell phone. The games can also be deployed on a mobile web portal for smart-phones and on native phone applications, e.g., the iPhone, and Android, Symbian, and RIM phones.

[0039] FIG. 2 illustrates an exemplary player registration process in accordance with one or more embodiments. In this example, a team of players is registered. A similar process can be used to register individual players. As shown in FIG. 2, a user desiring to register a team can register for the game by text message (at step 204), mobile web (at step 206), or by a native phone application (e.g., iPhone App) (at step 208). If the user registers by text message, he or she sends a given keyword to a specified number as illustrated at 210. If the user desires to register by mobile web, he or she can navigate to a particular site to register as illustrated at 211. If the user desires to register by a native phone application, he or she can download the native phone application, open it, and use a given keyword to register for the game as illustrated at 212. At step 214, the user receives a response with a team code and the welcome message entered by the game designer.

[0040] At step 215, other team members may join the user's team by sending in the team code. The other team members can choose to become pilot or copilot of the team, if desired. Each team member receives a response with the welcome message.

[0041] If the game was set up as a user-started a game, the pilot of the team can start the game by sending in a given start message, e.g., "SCVNGR START" to begin the game (at step 218). The game then begins. If the game designer had set up the game as auto-start, then the players skip to the next step as indicated at 220. If the game designer had specified that the game be started only by the game designer or administrator, then the administrator sends in a special command at 222 to begin the game and all of the players would then be moved to the next step.

[0042] FIG. 3 illustrates an exemplary game execution process in accordance with one or more embodiments. The game dynamics engine 110 analyzes the game-content and game-area inputted by the game designer and dynamically

selects a first clue/challenge sequence for the team at step 302 based on the routing-dynamic selected by the game designer and the relevant algorithm for that dynamic. If a linear routing was specified, the engine 110 follows the order specified by the game designer at 304. If "Smart Route" routing (which will be described in further detail below) was specified, the engine 110 selects a clue based on parameters such as the current positioning of other teams and proximity to next locations at step 306. If random routing was specified, the engine 110 selects randomly among available clues at step 308. If Adventure routing was specified, the engine 110 finds the starting clue at 309 as defined in a custom routing algorithms specified by the game designer.

[0043] At step 310, the first component of a clue/challenge is delivered to the team members (pilots and copilots).

[0044] At step 312, the team solves the first component and submits its answer. The game dynamics engine 110 analyzes the answer, preferably using an edit-distance program to account for misspellings, and compares the answer against the game designer inputted answer variations to attempt to determine if the answer submitted is correct or incorrect.

[0045] If the answer is incorrect, the team loses points and receives notification of their incorrect answer. Depending on the number of times that this clue can be answered incorrectly (as specified by the game designer), the team may be given multiple attempts. The game dynamics engine 110 decrements the number of attempts available to the user at step 314 for each answer provided. If the user still has attempts left, the user receives notification that they were incorrect and loses points at 316 and is given an opportunity to submit another response at 312. If there are no attempts left, the user can be notified at 318 and provided the next clue at 302.

[0046] If the answer is correct, the user is awarded points at step 320. The game dynamics engine 110 then searches for any challenges or other components associated with this clue. If one is found, that challenge is delivered to the team at step 322. The team submits their response at step 312.

[0047] If this is the final challenge for this clue/challenge sequence, then the team is sent to step 302.

[0048] At any point during the game, the teams can access various interactive commands including, e.g., the following: SCORE (to get their current score and rank), HELP (to get help with the game), REPEAT (to get their last message again), PILOT (to become the team pilot), COPILOT (to become a copilot), HINT (to get a hint on their current clue/challenge), SKIP (to skip to the next clue/challenge at step 302), QUIT (to quit or exit from the game entirely).

[0049] At any point during the game, the game administrator can access a variety of interactive commands, including, e.g., the following: RANK N (to get the top N ranked teams, where N is a positive integer), and BROADCAST <Keyword> <Message> (to send anything in the <message> parameter to all of the teams currently in the game with keyword equal to the <keyword> parameter).

[0050] If the game administrator has inputted a warning message to go out a certain period of time before the game is over (and the game is not using a SCRMBL game-dynamic), then all teams will receive the warning message when they have that period of time left.

[0051] When teams run out of time or finish the game by completing all of their remaining clues at 334, they will receive a notification with their score and an ending message at 336 that was entered during game development.

[0052] After the game, the participants may sign up for another game, or send in SCORE to receive their final score again.

[0053] A post game summary report can be sent to game administrators (e.g., to their online accounts), which allows them to view details about the game such as: the teams, their players, their scores, the paths they took, how long they played for, and a number of statistics about the game including the average number of clue/challenge sequences solved, etc. These statistics can be calculated after the first team finishes the game and are updated as other teams finish.

