TWO-DIMENSIONAL IMAGE DISPLAY METHOD AND APPARATUS FOR VIRTUAL OBJECT, AND DEVICE AND STORAGE MEDIUM

[0001] This application claims priority to Chinese Patent Application No. 202011140750.0, filed on October 22, 2020 and entitled "TWO-DIMENSIONAL IMAGE DISPLAY METHOD AND APPARATUS FOR VIRTUAL OBJECT, AND DEVICE AND STORAGE MEDIUM", which is incorporated herein by reference in its entirety.

FIELD

[0002] Embodiments of this application relate to the field of virtual scenes, and in particular, to a method and apparatus for displaying a two-dimensional figure of a virtual object, a device, and a storage medium.

BACKGROUND

[0003] In a virtual scene-based game application, a two-dimensional figure of a virtual object controlled by a user generally needs to be displayed.

[0004] In the conventional art, to improve a display effect of the two-dimensional figure of the virtual object, a two-dimensional figure image of the virtual object may be obtained by photographing a three-dimensional figure of the virtual object. For example, when setting a two-dimensional avatar of the virtual object, the user opens a three-dimensional figure display interface of a game role in an application program and then clicks on a control of "set as an avatar" in the three-dimensional figure display interface, so that an image of a region (for example, a head region) in the three-dimensional figure display interface is captured as an avatar of the game role, and the two-dimensional image is displayed at a position where the avatar needs to be displayed.

[0005] However, in the foregoing solution, the user needs to manually trigger to set the two-dimensional figure image of the virtual object. As a result, the two-dimensional figure image set by the user does not match the three-dimensional figure of the virtual object in many cases, resulting in a poor display effect of the two-dimensional figure of the virtual object.

SUMMARY

[0006] A method and apparatus for displaying a two-dimensional figure of a virtual object, a device, and a storage medium are provided according to embodiments of this application, which can improve a display effect of the two-dimensional figure of the virtual object. The technical solutions are described as follows.

[0007] According to an aspect, a method for displaying a two-dimensional figure of a virtual object is provided, which is performed by a computer device. The method includes:

displaying a target interface including a two-dimensional figure display position of the virtual object, where the virtual object is an object displayed in a virtual scene;

displaying a two-dimensional figure image of the virtual object at the two-dimensional figure display position; and

displaying, in a case that a two-dimensional figure updating condition is met, an updated two-dimensional figure image of the virtual object at the two-dimensional figure display position, where the updated two-dimensional figure image corresponds to a three-dimensional figure model of the virtual object.

[0008] According to an aspect, a method for displaying a two-dimensional figure of a virtual object is provided, which is performed by a computer device. The method includes:

loading a three-dimensional figure model of the virtual object in a case that a two-dimensional figure updating condition is met, where the virtual object is an object displayed in a virtual scene;

performing two-dimensional rendering on the three-dimensional figure model to obtain a two-dimensional figure image of the virtual object; and

displaying, in a case that a target interface including a two-dimensional figure display position of the virtual object is displayed, the two-dimensional figure image at the two-dimensional figure display position.

[0009] According to an aspect, a method for displaying a two-dimensional figure of a virtual object is provided, which is performed by a computer device. The method includes:

displaying an application interface of an application program corresponding to a virtual scene, wherein the application interface includes a two-dimensional figure display position;

displaying a two-dimensional figure image of the virtual object at the two-dimensional figure display position, wherein the virtual object is an object displayed in the virtual scene;

changing a figure of the virtual object in a case that an operation for changing the figure of the virtual object is received; and

displaying an updated two-dimensional figure image of the virtual object at the two-dimensional figure display position, where the updated two-dimensional figure image corresponds to a changed figure of the virtual object.

[0010] According to another aspect, an apparatus for displaying a two-dimensional figure of a virtual object is provided, which includes: an interface display module and an image display module. The interface display module is configured to display a target interface including a two-dimensional figure display position of the virtual object, where the virtual object is an object displayed in a virtual scene. The image display module is configured to display a two-dimensional figure image of the virtual object at the two-dimensional figure display position. The image display module is further configured to display, in a case that a two-dimensional figure updating condition is met, an updated two-dimensional figure image of the virtual object at the two-dimensional figure display position, where the updated two-dimensional figure image corresponds to a three-dimensional figure model of the virtual object.

[0011] In a possible implementation, the image display module includes: a model loading sub-module, a rendering sub-module and an image display sub-module. The model loading sub-module is configured to load the three-dimensional figure model of the virtual object in a case that the two-dimensional figure updating condition is met. The rendering sub-module is configured to perform two-dimensional rendering on the three-dimensional figure model to obtain the updated two-dimensional figure image of the virtual object. The image display sub-module is configured to display the updated two-dimensional figure image at the two-dimensional figure display position.

[0012] In a possible implementation, the model loading sub-module is configured to: obtain a posture parameter of the three-dimensional figure model based on the two-dimensional figure display position, where the posture parameter includes at least one of an action and an expression; and load the three-dimensional figure model based on the posture parameter.

[0013] In a possible implementation, the apparatus further includes: a first interface display module, a posture parameter obtaining module and a posture parameter saving module. The first interface display module is configured to display a posture parameter setting interface before the model loading sub-module loads the three-dimensional figure model of the virtual object. The posture parameter obtaining module is configured to obtain a posture parameter set for the two-dimensional figure display position in the posture parameter setting interface. The posture parameter saving

module is configured to save the two-dimensional figure display position corresponding to the posture parameter.

[0014] In a possible implementation, the model loading sub-module includes: a loading parameter obtaining unit and a loading unit. The loading parameter obtaining unit is configured to obtain a loading parameter of the three-dimensional figure model, where the loading parameter includes at least one of a resolution and a size. The loading unit is configured to load the three-dimensional figure model based on the loading parameter.

[0015] In a possible implementation, the apparatus is applied to a terminal, and the loading parameter obtaining unit is configured to obtain the loading parameter based on a computing capability parameter of the terminal.

[0016] In a possible implementation, the target interface is an application interface of an application program corresponding to the virtual object, and the model loading sub-module is configured to pre-load the three-dimensional figure model before the application interface is displayed.

[0017] In a possible implementation, the model loading sub-module is configured to: load the three-dimensional figure model in a case that a previous interface of the application interface is displayed; or load the three-dimensional figure model in a case that a load waiting interface is displayed, where the load waiting interface is an interface displayed before the application interface is completely loaded.

[0018] In a possible implementation, the apparatus further includes: a first information obtaining module, a second information obtaining module and a condition determining module. The first information obtaining module is configured to obtaining figure information of the two-dimensional figure image before being updated, where the figure information is used for describing figure features of a three-dimensional model of the two-dimensional figure image. The second information obtaining module is configured to obtain latest figure information of the virtual object. The condition determining module is configured to determine that the two-dimensional figure updating condition is met in a case that the figure information of the two-dimensional figure image before being updated is different from the latest figure information of the virtual object.

[0019] In a possible implementation, the second information obtaining module is configured to obtain the latest figure information of the virtual object from a server, where the latest figure information of the virtual object is uploaded to the server in a case that the virtual object is created or a figure of the virtual object changes.

[0020] In a possible implementation, the rendering sub-module is configured to: obtain a rendering parameter of the three-dimensional figure model based on the two-dimensional figure display position, where the rendering parameter includes at least one of a lens position and a rendering resolution; and perform two-dimensional rendering on the three-dimensional figure model based on the rendering parameter to obtain the updated two-dimensional figure image.

