SYSTEM AND METHOD FOR DYNAMICALLY EVOLVING CONTENT ASSOCIATED WITH NON-PLAYER CHARACTERS

[001] The present invention relates to non-player characters of a game environment. More particularly, the present invention relates to a system and method for dynamically evolving parameterised content such as traits, characteristics, and relationships associated with them.

BACKGROUND OF THE INVENTION

[002] Interactive media, such as video games, simulations, and virtual worlds, have become increasingly popular in recent years.

[003] As these experiences have become more complex and sophisticated, there has been a growing need for artificial intelligence (AI) to be used to control the behaviour of non-player characters (NPCs).

[004] NPCs, or non-player characters, are characters in video games and other interactive media that are not controlled by the player. They may be anything from shopkeepers to enemies to allies that are used to populate the game world and to provide the player with challenges, assistance, and entertainment.

[005] More specifically, non-player characters are an important part of most video games. These non-player characters make the game world more believable and immersive, and they provide the player with a variety of challenges and interactions.

[006] In recent years, there has been a growing interest in using artificial intelligence (AI) to create more complex and believable non-player characters in video games and/or interactive media. However, these non-player characters are limited to simple or rudimentary tasks, such as giving out quests, shopkeeping, pathfinding and/or combat. Furthermore, the uses of AI on non-player characters are often based on rule-based systems or state machines. These systems may be effective for controlling simple non-player characters, but they quickly become complex and difficult to manage for more complex non-player characters.

Moreover, these systems may often lead to the non-player characters behaving in a repetitive and predictable manner.

[007] One such invention is disclosed in the Chinese Patent Application CN109200583 that provides a system and method involving controlling the behaviour of the non-player characters in an online game for updating their battle strategies and combat behaviour. This prior art involves using the player's behaviour and results as training samples. Once recorded data is captured, it is processed to be used for training non-player characters' behavioural models in order to improve their combat capabilities. More specifically, machine learning algorithms are employed to train these behaviour models. This allows for a more immersive gameplay experience for the player. However, this system does not extend to non-combat applications as well as does not operate during gameplay, nor from other non-player entities. Furthermore, this system does not process from more than one entity interacting with the non-player character at a time.

[008] Therefore, there is a need for a system and method that allows the non-player characters to be modelled as autonomous agents with their desires and motivations.

[009] Additionally, there is a need for a system and method that allows the non-player characters to use their traits and priorities to create a schedule for themselves, which outlines what they want to do and when they want to do it.

[0010] In a nutshell, a system and method that may overcome the above-discussed shortcomings by providing a novel decision mechanism for dynamically evolving pre-installed content such as traits, characteristics, and relationships associated with one or more non-player characters (NPC) in a game environment.

SUMMARY OF INVENTION

[0011] The present invention intends to provide a system and method for dynamically evolving, via updating, content associated with one or more non-player characters in a game environment. In particular, the non-player characters are to evolve, by being updated in real-time, i.e., during run-time of the game. More specifically, a non-player characters are intended to evolve in response to the game environment, which may include other non-player characters, player characters, in-game items, or the like.

[0012] The system of the present invention comprises at least one processor that operates modules that include a memory module that manages content for each non-player character by assigning content associated to them and facilitating storage of the content as their memory, an activity module that allows non-player characters to conduct one or more actions, an interacting module that allows at least one interaction between non-player characters, at least one interaction between non-player characters and one or more player characters and/or at least one interaction between the characters and items in the game environment, a computing module that computes one or more modifications to be made to the content associated with the non-player characters based on the interactions, and an update module that informs the memory module to modify the content associated with the non-player characters. The computing module determines outcomes from the computed modifications for the computed modifications to be selectively applied upon content associated with the non-player characters based on the outcomes.

[0013] Preferably, the modules of the system further include a map module to provide at least one map comprising one or more locations within the game environment, wherein the activity module further allows the non-player characters to move between locations based on their associated content, and interactions between non-player characters, between non-player characters and one or more player characters, and/or between the non-player characters and the items, are based on locations of the non-player characters.

[0014] Preferably, the modules of the system further include a trait module, interfaced with the update module and activity module, to manage and update content that relates to traits

of the non-player characters and their actions to be performed based on their locations within the game environment.