[0054] In an ongoing game having an extended duration (e.g., one that lasts for several days, weeks, or months), it is possible for a game administrator to examine the post-game review data and then determine that parts of the game should be edited. In this case, the game administrator may return to the game development process, select and edit a desired field, and then have the revised game be active for the next round of the game.

[0055] The four exemplary game dynamics mentioned above (Hunt, SCRMBL, Adventure, and ToDo) are now described in further detail.

[0056] Hunt: This game-dynamic can include any number of clue/challenge sequences. Teams can be routed from one clue/challenge sequence to another based on the routing algorithm chosen by the game designer. The routing algorithm choices can be linear (in which the clue/challenge sequences are delivered in a specific order one after the other), random (in which a teams next clue/challenge sequence is chosen randomly), or by a so-called SmartRoute routing algorithm. SmartRoute can dynamically generate unique but intelligent paths for each team, generally ensuring that they travel through an optimal path based on several parameters, including distance to other potential locations, clustering of teams at other potential locations, and other parameters such as the relative importance of those locations. SmartRoute is described in further detail below.

[0057] SCRMBL: A SCRMBL (pronounced 'scramble') is a game dynamic that involves ongoing calculations to repeatedly alter timing for delivery of clues in response to interactive user-generated events. Every team's actions during the game can have the opportunity to affect the game path that other teams experience. From the user perspective, the game can operate, e.g., as described below.

[0058] SCRMBL game operates in a similar fashion to a regular scavenger hunt with teams solving clue/challenge sequences and being routed to the next clue/challenge sequence based on, e.g., the SmartRoute routing algorithm. However, rather than trying to win by accruing the most points in a given time period, teams in a SCRMBL are racing to reach a point threshold. The first team to

reach this point threshold receives the final clue (which can, e.g., be an especially difficult riddle) and also triggers a “land-rush” phase of the game. During the land-rush phase, every other team also begins receiving the final clue after a time-delay based on their current score. The difference between any team’s current score and the point threshold can be multiplied by some administrator-supplied constant (in seconds of delay / point) and added to an initial time offset in seconds (also supplied by the game administrator) to calculate the time delay that a team must suffer before receiving the final clue. During this time, the team may continue to solve other clue/challenge sequences to earn more points and thus decrease their time delay. The SCRMBL also has a time-threshold that can trigger the land-rush phase of the game. If no team receives enough points to trigger the point threshold before the time-threshold is reached, then the scramble will be triggered and the final clue will be delivered according to the method described above.

[0059] The Adventure game dynamics enables a game designer to easily program in game logic to create a customized low-level interactive routing-algorithm. This ability offers great flexibility to game administrators, enabling them to build reactive games that can receive user input and alter their path based on that input.

[0060] In the Adventure framework, the game designer builds out clues following the protocol described above, but also includes additional pieces of data. The designer defines what next location the team will be routed to if they complete the clue/challenge sequence correctly, where they will be routed if they complete it incorrectly, and optionally, where they will be routed based on different answers or answer sets (e.g., comma delimited strings of acceptable answers to initiate a given routing action). As an example, a designer may decide that answering RED to a clue should route the team to clue/challenge sequence 7, while BLUE would route them to clue/challenge sequence 8, and GREEN might route them somewhere else entirely. An incorrect answer, might also take the team to some other part of the game. This system offers low-level routing logic to be programmed into a complex system.

[0061] ToDo is a game dynamic that supplies all of the clue/challenge sequences to players immediately and enables them to solve them in the order

they desire. The system 102 can provide instantaneous feedback to the participants as they solve the clues and then deliver the associated challenges for that location.

[0062] The SmartRoute routing algorithm can free game administrators from having to handle game-routing logic and can provide intelligent routes for each team. SmartRoute seeks to optimize the routes of each team based on given parameters such as the proximity to available clue/challenge sequences and the clustering patterns of teams at those locations. Other parameters can include user-supplied information about the relative importance of those locations (whether or not they want participants to reach them with high probability) and difficulty of the clue/challenge sequence at that location. By optimizing with even just the two parameters mentioned above, SmartRoute can dynamically generate unique paths for each team that can be randomized (and made unpredictable for the end user), but intelligent (meaning that they travel, e.g., to nearby locations that are not currently crowded with other teams). This routing leads to enjoyable game play with teams not running into or following one another and also not being forced on overly long treks to reach their next location. The results of this routing pattern are particularly valuable in moderate sized game areas such as, e.g., university campuses and cities and areas where clustering could be problematic, such as museums and other indoor spaces.