[0021] In a possible implementation, the apparatus further includes: a second interface display module, a rendering parameter obtaining module and a rendering parameter saving module. The second interface display module is configured to display a rendering parameter setting interface before the model loading sub-module loads the three-dimensional figure model of the virtual object. The rendering parameter obtaining module is configured to obtain a rendering parameter set for the two-dimensional figure display position in the rendering parameter setting interface. The rendering parameter saving module is configured to save the two-dimensional figure display position corresponding to the rendering parameter.

[0022] In a possible implementation, the apparatus includes: a condition determining module. The condition determining module is configured to determine, before the model loading module loads the three-dimensional figure model of the virtual object, that the two-dimensional figure updating condition is met in a case that an operation for changing a figure of the virtual object is received.

[0023] According to another aspect, a computer device is provided, which includes a processor and a memory. The memory stores at least one computer instruction, the at least one computer instruction is loaded and executed by the processor to implement the method for displaying a two-dimensional figure of a virtual object described above.

[0024] According to another aspect, a computer-readable storage medium is provided, which stores at least one computer instruction, the at least one computer instruction is loaded and executed by a processor to implement the method for displaying a two-dimensional figure of a virtual object described above.

[0025] According to another aspect, a computer program product or a computer program is provided, the computer program product or the computer program includes computer instructions, the computer instructions are stored in a computer-readable storage medium. A processor of a computer device reads the computer instructions from the computer-readable storage medium and executes the computer instructions to cause the computer device to perform the method for displaying a two-dimensional figure of a virtual object described above.

[0026] With the technical solutions provided in the embodiments of this application, at least the following beneficial effects can be achieved. In a case that a two-dimensional figure updating condition is met, a two-dimensional figure image is automatically generated based on a three-dimensional figure model of a virtual object, so that a latest two-dimensional figure image can be displayed in an interface immediately. Therefore, a two-dimensional figure displayed in an interface corresponding to a virtual scene can always match a three-dimensional figure of the virtual object, thereby improving a display effect of the two-dimensional figure of the virtual object.

BRIEF DESCRIPTION OF THE DRAWINGS

[0027] To describe the technical solutions in the embodiments of this application more clearly, the following briefly introduces the drawings used for describing the embodiments. Apparently, the drawings in the following description show only some embodiments of this application, and a person of ordinary skill in the art may still derive other drawings from these drawings without creative efforts.

[0028] FIG. 1 is a structural block diagram of a computer system according to an exemplary embodiment of this application.

[0029] FIG. 2 is a flowchart of a method for displaying a two-dimensional figure of a virtual object according to an exemplary embodiment of this application.

[0030] FIG. 3 is a flowchart of a method for displaying a two-dimensional figure of a virtual object according to an exemplary embodiment of this application.

[0031] FIG. 4 is a schematic interface diagram of a parameter setting interface involved in the embodiment shown in FIG. 3.

[0032] FIG. 5 is a schematic logical diagram of a process for displaying a two-dimensional figure according to an exemplary embodiment of this application.

[0033] FIG. 6 is a schematic diagram of a process for rendering a two-dimensional image based on a three-dimensional model involved in the embodiment shown in FIG. 5.

[0034] FIG. 7 is a schematic diagram of a process for displaying a two-dimensional figure image involved in the embodiment shown in FIG. 5.

[0035] FIG. 8 is a structural block diagram of an apparatus for displaying a two-dimensional figure of a virtual object according to an exemplary embodiment of this application.

[0036] FIG. 9 is a structural block diagram of a computer device according to another exemplary embodiment of this application.

DESCRIPTION OF EMBODIMENTS

[0037] Exemplary embodiments are described in detail herein, and examples of the exemplary embodiments are shown in the drawings. When the following description involves the drawings, unless otherwise indicated, the same numerals in different drawings represent the same or similar elements. The implementations described in the following exemplary embodiments do not represent all implementations that are consistent with this application. On the contrary, the implementations are merely examples of devices and methods that are described in detail in the claims and that are consistent with some aspects of this application.

[0038] It is to be understood that "several" mentioned in this specification means one or more and "a plurality of" means two or more. "And/or" describes an association relationship for describing associated objects and represents that three relationships may exist. For example, A and/or B may represent the following three cases: Only A exists, both A and B exist, and only B exists. The character "/" generally indicates a relationship "or" between the associated objects.

[0039] For ease of understanding, several terms involved in this application are explained below.

[0040] 1) Virtual scene

[0041] A virtual scene is displayed (or provided) when an application program is run on a terminal. The virtual scene may be a simulated environment scene of a real world, or may be a semi-simulated semi-fictional three-dimensional environment scene, or may be an entirely fictional three-dimensional environment scene. The virtual scene may be any one of a two-dimensional virtual scene, a 2.5-dimensional virtual scene, or a three-dimensional virtual scene. Description is made by taking a three-dimensional virtual scene as an example in the following embodiments, but this is not limited. Optionally, the virtual scene is further used for a virtual scene battle between at least two virtual roles. Optionally, there are virtual resources available to the at least two virtual roles in the virtual scene. Optionally, the virtual scene includes a virtual world which is provided with a square map. The square map includes a lower left corner region and an upper right corner region that are arranged symmetrically. Virtual roles on two opposing camps occupy the regions respectively, and the objective of each camp is to destroy a target building/stronghold/base/nexus deep in the opponent's region to win victory.

[0042] 2) Virtual object

[0043] A virtual object is a movable object in a virtual scene. The movable object may be at least one of a virtual person, a virtual animal, and a cartoon person. Optionally, when the virtual scene

is a three-dimensional virtual scene, the virtual object may be a three-dimensional model. Each virtual object has a shape and a volume in the three-dimensional virtual scene, and occupies some space in the three-dimensional virtual scene. Optionally, a virtual role is a three-dimensional role constructed based on three-dimensional human skeleton technology. The virtual role wears different skins to implement different appearances. In some embodiments, the virtual role may also be implemented through a 2.5-dimensional model or a 2-dimensional model, which is not limited in the embodiments of this application.

[0044] The method provided in this application may be applied to a sandbox game, a virtual reality application program, a three-dimensional map program, a military simulation program, a first-person shooting game (FPS), a multiplayer online battle arena game (MOBA), a role-playing game, and the like. The following embodiments are described by taking an application in a game as an example.

[0045] A virtual scene-based game involves maps of one or more game worlds. The virtual scene in the game simulates scenes in the real world. A user may control a master virtual object in the game to perform actions such as walking, running, jumping, shooting, combating, driving, casting an ability, being attacked by another virtual object, being damaged by the virtual scene, and attacking another virtual object in the virtual scene, which has relatively high interactivity. In addition, multiple users may form a team online to perform an arena game.

[0046] FIG. 1 is a structural block diagram of a computer system according to an exemplary embodiment of this application. The computer system 100 includes: a first terminal 110, a server cluster 120, and a second terminal 130.

[0047] A client 111 supporting a virtual scene is installed and run on the first terminal 110, and the client 111 may be a multiplayer online battle program. When the first terminal runs the client 111, a user interface of the client 111 is displayed on a screen of the first terminal 110. The client may be any one of a military simulation program, a MOBA game, an escape shooting game, and a simulation game (SLG), a sandbox game, and a role-playing game. In this embodiment, description is made by using an example in which the client is a MOBA game. The first terminal 110 is used by a first user 101. The first user 101 uses the first terminal 110 to control a first virtual object located in the virtual scene to perform actions, and the first virtual object may be referred to as a master virtual object of the first user 101. The actions of the first virtual object include, but are not limited to: at least one of adjusting body postures, crawling, walking, running, riding, flying, jumping, driving, picking, shooting, attacking, and throwing. For example, the first virtual object is a first virtual person, such as a simulated person role or a cartoon person role.