[0015] Preferably, the modules of the system further include a commitment module, interfaced with the update module and activity module, to manage and update the actions to be performed by non-player characters based on a scheduled timeline and/or specific conditions within the game environment.

[0016] Preferably, the modules of the system further include a priority module, interfaced with the update module and activity module, to manage and update prioritisation of the actions to be performed by the non-player characters.

[0017] Preferably, the modules of the system further include a relationship module, interfaced with the update module and activity module, to manage and update information related to the interactions between the non-player characters and between the non-player characters and the player characters.

[0018] Preferably, the modules of the system further include an analysing module that analyses the one or more modifications based on the content of each non-player character.

[0019] Preferably, regarding the system, the outcomes of the one or more modifications determined by the computing module includes any one or a combination of positive outcomes or negative outcomes.

[0020] Preferably, regarding the system, the activity module allows each non-player character to be implemented with the modifications related to the positive outcomes for them to tend towards the positive outcomes in a future state of the game environment, and/or to be implemented with the modifications related to the negative outcomes for them to tend away from the negative outcomes in the future state of the game environment.

[0021] Preferably, regarding the system, the memory module is adapted to facilitate storage of the modifications associated with the non-player characters prior to the non-player characters being updated with the modifications.

[0022] The present invention further intends to provide a method for dynamically evolving content associated with one or more non-player characters in a game environment,

comprising the steps of managing content for each non-player character assigning content associated with them, and facilitating storage of the content as their memory, by a memory module, allowing the non-player characters to conduct one or more actions, allowing at least one interaction between non-player characters, at least one interaction between non-player characters and one or more player characters, and/or at least one interaction between the characters and items in the game environment, computing one or more modifications to be made to the content associated with the non-player characters based on the interactions, by a computing module, and informing the memory module to modify the content associated with the non-player characters, by an update module. The computing module determines outcomes from the computed modifications for the computed modifications to be selectively applied upon content associated with the non-player characters based on the outcomes.

[0023] Preferably, the method further comprises the step of providing at least one map each comprising one or more locations within the game environment, by a map module, wherein the activity module further allows the non-player characters to move between locations based on their associated content, and interactions between non-player characters, between non-player characters and one or more player characters, and/or between the non-player characters and the items, are based on locations of the non-player characters.

[0024] Preferably, the method further comprises the step of managing and updating, via a trait module, which is interfaced with the update module and activity module, traits of the non-player characters based on the locations and their actions to be performed based on their locations within the game environment.

[0025] Preferably, the method further comprises the step of managing and updating, via a commitment module, which is interfaced with the update module and activity module, actions that are to be performed by non-player characters based on a scheduled timeline and/or specific conditions within the game environment.

[0026] Preferably, the method further comprises the step of managing and updating, via a priority module, which is interfaced with the update module and activity module, prioritisations of the actions to be performed by the non-player characters.

[0027] Preferably, the method further comprises the step of managing and updating, via a relationship module, which is interfaced with the update module and activity module, the

content of the non-player characters based on information related to the interactions between the non-player characters, and interactions between the non-player characters and the player characters.

[0028] Preferably, the method further comprises the step of analysing, via an analysing module, the one or more modifications based on the content of the non-player characters, via a comparison of tags of the content.

[0029] Preferably, regarding the method, the outcomes of the one or more modifications determined by the computing module includes any one or a combination of positive outcomes or negative outcomes.

[0030] Preferably, the method further comprises the step of allowing, by the activity module, each non-player character to be implemented with the modifications related to the positive outcomes for them to tend towards the positive outcomes in a future state of the game environment, and/or to be implemented with the modifications related to the negative outcomes for them to tend away from the negative outcomes in the future state of the game environment.

[0031] Preferably, the method further comprises the step of facilitating, by the memory module, storage of the modifications associated with the non-player characters prior to the non-player characters being updated with the modifications.

[0032] This together with the other aspects of the present invention along with the various features of novelty that characterize the present disclosure is pointed out with particularity in claims annexed hereto and forms a part of the present invention.