[0063] By way of example, using those two initial parameters as a guide, the following is a mathematical description of how the scoring of each of the available locations is calculated according to the SmartRoute routing algorithm:

[0064] To choose what location to send a team to next, the algorithm begins by assigning a relative score s_i to all possible next locations i . Previously visited locations have a score of zero, and otherwise the score balances between distance traveled and the goal of avoiding clustering, as follows:

[0065]

$$s_i = \left(k_1 \left(\frac{d_i}{D} \right) + \left(\frac{n_i}{e_i} \right)^{k_2} \right)^{-k_3}$$

[0066] where:

[0067] • d_i is the distance to location i ,

[0068] • D is the diameter of the entire game (or maximal distance between any two locations),

[0069] • n_i is the number of teams that would be at location i if this team went there (i.e., one greater than the current number),

[0070] • e_i is the expected number of teams at location i (it could be set globally, e.g., as the total number of teams divided by total number of locations, or could be set or adjusted per location by hand to cause some locations to get more or less clustering),

[0071] • $k_1 \geq 0$ is a constant that says how important distances are relative to clustering (high values favor short distances whereas low values more strongly avoid clustering, and good values in many cases seem to range between 3 and 40)

[0072] • $k_2 \geq 0$ is a constant that shapes the response to clustering ($\frac{1}{2}$ to 2 can be a good range in many cases)

[0073] • $k_3 \geq 1$ is a constant that shapes the overall scoring function (values between 2 and 3 work well in many cases, and in the limit of a large k_3 , say over 10, a single largest score will dominate all others).

[0074] To see the impact of distances, consider the exemplary case where all locations have the same n_i , and for simplicity assume that it is e_i so the second term in parentheses is 1. Also, suppose a specific d_i is small, say very near zero. In this case the score will be 1. On the other hand, if d_i is large, then the score will be a large number to a negative power, i.e., very small.

[0075] To see the impact of the clustering term, consider the case where all distances are the same; it is easiest to suppose they are all 0. Then all scores

become $\left(\frac{n_1}{e_1}\right)^{-k_2 k_3}$.

[0076] So for locations that have few teams there already, the score will be high, and for locations with many teams there, the score will be very low. The exact shape of that function depends on the constants -k3 is discussed above, and k2 says how much the algorithm will work to avoid over-crowding at some location.

[0077] The next step is to choose randomly among all possible next locations, with probability p_i proportional to the scores:

$$p_i = s_i / \sum_j s_j$$

[0078]

[0079] This may be accomplished in linear time by calculating the cumulative distribution function:

$$c_i = \sum_{j \leq i} p_j$$

[0080]

[0081] next choosing a random number r between 0 and 1, and finally selecting the lowest value of i for which $c_i \geq r$.

[0082] The game generation and management system 102 in accordance with various embodiments can be used in a variety of fields for multiple purposes. These include, e.g., the educational field, where the technology can be used to conduct campus-wide activities, orientations, team-building exercises, and educational tours. In the corporate world, the system 102 can be used, e.g., for high-tech promotions and for executive team-building. Museums and cultural institutions can use this system for inexpensively implementing interactive mobile tours. Cities can use the system for a similar purpose to offer self-guided user-driven tours around a tourist site. Conventions can use the system 102 to guide conventioners around a conference or booth-floor and drive booth attendance in an easily trackable fashion. In addition, the system 102 can be used by individuals for many creative uses, generally centered around building location-based mobile games to be played by friends and family or to be deployed on the system 102 and played by the community at large.

[0083] Some of the underlying technology in the game-play engine 110 that handles the routing of teams based on their current location, proximity to other locations, the clustering of other teams and many other variables, has applications far beyond those listed above. The dynamic nature of this routing algorithm can make it valuable in situations that need one central source to control the paths of many other groups within a pre-defined area. This routing technology could be useful, e.g., in amusement parks and other real-world game areas as discussed above, but also in the field of virtual gaming.

[0084] In virtual gaming, there often exist scenarios in which interactions between virtual players must be coordinated to achieve some optimal path based on certain parameters. While these parameters may include those discussed above (e.g., proximity and clustering), the system 102 could be easily adapted to include other parameters that would make the underlying routing more applicable to games within virtual worlds such as, e.g., Second Life or other digital games. The game-dynamics engine 110 could also be used in an online virtual dating world, where different avatars are routed to meet others based on similar parameters currently used in routing teams around a real-world game area. The routing algorithm can thereby be transitioned from the real to the virtual worlds.

[0085] It is to be understood that although the invention has been described above in terms of particular embodiments, the foregoing embodiments are provided as illustrative only, and do not limit or define the scope of the invention. Various other embodiments, including but not limited to the following, are also within the scope of the claims. For example, elements and components described herein may be further divided into additional components or joined together to form fewer components for performing the same functions.