[0048] A client 131 supporting the virtual scene is installed and run on the second terminal 130, and the client 131 may be a multiplayer online battle program. When the second terminal 130 runs the client 131, a user interface of the client 131 is displayed on a screen of the second terminal 130. The client may be any one of a military simulation program, a MOBA game, an escape shooting game, and an SLG. In this embodiment, an example in which a client is a MOBA game is used for description. The second terminal 130 is used by a second user 102. The second user 102 uses the second terminal 130 to control a second virtual object located in the virtual scene to perform activities, and the second virtual object may be referred to as a master virtual object of the second user 102. For example, the second virtual object is a second virtual person, such as a simulated person role or a cartoon person role.

[0049] In an embodiment, the first virtual person and the second virtual person are located in the same virtual scene. In an embodiment, the first virtual person and the second virtual person may belong to the same camp, the same team, or the same organization, have a friend relationship, or have a temporary communication permission. In an embodiment, the first virtual person and the second virtual person may belong to different camps, different teams, or different organizations, or may be enemies of each other.

[0050] In an embodiment, the client installed on the first terminal 110 is the same as the client installed on the second terminal 130, or the clients installed on the two terminals are of the same type and on different operating system platforms (Android system or iOS system). The first terminal 110 may generally refer to one of multiple terminals, and the second terminal 130 may generally refer to another one of multiple terminals. In this embodiment, description is made by using the first terminal 110 and the second terminal 130 as an example. The first terminal 110 and the second terminal 130 are of the same or different device types, and the device type includes at least one of a smartphone, a tablet computer, an e-book reader, an MP3 player, an MP4 player, a laptop, and a desktop computer.

[0051] FIG. 1 shows only two terminals. However, in some embodiments, there may be multiple other terminals 140 accessing the server cluster 120. In an embodiment, one or more terminals 140 are terminals corresponding to a developer. A developing and editing platform for the client having a virtual scene is installed on the terminal 140. The developer may edit and update the client on the terminal 140 and transmit an updated client installation package to the server cluster 120 through a wired or wireless network. The first terminal 110 and the second terminal 130 may download the client installation package from the server cluster 120 to update the client.

[0052] The first terminal 110, the second terminal 130, and the another terminal 140 are connected to the server cluster 120 through a wireless network or a wired network.

[0053] The server cluster 120 includes at least one of one server, multiple servers, a cloud computing platform, and a virtualization center. The server cluster 120 is configured to provide a backend service for a client supporting a three-dimensional virtual scene. In an embodiment, the server cluster 120 is responsible for primary computing work, and the terminal is responsible for secondary computing work; or the server cluster 120 is responsible for secondary computing work, and the terminal is responsible for primary computing work; or a distributed computing architecture is adopted between the cloud server cluster 120 and the terminal to perform collaborative computing.

[0054] In a schematic example, the server cluster 120 includes a server 121 and a server 126. The server 121 includes a processor 122, a user account database 123, a battle service module 124, and a user-oriented input/output (I/O) interface 125. The processor 122 is configured to load instructions stored in the server 121, and process data in the user account database 121 and the battle service module 124. The user account database 121 is configured to store data of user accounts used by the first terminal 110, the second terminal 130, and the another terminal 140, for example, avatars of the user accounts, nicknames of the user accounts, battle effectiveness indexes of the user accounts, and service zones of the user accounts. The battle service module 124 is configured to provide multiple battle rooms for the users to battle, for example, 1V1 battle, 3V3 battle, 5V5 battle, and the like. The user-oriented I/O interface 125 is configured to establish a communication between the first terminal 110 and/or the second terminal 130 through a wireless network or a wired network for data exchange. In an embodiment, a smart signal module 127 is arranged in the server 126, and the smart signal module 127 is configured to implement some or all functional steps of a method for acquiring a virtual prop by a virtual object provided in the following embodiments.

[0055] With reference to the introduction of the nouns and the description of the implementation environment, a solution of automatically updating a two-dimensional figure image of a virtual object based on a three-dimensional virtual figure of the virtual object is provided in this application. A computer device may be configured to: display a target interface including a two-dimensional figure display position of the virtual object, where the virtual object is an object displayed in a virtual scene; display a two-dimensional figure image of the virtual object at the two-dimensional figure display position; and display an updated two-dimensional figure image of the virtual object at the two-dimensional figure display position in a case that a two-dimensional figure updating condition is met, where the updated two-dimensional figure image corresponds to a three-dimensional figure model of the virtual object.

[0056] Referring to FIG.2, FIG. 2 is a flowchart of a method for displaying a two-dimensional figure of a virtual object according to an exemplary embodiment of this application. The method may

be performed by a computer device on which an application program corresponding to the virtual scene is run. For example, the computer device may be a terminal (for example, any terminal in FIG. 1). As shown in FIG. 2, a process of displaying an updated two-dimensional figure image of the virtual object at the two-dimensional figure display position in a case that a two-dimensional figure updating condition is met may include the following steps 201 to 203.

[0057] In step 201, a three-dimensional figure model of the virtual object is loaded in a case that the two-dimensional figure updating condition is met, where the virtual object is an object displayed in the virtual scene.

[0058] In the embodiments of this application, the two-dimensional figure updating condition refers to a condition for the computer device to perform detection in a manner other than detecting a user operation.

[0059] In a possible implementation, the computer device determines whether the two-dimensional figure updating condition is met by detecting a running state of the application program corresponding to the virtual scene.

[0060] In a possible implementation, the computer device determines whether the two-dimensional figure updating condition is met by detecting a figure updating state of the virtual object.

[0061] In a possible implementation, the computer device determines whether the two-dimensional figure updating condition is met by detecting the running state of the application program and the figure updating state of the virtual object.

[0062] In step 202, two-dimensional rendering is performed on the three-dimensional figure model to obtain a two-dimensional figure image of the virtual object.

[0063] In a possible implementation, the computer device performs image acquisition on the three-dimensional figure model through a virtual camera to obtain the two-dimensional figure image.

[0064] In the embodiments of this application, in a case that the two-dimensional figure image of the virtual object already exists, the computer device may perform two-dimensional rendering on the three-dimensional figure model to obtain the updated two-dimensional figure image of the virtual object in a case that the two-dimensional figure updating condition is met.

[0065] In step 203, the two-dimensional figure image is displayed at the two-dimensional figure display position.

[0066] The computer device displays, in a case that a target interface including a two-dimensional figure display position of the virtual object is displayed, the two-dimensional figure image at the two-dimensional figure display position.

[0067] In a possible implementation, in a case that an operation for changing a figure of the virtual object is received, the computer device may determine that the two-dimensional figure updating condition is met.

[0068] In the embodiments of this application, the computer device automatically detects the two-dimensional figure updating condition and performs two-dimensional rendering on the three-dimensional figure model, so that the two-dimensional figure image of the virtual object can be updated immediately. Therefore, in a case that a target interface including a two-dimensional figure display position corresponding to the two-dimensional figure image is displayed, the latest two-dimensional figure image can be displayed.