[0033] For a better understanding of the present disclosure, its operating advantages, and the specified objective attained by its uses, reference should be made to the accompanying descriptive matter in which there are illustrated exemplary embodiments of the present invention.

[0034] One skilled in the art will readily appreciate that the invention is well adapted to carry out the objects and obtain the ends and advantages mentioned, as well as those inherent

therein. The embodiments described herein are not intended as limitations on the scope of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

[0035] To facilitate an understanding of the invention, there is illustrated in the accompanying drawings the preferred embodiments from an inspection of which when considered in connection with the following description, the invention, its construction and operation and many of its advantages would be readily understood and appreciated.

[0036] FIG. 1 illustrates a block diagram of a system for dynamically evolving content such as traits, characteristics, and relationships associated with one or more non-player characters in a game environment, which may include interactions between the characters and items in the game environment, according to various embodiments of the present invention.

[0037] FIG. 2 illustrates a block diagram of an activity module, according to various embodiments of the present invention.

[0038] FIG. 3 illustrates a block diagram of an interacting module, according to various embodiments of the present invention.

[0039] FIG. 4 illustrates a block diagram of a computing module, according to various embodiments of the present invention.

[0040] FIG. 5 illustrates a block diagram of content associated with non-player characters, according to various embodiments of the present invention.

[0041] FIG. 6 illustrates a flowchart of a method for dynamically evolving content such as traits, characteristics, and relationships associated with one or more non-player characters in a game environment, according to various embodiments of the present invention.

[0042] FIG. 7 illustrates a block diagram describing the evaluation and computational modifications for changes in the activities of the non-player character.

DETAILED DESCRIPTION OF THE INVENTION

[0043] The present invention relates to a system and method for dynamically evolving pre-installed content such as traits, characteristics, and relationships associated with the non-player characters, preferably for an online and/or offline game environment.

[0044] The exemplary embodiments described herein detail for illustrative purposes are subjected to many variations. It should be emphasized, however, that the present invention is not limited to as disclosed.

[0045] It is understood that various omissions and substitutions of equivalents are contemplated as circumstances may suggest or render expedient, but these are intended to cover the application or implementation without departing from the scope of the present invention.

[0046] From hereon, the following terms have the meanings indicated below.

[0047] The terms “a” and “an” herein do not denote a limitation of quantity, but rather denote the presence of at least one of the referenced items.

[0048] The terms “having”, “comprising”, “including”, and “variations” thereof signify the presence of a component.

[0049] The inventive aspects of the invention along with various components and engineering involved will now be explained with reference to **FIGS 1 – 7** herein.

[0050] **FIG. 1** illustrates the block diagram of the system **100** for dynamically evolving content associated with one or more non-player characters in a game environment. It is to be noted that the content may initially be pre-installed.

[0051] Referring to **FIG. 1**, the system **100** includes a processor that operates modules that include a memory module **160**, a trait module **105**, a map module **110**, an activity module **120**, an interacting module **130**, a computing module **140**, and an updating module **150**. It is

to be noted that these modules may substantially be interfaced with each other. The system **100** is to be explained in respect of at least one non-player character. However, this should not be construed as a limitation of the invention.

[0052] Referring to **FIG. 1**, the memory module **160** is to manage content for each non-player character by assigning content associated with them. Furthermore, the memory module **160** to facilitate storage of the said content as their memory within a database, which may be at least one data storage device. With reference to **FIG. 5**, the content may include, but shall not be limited to traits, priorities, relationships, schedule, and other characteristics or parameters.

[0053] Referring to **FIG. 1**, the map module **110** is configured to provide a map that includes one or more locations of the game environment. Data of the map may be pre-generated based on assets stored in the database and/or produced during run-time of the game. More specifically, the map is a collection of locations in the game environment with pre-determined rules regarding movement between said locations.

[0054] Referring to **FIG. 1**, the activity module **120** is configured for allowing the non-player characters to conduct one or more actions based on the content associated with them.

[0055] Referring to **FIG. 1**, the interacting module **130** is configured for allowing at least one interaction between non-player characters, at least one interaction between non-player characters and one or more player characters, and/or at least one interaction between the characters and items in the game environment.