[0086] The game development platform 104, the game dynamics engine 110, and other processes described herein may be implemented in software, hardware, firmware, or any combination thereof. The processes are preferably implemented in one or more computer programs executing on a programmable computer including a processor, a storage medium readable by the processor (including, e.g., volatile and non-volatile memory and/or storage elements), and input and output devices. Each computer program can be a set of instructions (program

code) in a code module resident in the random access memory of the computer. Until required by the computer, the set of instructions may be stored in another computer memory (e.g., in a hard disk drive, or in a removable memory such as an optical disk, external hard drive, memory card, or flash drive) or stored on another computer system and downloaded via the Internet or other network.

[0087] Each computer program may be implemented in any programming language including, e.g., an assembly language, a machine language, a high-level procedural programming language, or an object-oriented programming language. The programming language may, e.g., be a compiled or interpreted programming language.

[0088] Having described preferred embodiments of the present invention, it should be apparent that modifications can be made without departing from the spirit and scope of the invention.

[0089] Method claims set forth below having steps that are numbered or designated by letters should not be considered to be necessarily limited to the particular order in which the steps are recited.

[0090] What is claimed is:

CLAIMS

1. A computer-implemented method for generating and managing a customized interactive multiplayer location-based mobile game, comprising the steps of:

(a) receiving at a computer system content and high-level game logic from a game designer to be used in developing a location-based mobile game;

(b) generating a playable location-based mobile game based on the content and high-level game logic using the computer system;

(c) registering using the computer system a plurality of players desiring to play the multiplayer location-based mobile game using their personal mobile communications devices; and

(d) managing execution of the game for each player or group of players by routing the player or group of players to a plurality of locations to perform a location-based task at each location, and interacting with the player or group of players after each of the location-based tasks has been performed, wherein managing execution of the game and interacting with the player or group of players comprises using the computer system to communicate with a personal mobile communications device of the player or group of players.

2. The method of claim 1 wherein routing the player or group of players comprises providing the player or group of players with a clue hinting at the next location to which the player or group of players must travel to complete a location-based task.

3. The method of claim 1 wherein interacting with the player or group of players comprises determining whether a location-based task has been successfully completed by the player or group of players.

4. The method of claim 1 wherein each location-based task comprises solving a riddle relating to the location or capturing an image at the location and transmitting the image to the computer system.

5. The method of claim 1 further comprising calculating a score for each player or group of players based on the player's or group of players' completion of each location-based task.

6. The method of claim 1 further comprising providing a hint to the player or group of players to assist the player or group of players in completing the location-based task.

7. The method of claim 1 further comprising running a simulation of the game before execution of the game and providing suggested modifications to the game designer based on an analysis of the game with historical data on execution of other prior games.

8. The method of claim 1 wherein at least some of the personal mobile communications devices comprise an SMS compatible cellular telephone.

9. The method of claim 1 wherein at least some of the personal mobile communications devices comprise a smart phone, and wherein the game is deployed on a mobile web portal accessed by the smart phone or using an application downloaded on the smart phone.

10. The method of claim 1 wherein routing each player or group of players to the plurality of locations comprises dynamically routing different players or groups of players through different paths through the locations.

11. The method of the claim 10 wherein the different paths are selected based on the distance between the locations or the relative importance of the locations, or to avoid potential clustering of players or groups of players at locations.

12. The method of claim 1 further comprising providing a final location-based task to the first player or group of players to reach a threshold score, and providing the final location-based task to other players or groups of players after a time delay based on their respective scores.

13. The method of claim 1 further comprising determining a location to route a player or group of players to based on whether a location-based task has

been successfully performed.

14. The method of claim 1 further comprising providing each of the players or groups of players all location-based tasks of the game at one time.

15. A computer system for generating and managing a customized interactive multiplayer location-based mobile game, comprising:

a game development platform configured to receive content and high-level game logic from a game designer to be used in developing a location-based mobile game, and to generate a playable location-based mobile game based on the content and high-level game logic using the computer system; and

a game dynamics engine configured to register a plurality of players desiring to play the multiplayer location-based mobile game using their personal mobile communications devices; and to manage execution of the game for each player or group of players by routing the player or group of players to a plurality of locations to perform a location-based task at each location, and to interact with the player or group of players after each of the location-based tasks has been performed, wherein managing execution of the game and interacting with the player or group of players comprises using the computer system to communicate with a personal mobile communications device of the player or group of players.

16. The computer system of claim 15 wherein the game dynamics engine is configured to route the player or group of players by providing the player or group of players with a clue hinting at the next location to which the player or group of players must travel to complete a location-based task.

17. The computer system of claim 15 wherein the game dynamics engine is configured to determine whether a location-based task has been successfully completed by the player or group of players.