[0069] According to the solution in the embodiments of this application, the computer device may perform, without a user actively setting the two-dimensional figure image, the following operations to automatically update the two-dimensional figure image of the virtual object:

[0070] S1: displaying an application interface of an application program corresponding to a virtual scene, where the application interface includes a two-dimensional figure display position;

[0071] S2: displaying a two-dimensional figure image of the virtual object at the two-dimensional figure display position, where the virtual object is an object displayed in the virtual scene;

[0072] S3: changing a figure of the virtual object in a case that an operation for changing the figure of the virtual object is received; and determining that a two-dimensional figure updating condition is met; and

[0073] S4: displaying an updated two-dimensional figure image of the virtual object at the two-dimensional figure display position, where the updated two-dimensional figure image corresponds to a changed figure of the virtual object.

[0074] In view of the above, according to the solution shown in the embodiments of this application, in a case that a two-dimensional figure updating condition is met, a two-dimensional figure image is automatically generated based on a three-dimensional figure model of a virtual object, so that the latest two-dimensional figure image can be displayed in an interface immediately. Therefore, a two-dimensional figure displayed in an interface corresponding to a virtual scene can always match a three-dimensional figure of the virtual object, thereby improving a display effect of a two-dimensional figure of the virtual object.

[0075] The computer device detecting whether the two-dimensional figure updating condition is met based on latest figure information of the virtual object is taken as an example. Referring to FIG.3, FIG. 3 is a flowchart of a method for displaying a two-dimensional figure of a virtual object according to an exemplary embodiment of this application. The method may be performed by a computer device on which an application program corresponding to the virtual scene is run. For example, the computer device may be a terminal (for example, any terminal in FIG. 1). As shown in FIG. 3, the method may include the following steps 301 to 307.

[0076] In step 301, an application program corresponding to a virtual scene is started, where the virtual scene includes a virtual object.

[0077] In the embodiments of this application, an application program corresponding to the virtual scene is installed in the terminal. After the terminal starts the application program, the user may log in to an application account in the application program, and create and control one or more virtual objects corresponding to the application account.

[0078] In step 302, in a case that a cached two-dimensional figure image already exists at a two-dimensional figure display position corresponding to the virtual object, figure information of the cached two-dimensional figure image is obtained.

[0079] The cached two-dimensional figure image refers to a two-dimensional figure image of the virtual object before being updated. Before the two-dimensional figure image is updated, the computer device may first obtain figure information of the two-dimensional figure image before being updated.

[0080] In a possible implementation, the two-dimensional figure display position is shown as an application interface in the application program of the virtual scene.

[0081] In a possible implementation, the two-dimensional figure display position is shown as an interface outside the application program corresponding to the virtual scene, for example, an account information page corresponding to the virtual object in a website of the application program corresponding to the virtual scene, and the user may view information such as a two-dimensional figure of each virtual object in the account information page.

[0082] The figure information is used for describing figure features of a three-dimensional model.

[0083] In a possible implementation, the figure information includes appearance feature information of each part in a three-dimensional model of the virtual object. The appearance feature information may include at least one of body appearance and decorative appearance. For example,

the body appearance includes a hair style, a hair color, a face shape, an eye shape, a pupil color, and the like, and the decorative appearance includes a fashion dress, a decorative widget, and the like.

[0084] In a possible implementation, appearance of the virtual object is automatically generated by a system (for example, appearance is randomly generated) or customized by the user. For example, appearance of each part (for example, the head or the limbs) in the virtual object may be customized by the user while creating the virtual object. Alternatively, after the virtual object is created, the user may reset the appearance through a figure changing interface.

[0085] In a possible implementation, the decorative appearance of the virtual object is changed by the user through controlling the virtual object to equip with a fashion dress or widget. For example, the user may control the virtual object to equip with a fashion dress or widget in the figure changing interface.

[0086] In the embodiments of this application, a two-dimensional figure image of the virtual object that has been previously generated may be stored in the computer device, and the computer device further records figure information of the virtual object corresponding to the two-dimensional figure image when generates the two-dimensional figure image of the virtual object. The generated and stored two-dimensional figure image is referred to as a cached two-dimensional figure image.

[0087] After the application program is started, the user account is logged in, and the user is detected to select a virtual object to enter the virtual scene, the computer device may detect whether a cached two-dimensional figure image of the virtual object corresponding to each two-dimensional figure display position exists locally, and obtains figure information of the cached two-dimensional figure image in a case that the cached two-dimensional figure image of the virtual object corresponding to each two-dimensional figure display position exists.

[0088] In step 303, latest figure information of the virtual object is obtained.

[0089] In a possible implementation, the computer device obtains the latest figure information of the virtual object from a server, where the latest figure information of the virtual object is uploaded to the server in a case that the virtual object is created or a figure of the virtual object changes.

[0090] In the embodiments of this application, in a process that the user creates a virtual object or the user controls a virtual object, if a figure of the virtual object changes (for example, the user changes a body figure of the virtual object through a figure customization interface of the virtual object, or the user controls the virtual object to equip with a new fashion dress or decorative widget), the application program in the computer device generates/changes the figure of the virtual object, and uploads the latest figure information of the virtual object to the server.

[0091] In step 304, it is determined that a two-dimensional figure updating condition is met in a case that the figure information of the cached two-dimensional figure image does not match the latest figure information of the virtual object.

[0092] That is, it is determined that the two-dimensional figure updating condition is met in a case that the figure information of the two-dimensional figure image before being updated does not match the latest figure information of the virtual object.

[0093] In the embodiments of this application, the computer device compares the figure information of the cached two-dimensional figure image with the latest figure information of the virtual object on a piece basis, determines that the figure information does not match the latest figure information in a case that any piece differs, and determines that the two-dimensional figure updating condition is met.

[0094] In an embodiment, in a case that all pieces of the figure information of the cached two-dimensional figure image are the same as respective pieces of the latest figure information of the virtual object, the computer device determines that the figure information matches the latest figure information, and determines that the two-dimensional figure updating condition is not met.

[0095] In the embodiments of this application, in a process that the user controls the virtual object in the computer device, if the figure of the virtual object changes and the cached two-dimensional figure image in the computer device is not updated, the computer device may detect that the two-dimensional figure updating condition is met and trigger a subsequent two-dimensional figure image generation process.

[0096] In another possible implementation, in a case that the computer device detects that the cached two-dimensional figure image of the virtual object corresponding to each two-dimensional figure display position does not exist locally, the computer device determines that the two-dimensional figure updating condition is met.

[0097] In the embodiments of this application, after the user creates a new virtual object in the computer device or opens the application program in another new computer device and logs in to a user account of a created virtual object, the two-dimensional figure image of the virtual object corresponding to each two-dimensional figure display position may not exist in the computer device, and in this case, the computer device determines that the two-dimensional figure updating condition is met.

[0098] In another possible implementation, the computer device may determine that the two-dimensional figure updating condition is met in a case that an operation for changing a figure of the virtual object is received.

[0099] In step 305, a three-dimensional figure model of the virtual object is loaded in a case that the two-dimensional figure updating condition is met.

[0100] In the embodiments of this application, after the computer device detects that the two-dimensional figure updating condition is met, the computer device may load a three-dimensional figure model of the virtual object, to subsequently generate a latest two-dimensional figure image based on the three-dimensional figure model.

[0101] In a possible implementation, the loading a three-dimensional figure model of the virtual object includes:

obtaining a posture parameter of the three-dimensional figure model based on the two-dimensional figure display position, where the posture parameter includes at least one of an action and an expression; and loading the three-dimensional figure model based on the posture parameter.