[0056]Referring to **FIG. 1**, the trait module **105** manages traits of at least one non-player character based on their locations. The trait module **105** may further manage and determine the one or more activities performed by the said non-player character in the game environment. In one further embodiment, the trait module **105** may also be interfaced with the map module **110** for traits of one or more non-player characters to be managed and determined based on locations within the game environment and their corresponding items.

[0057]Referring to **FIG. 1**, the computing module **140** is to compute one or more modifications to be made to the content of the non-player characters based on their

interactions. Preferably, from the computed modifications, the computing module **140** may further determine one or more outcomes that relate to a future state of the non-player characters in the game environment. Based on the outcomes, the computing module **140** may select some of the computed modifications for them to be selectively applied upon content associated with the non-player characters. In particular, the content may include relationships of non-player characters with other characters that they had interacted with. It is to be noted that the computing module **140** is further involved in performing relevant computations for updating information regarding map locations and their performable activities.

[0058] Referring to **FIG. 1**, the updating module **150** is configured to update the selected modifications in the content associated with the non-player characters by informing the memory module **160** to apply the said modifications upon the non-player characters.

[0059] Referring to **FIG. 1**, it is to be noted that the activity module **120** may further allow the non-player characters to move between locations, which had been provided by the map module **110**, based on the associated content of the non-player characters.

[0060] Referring to **FIG. 1**, it is to be noted that the interacting module **130** may further allow interactions between non-player characters, between non-player characters and one or more player characters, and/or between the non-player characters and the items, which are based on locations of the non-player characters.

[0061] FIG. 2 illustrates a block diagram of the activity module **120**, according to various embodiments of the present invention. As shown in **FIG. 2**, the activity module **120** comprises a commitment module **122** and a priority module **124** as its sub-modules.

[0062] Referring to **FIG. 2**, the commitment module **122** of the activity module **120** manages the one or more activities to be performed by at least one non-player character based on a scheduled timeline within the game environment.

[0063] Referring to **FIG. 2**, the priority module **124** of the activity module **120** manages the prioritised activities to be performed by one non-player character as the said non-player character is initialised in the game environment. More specifically, the priority module **124**

may have a higher maximum priority in in creating commitments than interactions between characters and/or actions performed by the characters. Priorities may also be conditional, enforcing commitments based on meeting several preconditions. Priorities may also involve proposing certain activities to characters that meet specific conditions.

[0064] FIG. 3 illustrates a block diagram of the interacting module **130**, according to various embodiments of the present invention. As shown in **FIG. 3**, the interacting module **130** comprises a relationship module **132** as its sub-modules.

[0065] Referring to **FIG. 3**, the relationship module **132** of the interacting module **130** manages information related to the interaction interactions between the non-player characters, between the non-player characters and the player characters, and/or between the characters and items in the game environment. It is to be noted that the relationship module **132** may create a relationship based on the said interactions if none existed before.

[0066] FIG. 4 illustrates a block diagram of the computing module **140**, according to various embodiments of the present invention. As shown in **FIG. 4**, the computing module **140** comprises an analysing module **142**, a determining module **144**, and a selecting module **146**, as its sub-modules.

[0067] Referring to **FIG. 4**, the analysing module **142** of the computing module **140** is adapted to analyse the one or more modifications. This may be done based on the content associated with a non-player character being analysed and to be modified.

[0068] Referring to **FIG. 4**, the determining module **144** of the computing module **140** is to determine one or more outcomes from the one or more modifications. In particular, the outcomes of the modifications are at least any one or a combination of positive outcomes or negative outcomes.

[0069] In particular, it is to be stated that the determining module **144** of the computing module **140** evaluates the outcomes. In one embodiment, this evaluation may be done in a binary manner. In one alternative embodiment, this evaluation may be done in a non-binary manner, more specifically, a sentiment calculation based on one or more integer values.

[0070] Referring to **FIG. 4**, the selecting module **146** of the computing module **140** is to select the computed modifications based on the outcomes. In particular, the selected one or more modifications, such as interaction information associated with at least one non-player character, are stored in the database by the memory module **160**. Furthermore, these selected modifications may be used to inform future interactions as well as aid in dialogue generation.