18. The computer system of claim 15 wherein the game dynamics engine is configured to calculate a score for each player or group of players based on the player's or group of players' completion of each location-based task.

19. The computer system of claim 15 wherein the game dynamics engine

is configured to provide a hint to the player or group of players to assist the player or group of players in completing the location-based task.

20. The computer system of claim 15 wherein the game development platform is configured to run a simulation of the game before execution of the game and providing suggested modifications to the game designer based on an analysis of the game with historical data on execution of other prior games.

21. The computer system of claim 15 wherein the game dynamics engine is configured to dynamically route different players or groups of players through different paths through the locations.

22. The computer system of claim 21 wherein the different paths are selected based on the distance between the locations or the relative importance of the locations, or to avoid potential clustering of players or groups of players at locations.

23. The computer system of claim 15 wherein the game dynamics engine is configured to provide a final location-based task to the first player or group of players to reach a threshold score, and provide the final location-based task to other players or groups of players after a time delay based on their respective scores.

24. The computer system of claim 15 wherein the game dynamics engine is configured to determine a location to route a player or group of players to based on whether a location-based task has been successfully performed.

25. The computer system of claim 15 wherein the game dynamics engine is configured to provide each of the players or groups of players all location-based tasks of the game at one time.

26. A computer program product residing on a computer readable medium having a plurality of instructions stored thereon which, when executed by a computer processor, cause that processor to:

(a) receive content and high-level game logic from a game designer to be used in developing a location-based mobile game;

(b) generate a playable location-based mobile game based on the content and high-level game logic;

(c) register a plurality of players desiring to play the multiplayer location-based mobile game using their personal mobile communications devices; and

(d) manage execution of the game for each player or group of players by routing the player or group of players to a plurality of locations to perform a location-based task at each location, and interact with the player or group of players after each of the location-based tasks has been performed, wherein managing execution of the game and interacting with the player or group of players comprises communicating with a personal mobile communications device of the player or group of players.

27. The computer program product of claim 26 further comprising instructions for providing the player or group of players with a clue hinting at the next location to which the player or group of players must travel to complete a location-based task.

28. The computer program product of claim 26 further comprising instructions for calculating a score for each player or group of players based on the player's or group of players' completion of each location-based task.

29. The computer program product of claim 26 further comprising instructions for providing a hint to the player or group of players to assist the player or group of players in completing the location-based task.

30. The computer program product of claim 26 further comprising instructions for running a simulation of the game before execution of the game and providing suggested modifications to the game designer based on an analysis of the game with historical data on execution of other prior games.

31. The computer program product of claim 26 further comprising instructions for dynamically routing different players or groups of players through different paths through the locations.