[0102] In the embodiments of this application, for different two-dimensional figure display positions, postures of corresponding two-dimensional figures may also be different. In an embodiment, the same two-dimensional figure display position may include one or more posture parameters. For example, in a case that the virtual object performs different actions, the posture parameters of the same two-dimensional figure display position may be different.

[0103] For example, an avatar display position on an upper left corner and a two-dimensional figure display position on a lower right corner in a game interface are taken as an example. An action and an expression corresponding to the avatar display position are respectively a standing action and a smile expression. In a case that the virtual object runs, an action and an expression at the two-dimensional figure display position are respectively a running action and a breath expression. In a case that the virtual object casts an ability, the action and the expression corresponding to the two-dimensional figure display position are respectively an ability casting action (for example, an action of throwing out an item) and a serious expression.

[0104] In the embodiments of this application, the computer device may store posture parameters corresponding to different two-dimensional figure display positions, query a posture parameter according to a two-dimensional figure display position in a case that the three-dimensional figure model is loaded, and load the three-dimensional figure model according to the queried posture parameter and the latest figure parameter of the virtual object.

[0105] In a possible implementation, before the loading a three-dimensional figure model of the virtual object, the method further includes:

displaying a posture parameter setting interface; obtaining a posture parameter set for the two-dimensional figure display position in the posture parameter setting interface; and saving the two-dimensional figure display position corresponding to the posture parameter.

[0106] In the embodiments of this application, a posture parameter customization function may be provided. That is, the user may set an action and a posture of a two-dimensional figure displayed on each two-dimensional figure display position through the posture parameter setting interface. After the user sets the posture parameter of the two-dimensional figure displayed on each two-dimensional figure display position through the posture parameter setting interface, the computer device may store the posture parameter set by the user and each two-dimensional figure display position for subsequent query and use.

[0107] For example, referring to FIG.4, FIG. 4 is a schematic interface diagram of a parameter setting interface according to an embodiment of this application. As shown in FIG. 4, the parameter setting interface includes two-dimensional figure display positions 41 and posture parameter setting controls 42 respectively corresponding to the two-dimensional figure display positions 41. The posture parameter setting controls 42 include controls used for setting actions and controls used for setting expressions. In the solution shown in FIG. 4, the posture parameter setting controls 42 are embodied as a drop-down list, and the user may trigger an action/expression list through the drop-down list, to set an action/expression corresponding to a current position.

[0108] In a possible implementation, the loading a three-dimensional figure model of the virtual object includes:

obtaining a loading parameter of the three-dimensional figure model, where the loading parameter includes at least one of a resolution and a size; and loading the three-dimensional figure model based on the loading parameter.

[0109] The resolution and the size may affect definition of a two-dimensional figure image subsequently generated based on the three-dimensional figure model. For example, a high resolution of the three-dimensional figure model may lead to a high definition of the two-dimensional figure image subsequently generated based on the three-dimensional figure model, and a small size of the three-dimensional figure model may lead to a low definition of the two-dimensional figure image subsequently generated based on the three-dimensional figure model and adjusted to a size set for a two-dimensional figure display position.

[0110] In a possible implementation, in a case that the solution in the embodiments of this application is performed by a terminal, the obtaining a loading parameter of the three-dimensional figure model includes:

obtaining the loading parameter based on a computing capability parameter of the terminal.

[0111] In a possible implementation, the computing capability parameter includes a number of cores of a processor (for example, a central processing unit/graphic processing unit), a core frequency, a screen resolution, a size of an internal memory, and the like.

[0112] The solution in this application pre-process the loaded three-dimensional figure model to adapt to rendering capabilities of different models. For example, in a low-end model, a low-precision model whose precision and size are small is selected and loaded according to a reading and writing capability and a rendering capability of the model; and in a model having good performance, a high-precision model is normally loaded.

[0113] In a possible implementation, in a case that the target interface including the two-dimensional figure display position is an application interface of the application program corresponding to the virtual object, the loading a three-dimensional figure model of the virtual object includes:

pre-loading the three-dimensional figure model before the application interface is displayed.

[0114] In a possible implementation, the pre-loading the three-dimensional figure model before the application interface is displayed includes:

loading the three-dimensional figure model in a case that a previous interface of the application interface is displayed; or

loading the three-dimensional figure model in a case that a load waiting interface is displayed, where the load waiting interface is an interface displayed before the application interface is completely loaded.

[0115] In a possible implementation, the previous interface is an interface being triggered to enter the application interface. For example, the previous interface may be an interface in which an interface entrance of the application interface is located. For example, in a case that the computer device displays an interface A, the interface A includes an entrance for being triggered to display an interface B, and the interface B includes the two-dimensional figure display position, the interface A may be referred to as a previous interface of the interface B. Alternatively, the previous interface may be an interface of a virtual scene for being triggered to enter the application interface. For example, in a case that the computer device displays an interface of a virtual scene of a map A, the virtual scene

includes a specified virtual object or prop entering a virtual scene B of a map B, and the user controls the virtual object to interact with the specified virtual object or prop to enter an interface corresponding to the virtual scene B. In this case, the interface of the virtual scene B is referred to as the application interface, and the interface of the virtual scene A is referred to as the previous interface of the application interface.

[0116] In a possible implementation, the previous interface is an interface displayed on an upper layer of the application interface during display of the application interface. For example, in a case that the computer device displays an interface A, an interface B is triggered to be displayed on an upper layer of the interface A in a superimposition manner through an entrance in the interface A, after the interface B is closed, the computer device returns to display the interface A, and the interface A includes the two-dimensional figure display position, the interface B may be referred to as a previous interface of the interface A.

[0117] In a possible implementation, the load waiting interface is an interface displayed in a case of jumping from one interface to display another interface and the another interface is not completely loaded. For example, in a case that the computer device displays an interface A, display of an interface B is triggered, a rendering time for a first frame of the interface B is long, and seamless jumping cannot be implemented. In this case, the computer device first displays an intermediate interface, the intermediate interface may display a display progress bar or progress pattern (for example, a rotating circle) of the interface B, and the intermediate interface is referred to as the load waiting interface.

[0118] That is, in the embodiments of this application, in terms of a loading time, the three-dimensional figure model may be pre-loaded according to an actual usage scene. For example, the three-dimensional figure model is loaded during loading, so that the latest two-dimensional figure image of the three-dimensional figure can be displayed as soon as possible when the application page is loaded, thereby improving the updating and display timeliness of the two-dimensional figure image.

[0119] In step 306, two-dimensional rendering is performed on the three-dimensional figure model to obtain a two-dimensional figure image of the virtual object.

[0120] In the embodiments of this application, the computer device may obtain an updated two-dimensional figure image of the virtual object by performing two-dimensional rendering on the three-dimensional figure model.

[0121] In a possible implementation, the performing two-dimensional rendering on the three-dimensional figure model to obtain a two-dimensional figure image of the virtual object includes:

obtaining a rendering parameter of the three-dimensional figure model based on the two-dimensional figure display position, where the rendering parameter includes at least one of a lens position and a rendering resolution; and performing two-dimensional rendering on the three-dimensional figure model based on the rendering parameter to obtain the two-dimensional figure image.

[0122] In the embodiments of this application, for different two-dimensional figure display positions, rendering parameters of corresponding two-dimensional figures may also be different. In an embodiment, the same two-dimensional figure display position may include one or more rendering parameters. For example, in a case that the virtual object performs different actions, the rendering parameters of the same two-dimensional figure display position may be different.