[0071] With this, it is to be stated that the computing module **140** shall evaluate possible future actions or states of the non-player character as outcomes, and select one or more computed modifications to be applied upon the non-player character, by taking into account data that can be updated during run-time of the game, to allow behaviour of the non-player character to tend towards the outcomes.

[0072] FIG. 6 illustrates the first flowchart describing a method **60** for dynamically evolving content such as traits, characteristics, and relationships associated with one or more non-player characters in a game environment, according to various embodiments of the present invention. It is noted that the steps described in these flowcharts are not to be interpreted as non-limiting, and minor modifications to the steps (e.g. additions, omissions, or swaps) are permissible by a skilled person without substantial deviation from as described.

[0073] The steps of method **60** may begin with step **61**, which involves assigning a non-player character with content associated with them, by the memory module **160**. The memory module may further perform the step of facilitating storage of these content in the database.

[0074] Following step **61** is step **62**, which involves providing a map, by the map module **110**, comprising one or more locations within the game environment.

[0075] Following step **62** is step **63**, which involves allowing the non-player characters to utilise established traits and relationships in order to initiate activities, by the activity module **120**. The non-player characters may further be allowed to establish commitments for the immediate future in the game environment.

[0076] Following step **63** is step **64**, which involves allowing, via the activity module **120**, the non-player characters to move between the one or more locations. Preferably, the non-player characters move between locations based on content that includes their established commitments.

[0077] Following step **64** is step **65**, which involves allowing, via an interacting module **130**, at least one non-player character to interact with one or more non-player characters, player characters, and/or items at the one or more locations of the game environment.

[0078] Following step **65** is step **66**, which involves computing, via a computing module **140**, one or more modifications in the content associated with at least one non-player character.

[0079] Finally, following step **66** is step **67**, which involves updating, via an updating module **150**, content associated with at least one non-player character with selected modifications for these modifications to be applied upon the non-player character and/or to be stored in the database by the memory module **160**.

[0080] **FIG. 7** illustrates a second flowchart describing a method **70** that is related to the evaluation and computational modifications for changes in the activities of the non-player character, according to various embodiments of the present invention. It is noted that the steps described in these flowcharts are not to be interpreted as non-limiting, and minor modifications to the steps (e.g. additions, omissions, or swaps) are permissible by a skilled person without substantial deviation from as described. Furthermore, while specific features that were previously described in **FIGS 1 – 6** may not be fully described, it is to be understood that they are to be implicated.

[0081] The steps of method **70** may begin with step **71**, which involves starting and performing activities, by at least one non-player character, via a random event in the game environment or via their commitments.

[0082] Following step **71** is step **72**, which involves analysing, by the computing module **140**, the present relationships of each non-player character with the other non-player characters or player characters. This may be done in a continuous manner. Furthermore, as

analysis is being done, the interactions between non-player characters and between non-player characters and one or more player characters may affect the parameters of the non-player characters.

[0083] Following step **72** is step **73**, which is a decision step whereby it is determined whether or not the relationship parameter of one non-player character with other characters, which may be non-player characters and/or player characters, exceeded a certain threshold. This event may be picked up by the computing module **140** during its analysis. Should this not be the case, step **73** proceeds to step **74**. Should this be the case, step **74** proceeds to step **75**.

[0084] In step **74**, since it was determined that the relationship parameter of one non-player character with other characters, which may be non-player characters and/or player characters, did not exceed a certain threshold. Hence, the step of feeding each relevant trait and relationship parameter into a first modification function of the computing module **140** for them to be updated is performed. In particular, since the relationship parameter is not exceeded, the first modification function shall only modify the parameters according to their interactions with the non-player characters and/or player characters.

[0085]In step **75**, since it was determined that the relationship parameter of one non-player character with other characters, which may be non-player characters and/or player characters, exceeded a certain threshold. Hence, the step of feeding each relevant trait and relationship parameter into a second modification function of the computing module **140** for them to be updated is performed. In particular, since the relationship parameter is exceeded, the second modification function triggers a special event or outcome in the game environment whereby parameters of the non-player character shall be modified for them to have a tendency to interact more with the characters that caused the relationship parameter to be exceeded.