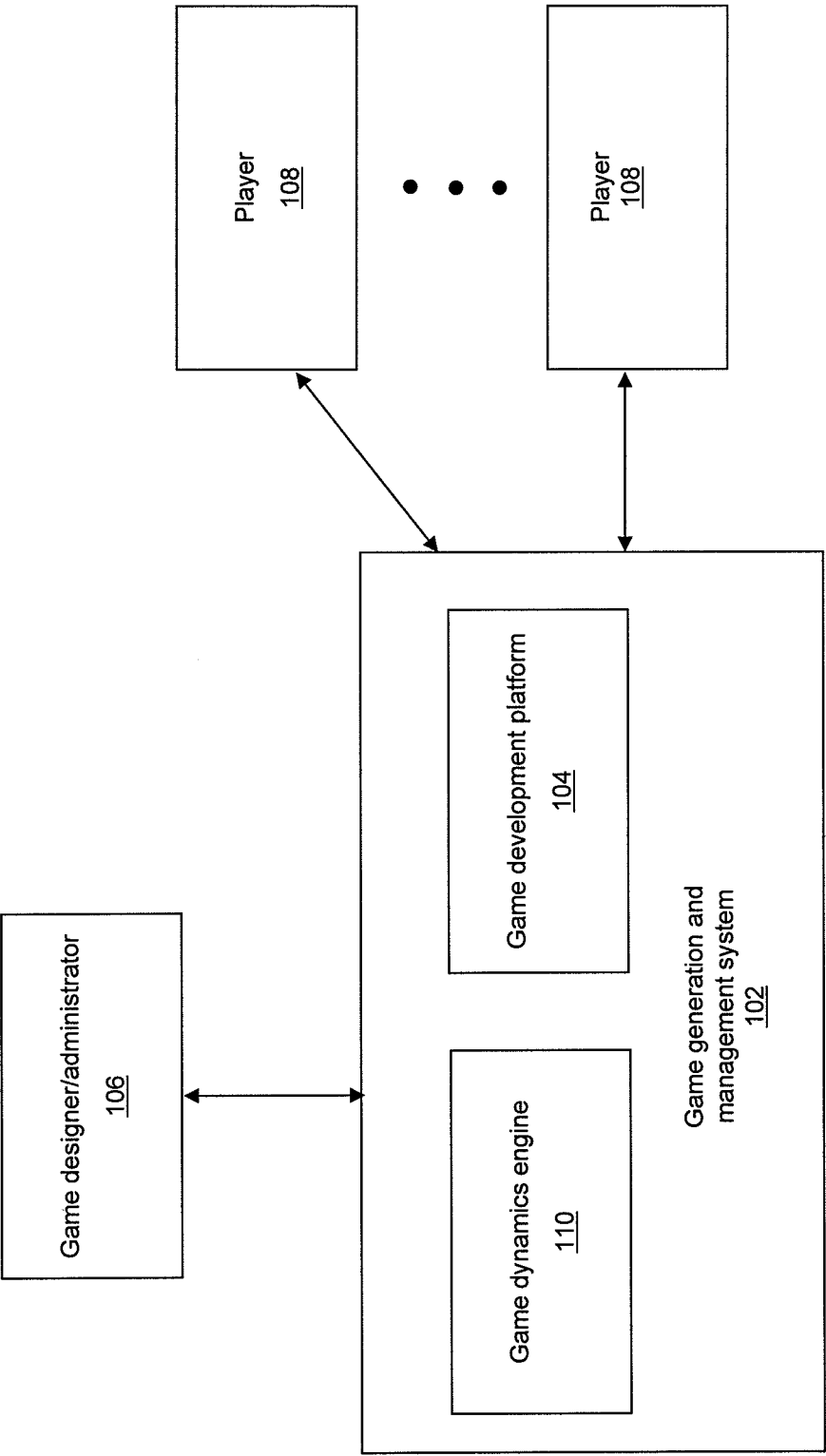
32. The computer program product of claim 31 wherein the different

paths are selected based on the distance between the locations or the relative importance of the locations, or to avoid potential clustering of players or groups of players at locations.

33. The computer program product of claim 26 further comprising instructions for providing a final location-based task to the first player or group of players to reach a threshold score, and providing the final location-based task to other players or groups of players after a time delay based on their respective scores.

34. The computer program product of claim 26 further comprising instructions for determining a location to route a player or group of players to based on whether a location-based task has been successfully performed.

35. The computer program product of claim 26 further comprising instructions for providing each of the players or groups of players all location-based tasks of the game at one time.