[0123] For example, an avatar display position on an upper left corner and a two-dimensional figure display position on a lower right corner in a game interface are taken as an example. A lens position and a rendering resolution of the avatar display position are respectively a front face position and 720p. In a case that the virtual object runs, a lens position and a rendering resolution of the two-dimensional figure display position are respectively a side position and 720p. In a case that the virtual object casts an ability, the lens position and the rendering resolution of the two-dimensional figure display position are respectively a side angle and 1080p.

[0124] In the embodiments of this application, the computer device may store rendering parameters corresponding to different two-dimensional figure display positions, query a rendering parameter according to a two-dimensional figure display position in a case of performing two-dimensional rendering on the three-dimensional figure model, and perform two-dimensional rendering on the three-dimensional figure model based on the queried rendering parameter.

[0125] In a possible implementation, before the loading a three-dimensional figure model of the virtual object, the method further includes:

displaying a rendering parameter setting interface; obtaining a rendering parameter set for the two-dimensional figure display position in the rendering parameter setting interface; and saving the two-dimensional figure display position corresponding to the rendering parameter.

[0126] In the embodiments of this application, a rendering parameter customization function may be provided. That is, the user may set a lens position and a resolution of a two-dimensional figure displayed on each two-dimensional figure display position through the rendering parameter setting interface. After the user sets the rendering parameter of the two-dimensional figure displayed on each two-dimensional figure display position through the rendering parameter setting interface, the

computer device may store the rendering parameter set by the user and each two-dimensional figure display position for subsequent query and use.

[0127] The posture parameter setting interface and the rendering parameter setting interface may be the same interface or may be different interfaces.

[0128] For example, the posture parameter setting interface and the rendering parameter setting interface being the same interface is taken as an example. As shown in FIG. 4, the parameter setting interface further includes rendering parameter setting controls 43 corresponding to the two-dimensional figure display positions 41. The rendering parameter setting controls 43 include controls used for setting lens positions and controls used for setting resolutions. In the solution shown in FIG. 4, the rendering parameter setting controls 43 are embodied as a drop-down list, and the user may trigger a lens position/resolution list through the drop-down list, to set a lens position/resolution of a current position.

[0129] In the solution shown in the embodiments of this application, in a case that a three-dimensional figure is converted into a two-dimensional figure image, a characteristic of Camera.Render in the unity may be utilized to output content of a specific action of a role photographed by a virtual camera onto rendertexture created by the unity. In this way, rendertexture owns data of a figure of the role, and texture is then converted into a picture type by reading pixels on the texture, to convert a three-dimensional figure into a two-dimensional image and save the two-dimensional image.

[0130] In a possible implementation, a manner point bounding for a lens is used for processing lens positions of the virtual camera. That is, a lens parameter (that is, the lens position) is preset on the three-dimensional figure model, and a lens is directly invoked during real-time rendering.

[0131] In step 307, in a case that a target interface including a two-dimensional figure display position of the virtual object is displayed, the two-dimensional figure image is displayed at the two-dimensional figure display position.

[0132] In a possible implementation, after the computer device generates the latest two-dimensional figure image of the virtual object, the computer device may store the latest two-dimensional figure image locally, and reads the latest two-dimensional figure image locally and displays the latest two-dimensional figure image at a two-dimensional figure display position corresponding to the virtual object in a case that a target interface including the two-dimensional figure display position is displayed subsequently. This manner of real-time rendering and local

storage only consumes a little traffic of data transmission and does not consume a large amount of traffic to download an image.

[0133] In another possible implementation, after the computer device generates the latest two-dimensional figure image of the virtual object, the computer device may upload the latest two-dimensional figure image to a network side, for example, to a content delivery network, and pulls the latest two-dimensional figure image from the content delivery network and displays the latest two-dimensional figure image at a two-dimensional figure display position corresponding to the virtual object in a case that a target interface including the two-dimensional figure display position is displayed subsequently. This manner of network storage may reduce the number of generation times of the two-dimensional figure image in a case that the user logs in to the same account alternately through multiple computer devices.

[0134] With the solution in the embodiments of this application, according to a scene that needs to be displayed, the computer device may control an angle of a lens, a displayed role, and an expression and an action of the role to ensure a display effect in the scene.

[0135] In addition, by displaying through the customized lens, a generated image may be controlled as an avatar, a half-length image, or a full-length image (for example, controlled through a position of the lens), and images with different degrees of precision are outputted, to meet requirements of different usage scenes in a game.

[0136] In addition, with this solution, a customized display figure may also be generated in combination with photography, and a player can customize an action, an expression, and a lens angle, and an image can be generated by rendering in real time according to a figure of the player.

[0137] In view of the above, according to the solution in the embodiments of this application, in a case that a two-dimensional figure updating condition is met, a two-dimensional figure image is automatically generated based on a three-dimensional figure model of a virtual object, so that the latest two-dimensional figure image can be displayed in an interface immediately. Therefore, a two-dimensional figure displayed in an interface corresponding to a virtual scene can always match a three-dimensional figure of the virtual object, thereby improving a display effect of a two-dimensional figure of the virtual object.

[0138] The solution shown in FIG. 2 or FIG. 3 being applicable to a game application program is taken as an example, that is, the virtual scene is a game scene, and the virtual object is a player role controlled by the user through a terminal. Referring to FIG.5, FIG. 5 is a schematic logical diagram of a process for displaying a two-dimensional figure according to an exemplary embodiment of this

application. As shown in FIG. 5, a process of displaying a two-dimensional figure may include the following steps S51 to S59.

[0139] In step S51, a game application program is started.

[0140] In step S52, a player logs in to a user account.

[0141] After the terminal starts the game application program, the user account of the player may be automatically logged in, or the player manually inputs the account, or the player selects an existing associated account.

[0142] In step S53, the player creates a role or changes a figure of a role.

[0143] In a case that the player logs in to a game firstly with a user account, the player may create a player role in a role creating interface, and an initial figure of the player role may be manually customized by the player or randomly set.

[0144] In a game process, the player may change the figure of the player role through a figure changing interface, to change body appearance or change a fashion dress/widget. In an embodiment, in a case that the player controls the player role to equip with/change an equipment having an appearance effect, the figure of the player role may also be changed.

[0145] In step S54, figure information of the player role is recorded and the figure information to a server is uploaded, and the process proceeds to S56.

[0146] After the player creates a role or change a figure of the role, the terminal may record latest figure information of the player role and update the recorded figure information to the server.

[0147] In step S55, in a case that the player does not change the figure of the role, the terminal further determines whether locally stored figure information of a two-dimensional figure image of the player role is the same as the latest figure information of the player role stored in the server. If the locally stored figure information of a two-dimensional figure image of the player role is the same as the latest figure information of the player role stored in the server, S56 is performed; otherwise, if the locally stored figure information of a two-dimensional figure image of the player role is different from the latest figure information of the player role stored in the server, S57 is performed.

[0148] In step S56, the terminal uses the locally stored two-dimensional figure image of the player role as a to-be-displayed two-dimensional figure image.

[0149] In step S57, the terminal loads a three-dimensional figure model of the player role based on the latest figure information of the player role, and performs rendering to obtain a two-dimensional figure image.

[0150] In this solution, if the user changes a login device, whether information stored on a new device is the same as information recorded on the server may be determined, the cached figure image is used if information stored on a new device is the same as information recorded on the server, and pre-rendering is performed if information stored on a new device is different from information recorded on the server.

[0151] The terminal generates different two-dimensional figure images for different positions according to the solution in the embodiment shown in FIG. 2 or FIG. 3 in combination with the figure information and a position relationship of the player role.