[0086] Following step **74** or step **75** is step **76**, which involves storing modifications in the relationship and traits, and updating the non-player character for them to have an updated trait parameter and relationship parameter.

[0087] Finally, following step **76** is step **77**, which involves implementing the updated trait parameter and relationship parameter upon the non-player character for their activities to be performed in the game environment to be updated.

[0088] With reference to **FIGS 1 – 7**, it is to be noted that the trait module **105**, the commitment module **122**, the priority module **124**, and the relationship module **132** are to be updated when the activity module **120** allows the non-player character to perform their activities in the game environment.

[0089] With reference to **FIGS 1 – 7**, it is to be noted that commitments of the non-player character may establish their tendency to perform certain activities in various or specific locations of the map, in the future state of the game environment.

[0090] With reference to **FIGS 1 – 7**, it is to be noted that each non-player character has a random chance to interact with other characters, which includes other non-player characters and one or more player characters, in the same map area.

[0091] With reference to **FIGS 1 – 7**, it is to be noted that a trait of a non-player character refers to their relationship with a concept that determines the qualities and attributes of the non-player character in the game environment. As such, a trait of a non-player character may be stated to be composed of two parts. The first part of a trait relates to an encoded concept that in which the non-player character adheres thereto for them to carry out activities in the game environment. The second part of the trait may relate to a modifier value that is quantitative information associated with the first part of the trait.

[0092] In a preferred embodiment of the present invention, the means of providing encoded concepts for establishing the first part of one trait of the non-player characters in the game environment is preferably done by provision of word embeddings, such as the GloVe vector representations, to the game engine so that they may be interpreted accordingly by the non-player characters in the game environment. These word embeddings may further enable each non-player character to have a degree of flexibility while interacting with each character and the game environment, since these embeddings allow for similarity comparison of encoded concepts, via cosine similarity or vector distance.

[0093] In a preferred embodiment of the present invention, the second part of one trait of the non-player character in the game environment, which is the modifier value, determines the strength of the relationship between the non-player character and the trait, which could either be a positive or negative relationship.

[0094] With reference to **FIGS 1 – 7**, it is to be noted that the activity module **120** is configured to allow at least one non-player character to implement the one or more modifications with the positive outcomes in future gameplay.

[0095] More specifically, positive outcomes may result in characters implementing deviations to their routine or otherwise pre-establishing interactions in line with interests they may have or create interactions with more than one other character at a time.

[0096] With reference to **FIGS 1 – 7**, it is to be noted that the activity module **120** is configured to allow at least one non-player character to avoid the one or more modifications with the negative outcomes in the future gameplay.

[0097] More specifically, negative outcomes may result in characters implementing deviations to their routine or otherwise pre-establishing interactions in line with disinterests they may have or avoid interactions with more than one other character at a time.

[0098] Additionally, certain outcomes may also be used to trigger further development such as the development of romantic connections between characters or triggering a creation of additional characters to produce an ever-evolving game environment.

[0099] Advantageously, as the present invention uses concept encoding with similarity comparisons, the non-player characters within the game environment may react to almost any situation in a non-linear manner.

[00100] Advantageously as well, the present invention may seamlessly ensure compatibility between parameters of the non-player character and their activities in the game environment. More specifically, the present invention may ensure that updated parameters of the non-player characters are compatible before their parameters, such as their commitments, are

properly updated for them to perform their activities in the game environment. With this, a user controlling a player character may receive a natural and engaging experience with the non-player characters of the game environment whereby new and user-based content may be generated as the user interacts with the game environment.

[00101] Advantageously as well, the present invention may facilitate relationships between the non-player characters and a number of nebulous concepts. More specifically, the present invention performs an evaluation of the non-player character in the game environment based on their interactions, and in conjunction with the implementation of concept encoding, the non-player character may respond appropriately towards these interactions and may have changes in their activities in the future state of the game environment.

[00102] Advantageously as well, the present invention enables the non-player characters to be modelled as autonomous entities in the game environment with their desires and motivations. Further, each non-player character may have (i) a set of traits, which represent their personality and characteristics, and (ii) a set of priorities, which represent what they want to achieve in a future state of the game environment.