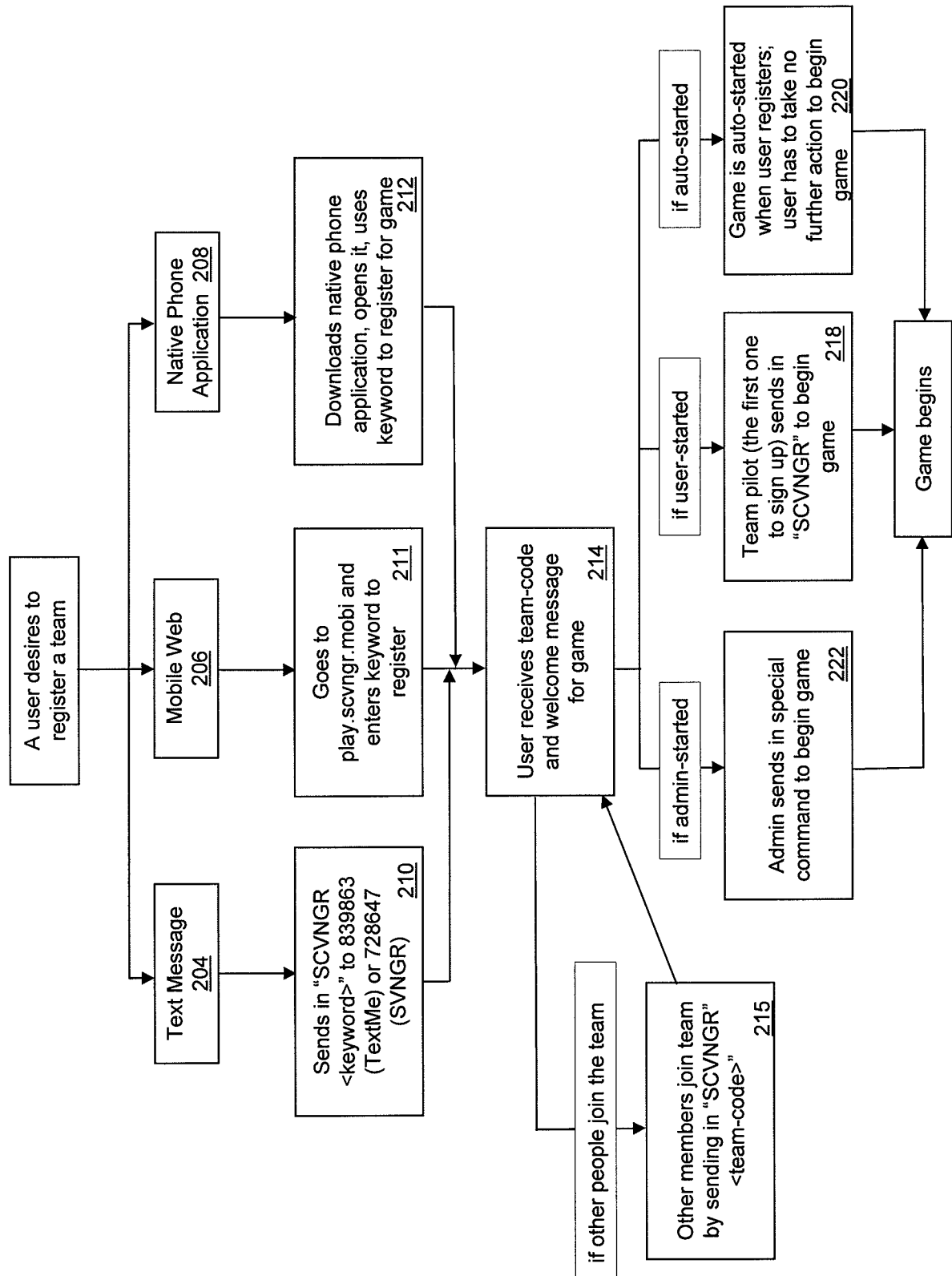


FIG. 2

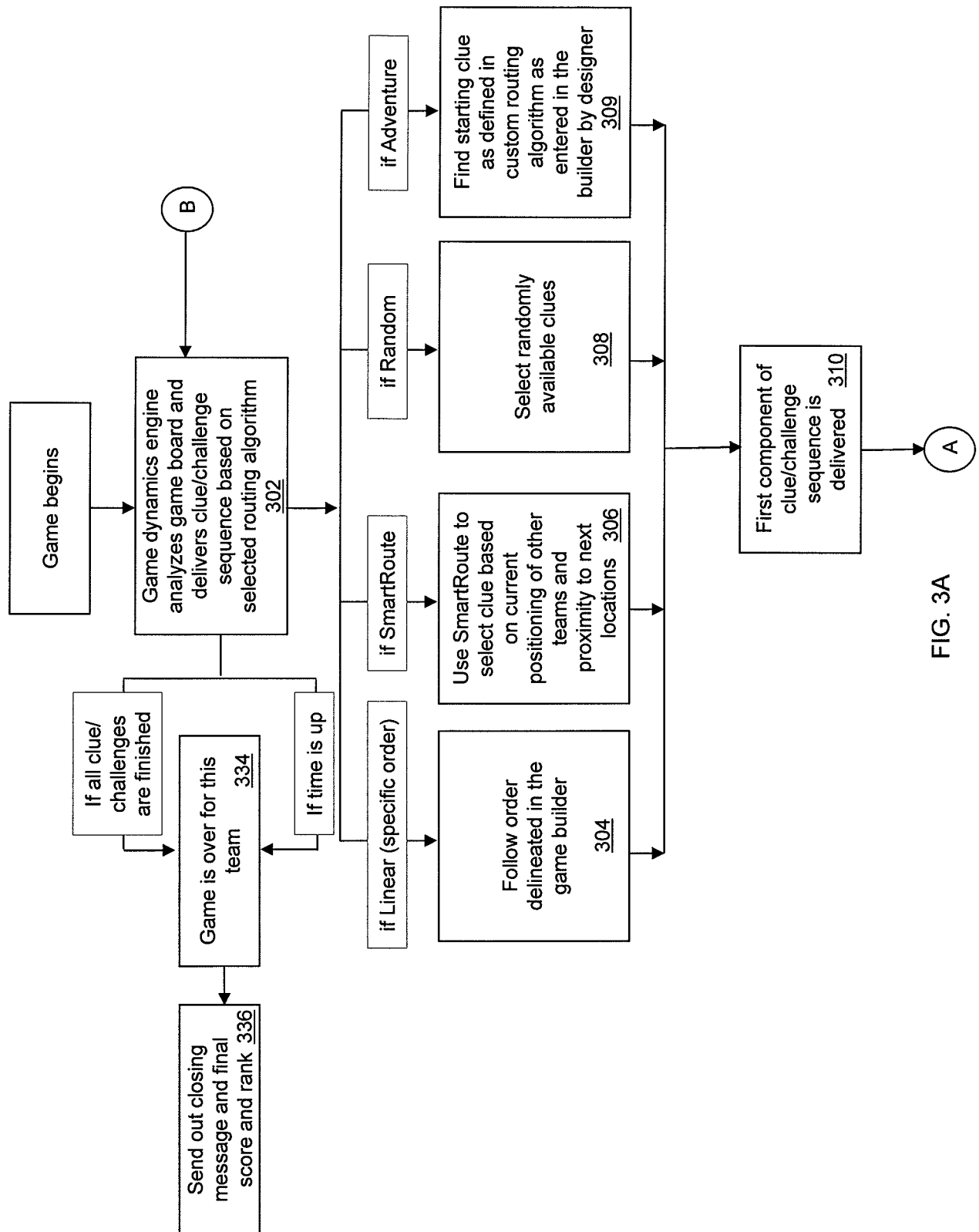