[0152] For example, referring to FIG.6, FIG. 6 is a schematic diagram of a process for rendering a two-dimensional image based on a three-dimensional model involved in an embodiment of this application. As shown in FIG. 6, for a three-dimensional figure model 61 of the player role, based on parameters such as a lens position, a resolution, an action, and an expression corresponding to an avatar display position, the terminal acquires a two-dimensional figure image 62 of an avatar and a two-dimensional figure image 63 of an upper body.

[0153] In step S58, the terminal outputs the two-dimensional figure image obtained through rendering as a to-be-displayed two-dimensional figure image, and locally stores the two-dimensional figure image obtained through rendering and the latest figure information.

[0154] In step S59, the to-be-displayed two-dimensional figure image is loaded in an interface of a game scene.

[0155] For example, referring to FIG.7, FIG. 7 is a schematic diagram of a process for displaying a two-dimensional figure image involved in an embodiment of this application. As shown in FIG. 7, in a game scene 70, an upper left corner position is referred to as an avatar display position 71, and a lower right corner position is referred to as a half-length image display position 72. With reference to the two-dimensional figure image shown in FIG. 6, the terminal displays the two-dimensional figure image 62 of the avatar at the position 71, and displays the two-dimensional figure image 63 of the upper body at the position 72.

[0156] In this solution, in a case that the player creates a role or changes a figure, information of a figure of a trainer may be uploaded and recorded. After the recording, a role figure may be pre-loaded and rendering may be performed while ensuring that information of the player is not changed (for example, in a case that the player clicks a figure changing completion button or exits a figure changing interface). The figure image obtained by rendering may be stored locally and invoked in a corresponding usage scene.

[0157] That is, the action of the two-dimensional image generated according to this solution may be adjusted according to a usage scene, and dressing may be changed in real time according to changes of the figure of the trainer, thereby ensuring a better display effect.

[0158] FIG. 8 is a structural block diagram of an apparatus for displaying a two-dimensional figure of a virtual object according to an exemplary embodiment of this application. As shown in FIG. 8, the apparatus includes: an interface display module 801 and an image display module 802. The interface display module 801 is configured to display a target interface including a two-dimensional figure display position of the virtual object, where the virtual object is an object displayed in a virtual scene. The image display module 802 is configured to display a two-dimensional figure image of the virtual object at the two-dimensional figure display position. The image display module 802 is further configured to display an updated two-dimensional figure image of the virtual object at the two-dimensional figure display position in a case that a two-dimensional figure updating condition is met, where the updated two-dimensional figure image corresponds to a three-dimensional figure model of the virtual object.

[0159] In a possible implementation, the image display module 802 includes: a model loading sub-module, a rendering sub-module and an image display sub-module. The model loading sub-module is configured to load the three-dimensional figure model of the virtual object in a case that the two-dimensional figure updating condition is met. The rendering sub-module is configured to perform two-dimensional rendering on the three-dimensional figure model to obtain the updated two-dimensional figure image of the virtual object. The image display sub-module is configured to display the updated two-dimensional figure image at the two-dimensional figure display position.

[0160] In a possible implementation, the model loading sub-module is configured to: obtain a posture parameter of the three-dimensional figure model based on the two-dimensional figure display position, where the posture parameter includes at least one of an action and expression; and load the three-dimensional figure model based on the posture parameter.

[0161] In a possible implementation, the apparatus further includes: a first interface display module, a posture parameter obtaining module and a posture parameter saving module. The first interface display module is configured to display a posture parameter setting interface before the model loading sub-module loads the three-dimensional figure model of the virtual object. The posture parameter obtaining module is configured to obtain a posture parameter set for the two-dimensional figure display position in the posture parameter setting interface. The posture parameter saving module is configured to save the two-dimensional figure display position corresponding to the posture parameter.

[0162] In a possible implementation, the model loading sub-module includes: a loading parameter obtaining unit and a loading unit. The loading parameter obtaining unit is configured to obtain a loading parameter of the three-dimensional figure model, where the loading parameter includes at least one of a resolution and a size. The loading unit is configured to load the three-dimensional figure model based on the loading parameter.

[0163] In a possible implementation, the apparatus is applied to a terminal, and the loading parameter obtaining unit is configured to obtain the loading parameter based on a computing capability parameter of the terminal.

[0164] In a possible implementation, the target interface is an application interface of an application program corresponding to the virtual object, and the model loading sub-module is configured to pre-load the three-dimensional figure model before the application interface is displayed.

[0165] In a possible implementation, the model loading sub-module is configured to: load the three-dimensional figure model in a case that a previous interface of the application interface is displayed; or load the three-dimensional figure model in a case that a load waiting interface is displayed, where the load waiting interface is an interface displayed before the application interface is completely loaded.

[0166] In a possible implementation, the apparatus further includes: a first information obtaining module, a second information obtaining module and a condition determining module. The first information obtaining module is configured to obtaining figure information of the two-dimensional figure image before being updated, where the figure information is used for describing figure features of a three-dimensional model of the two-dimensional figure image. The second information obtaining module is configured to obtain latest figure information of the virtual object. The condition determining module is configured to determine that the two-dimensional figure updating condition is met in a case that the figure information of the two-dimensional figure image before being updated is different from the latest figure information of the virtual object.

[0167] In a possible implementation, the second information obtaining module is configured to obtain the latest figure information of the virtual object from a server, where the latest figure information of the virtual object is uploaded to the server in a case that the virtual object is created or a figure of the virtual object changes.

[0168] In a possible implementation, the rendering sub-module is configured to: obtain a rendering parameter of the three-dimensional figure model based on the two-dimensional figure

display position, where the rendering parameter includes at least one of a lens position and a rendering resolution; and perform two-dimensional rendering on the three-dimensional figure model based on the rendering parameter to obtain the updated two-dimensional figure image.

[0169] In a possible implementation, the apparatus further includes: a second interface display module, a rendering parameter obtaining module and a rendering parameter saving module. The second interface display module is configured to display a rendering parameter setting interface before the model loading sub-module loads the three-dimensional figure model of the virtual object. The rendering parameter obtaining module is configured to obtain a rendering parameter set for the two-dimensional figure display position in the rendering parameter setting interface. The rendering parameter saving module is configured to save the two-dimensional figure display position corresponding to the rendering parameter.

[0170] In a possible implementation, the apparatus includes: a condition determining module. The condition determining module is configured to determine, before the model loading sub-module loads the three-dimensional figure model of the virtual object, that the two-dimensional figure updating condition is met in a case that an operation for changing a figure of the virtual object is received.

[0171] In a possible implementation, the apparatus includes: a figure changing module. The figure changing module is configured to change a figure of the virtual object in a case that an operation for changing the figure of the virtual object is received.

[0172] In view of the above, according to the solution shown in the embodiments of this application, in a case that a two-dimensional figure updating condition is met, a two-dimensional figure image is automatically generated based on a three-dimensional figure model of a virtual object, so that the latest two-dimensional figure image can be displayed in an interface immediately. Therefore, a two-dimensional figure displayed in an interface corresponding to a virtual scene can always match a three-dimensional figure of the virtual object, thereby improving a display effect of a two-dimensional figure of the virtual object.

[0173] FIG. 9 is a structural block diagram of a computer device 900 according to an exemplary embodiment of this application. The computer device 900 may be a smartphone, a tablet computer, a Moving Picture Experts Group Audio Layer III (MP3) player, a Moving Picture Experts Group Audio Layer IV (MP4) player, a notebook computer, or a desktop computer. The computer device 900 may be further referred to as another name such as user equipment, a portable terminal, a laptop terminal, or a desktop terminal.