[00103] Advantageously as well, the present invention enables non-player characters in a game environment to create a schedule for themselves based on their traits and priorities, which shall outline what they want to do and when they want to do it in the future state of the game environment.

[00104] Advantageously as well, the present invention enables non-player characters in a game environment to act intelligently and have a nuanced decision-making process. This results in non-player characters that are more believable and realistic. This will provide a user, who controls a player character, with more engaging and challenging interactions.

[00105] Advantageously as well, the present invention enables non-player characters to be adaptable in a wide range of game environment scenarios, such as games that involve life-simulation or games with a focus on character interaction that have a dynamic cast of non-player characters.

[00106] Advantageously as well, the present invention enables a user, who controls a player character, to interact with the non-player characters and the game environment to build personalised relationships with said non-player characters, as stored information regarding prior interactions between the non-player characters and the player character may be used t cohesive generated a dynamic and ever-evolving game environment for greater immersion.

[00107] The system and method, as described in the disclosed teachings or any of its components, may be embodied in the form of a computer-implemented method. Typical examples of a computer included in the computer-implemented method include a general-purpose computer, a personal digital assistant, a cell phone, a programmed microprocessor, a microcontroller, a peripheral integrated circuit element, and other devices or arrangements of devices that are capable of implementing the steps that constitute the method of the disclosed teachings.

[00108] The system and method of the present invention may be implemented using a general-purpose computer, such a general-purpose computer can include an input device such as a keyboard, a mouse, a trackball, a touchpad, a microphone, or a camera. The computer can include a display unit, such as a projector, a monitor, or a touch screen. A touch screen can also be used as an input device. Specifically, the computer may comprise a hardware processor, wherein the processor is connected to a communication bus.

[00109] The system and method of the present invention may further include a memory, which may be any one or a combination of non-volatile memory or volatile memory such as Random Access Memory (RAM), Read Only Memory (ROM), or the like.

[00110] The system and method of the present invention may further include a non-transitory computer-readable medium such as a storage device, which may be a hard disk drive or a removable storage drive such as a flash drive, optical disk drive, or the like. The storage device can also comprise other similar means for loading computer programs or other instructions into the computer in which the system and method of the present invention are implemented thereon.

[00111] The system and method of the present invention may further include a communication device to communicate with a remote computer through a network. The communication device may include a wireless communication port, a data cable connecting the computer of the computer-implemented method with the network, or the like. The network may be a Local Area Network (LAN) or a Wide Area Network (WAN) such as the Internet, or the like. The remote computer that is connected to the network can be a general-purpose computer, a server, a personal digital assistant, or the like. Furthermore, the computer in which the system and method of the present invention are implemented thereon may access information from the remote computer through the network.

[00112] The sets of instructions stored on the computer-readable medium include various commands that instruct the computer in which the system and method of the present invention are implemented thereon to perform specific tasks, such as the steps that comprise the methods of the disclosed teachings.

[00113] The sets of instructions may be embodied in the form of a transitory medium, such as a software program.

[00114] The software program may be in various forms such as method software or application software. Furthermore, the software program might be in the form of a collection of separate programs, a program module with a larger program or a portion of a program module. Moreover, the software program may be configured to accept additional programs or data files to expand or enable further functionalities in the future.

[00115] The software program or programs may be provided as a computer program product, such as in the form of a computer-readable medium with the program or programs containing the set of instructions embodied therein. The processing of input data by the computer of the computer-implemented method may be in response to user commands or in response to the results of previous processing or in response to a request made by another computer.

[00116] The foregoing descriptions of specific embodiments of the present invention have been presented for purposes of description. They are not intended to be exhaustive or to limit

the present invention to the precise forms disclosed, and obviously, many modifications and variations are possible in light of the above teaching.

[00117] Further, the embodiments were chosen and described in order to best explain the principles of the present invention and its practical application, and thereby enable others skilled in the art to best utilize the present invention and various embodiments with various modifications as are suited to the particular use contemplated.

[00118] It is understood that various omissions and substitutions of equivalents are contemplated as circumstances may suggest or render expedient, but such omissions and substitutions are intended to cover the application or implementation without departing from the spirit or scope of the present invention.