FIG. 3A

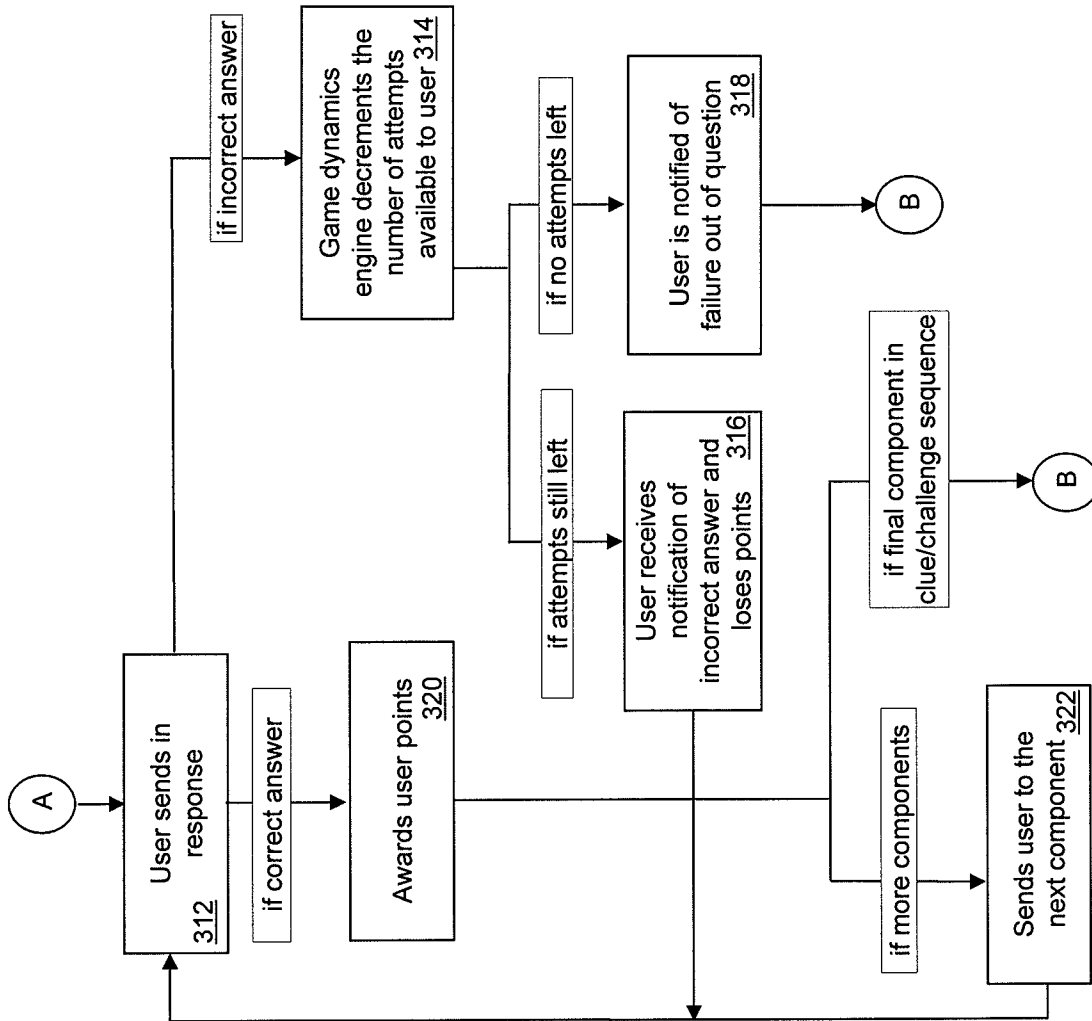


FIG. 3B