[0174] Generally, the computer device 900 includes a processor 901 and a memory 902.

[0175] The memory 902 may include one or more computer-readable storage medium. The computer-readable storage medium may be non-transitory. The memory 902 may further include a high-speed random access memory and a non-volatile memory, for example, one or more disk storage devices or flash storage devices. In some embodiments, a non-transitory computer-readable storage medium in the memory 902 is configured to store at least one computer instruction, the at least one computer instruction is configured to be executed by the processor 901 to implement the method provided in the method embodiments of this application.

[0176] In some embodiments, the computer device 900 further includes a peripheral device interface 903 and at least one peripheral device. The processor 901, the memory 902, and the peripheral device interface 903 may be connected through a bus or a signal cable. Each peripheral device may be connected to the peripheral device interface 903 through a bus, a signal cable, or a circuit board. Specifically, the peripheral device includes at least one of a radio frequency circuit 904, a display screen 905, a camera component 906, an audio circuit 907, a positioning component 908, and a power supply 909.

[0177] In some embodiments, the computer device 900 may further include one or more sensors 910. The one or more sensors 910 include, but are not limited to: an acceleration sensor 911, a gyroscope sensor 912, a pressure sensor 913, a fingerprint sensor 914, an optical sensor 915, and a proximity sensor 916.

[0178] A person skilled in the art may understand that the structure shown in FIG. 9 does not constitute any limitation to the computer device 900, and the computer device may include more components or fewer components than those shown in the figure, or some components may be combined, or a different component deployment may be used.

[0179] A person of ordinary skill in the art may understand that all or some of the steps of the methods in the embodiments may be implemented by a program instructing relevant hardware. The program may be stored in a computer-readable storage medium. The computer-readable storage medium may be the computer-readable storage medium included in the memory in the foregoing embodiment, or may be a computer-readable storage medium that exists independently and that is not assembled in a terminal. The computer-readable storage medium stores at least one instruction, at least one program, a code set, and an instruction set, the at least one instruction, the at least one program, the code set, or the instruction set being loaded and executed by a processor to implement the method provided in the foregoing embodiments of this application.

[0180] In an embodiment, the computer-readable storage medium may include a read-only memory (ROM), a random access memory (RAM), a solid state drive (SSD), or an optical disc. The RAM may include a resistance random access memory (ReRAM) and a dynamic random access memory (DRAM). The sequence numbers of the foregoing embodiments of this application are merely for description purpose, and are not intended to indicate preference of the embodiments.

[0181] A person of ordinary skill in the art may understand that all or some of the steps in the embodiments may be implemented by hardware or a program instructing relevant hardware. The program may be stored in a computer-readable storage medium. The storage medium may be a ROM, a magnetic disk, or an optical disc.

[0182] In an exemplary embodiment, a computer program product or a computer program is provided. The computer program product or the computer program includes computer instructions, and the computer instructions are stored in a computer-readable storage medium. The processor of the computer device reads the computer instructions from the computer-readable storage medium, and the processor executes the computer instructions, to cause the computer device to perform the method in the foregoing embodiments.

[0183] A person skilled in the art can easily figure out other implementation solutions of this application after considering this specification and practicing the solution that is disclosed herein. This application is intended to cover any variations, uses, or adaptive changes of this application. Such variations, uses, or adaptive changes follow the general principles of this application, and include well-known knowledge and conventional technical means in the art that are not disclosed in this application. This specification and the embodiments are merely considered as examples, and the protection scope of this application is pointed out in the claims.

[0184] This application is not limited to the accurate structures/solutions that are described above and that are shown in the accompanying drawings, and modifications and changes may be made without departing from the scope of this application. The scope of this application is subject only to the appended claims.

CLAIMS

1. A method for displaying a two-dimensional figure of a virtual object, performed by a computer device, the method comprising:

displaying a target interface comprising a two-dimensional figure display position of the virtual object, wherein the virtual object is an object displayed in a virtual scene;

displaying a two-dimensional figure image of the virtual object at the two-dimensional figure display position; and

displaying, in a case that a two-dimensional figure updating condition is met, an updated two-dimensional figure image of the virtual object at the two-dimensional figure display position, wherein the updated two-dimensional figure image corresponds to a three-dimensional figure model of the virtual object.

2. The method according to claim 1, wherein the displaying, in a case that a two-dimensional figure updating condition is met, an updated two-dimensional figure image of the virtual object at the two-dimensional figure display position comprises:

loading the three-dimensional figure model of the virtual object in a case that the two-dimensional figure updating condition is met;

performing two-dimensional rendering on the three-dimensional figure model to obtain the updated two-dimensional figure image of the virtual object; and

displaying the updated two-dimensional figure image at the two-dimensional figure display position.

3. The method according to claim 2, wherein the loading the three-dimensional figure model of the virtual object comprises:

obtaining a posture parameter of the three-dimensional figure model based on the two-dimensional figure display position, wherein the posture parameter comprises at least one of an action and an expression; and

loading the three-dimensional figure model based on the posture parameter.

4. The method according to claim 3, wherein before the loading the three-dimensional figure model of the virtual object, the method further comprises:

displaying a posture parameter setting interface;

obtaining a posture parameter set for the two-dimensional figure display position in the posture parameter setting interface; and

saving the two-dimensional figure display position corresponding to the posture parameter.

5. The method according to claim 2, wherein the loading the three-dimensional figure model of the virtual object comprises:

obtaining a loading parameter of the three-dimensional figure model, wherein the loading parameter comprises at least one of a resolution and a size; and

loading the three-dimensional figure model based on the loading parameter.

6. The method according to claim 5, wherein the method is performed by a terminal, and the obtaining a loading parameter of the three-dimensional figure model comprises:

obtaining the loading parameter based on a computing capability parameter of the terminal.

7. The method according to claim 2, wherein the target interface is an application interface of an application program corresponding to the virtual object, and the loading the three-dimensional figure model of the virtual object comprises:

pre-loading the three-dimensional figure model before the application interface is displayed.

8. The method according to claim 7, wherein the pre-loading the three-dimensional figure model before the application interface is displayed comprises:

loading the three-dimensional figure model in a case that a previous interface of the application interface is displayed; or

loading the three-dimensional figure model in a case that a load waiting interface is displayed, wherein the load waiting interface is an interface displayed before the application interface is completely loaded.

9. The method according to claim 2, wherein before the loading the three-dimensional figure model of the virtual object, the method further comprises:

obtaining figure information of the two-dimensional figure image before being updated, wherein the figure information is used for describing figure features of a three-dimensional model of the two-dimensional figure image;

obtaining latest figure information of the virtual object; and

determining that the two-dimensional figure updating condition is met in a case that the figure information of the two-dimensional figure image before being updated is different from the latest figure information of the virtual object.

10. The method according to claim 9, wherein the obtaining latest figure information of the virtual object comprises:

obtaining the latest figure information of the virtual object from a server, wherein the latest figure information of the virtual object is uploaded to the server in a case that the virtual object is created or a figure of the virtual object changes.

11. The method according to claim 2, wherein the performing two-dimensional rendering on the three-dimensional figure model to obtain the updated two-dimensional figure image of the virtual object comprises:

obtaining a rendering parameter of the three-dimensional figure model based on the two-dimensional figure display position, wherein the rendering parameter comprises at least one of a lens position and a rendering resolution; and

performing two-dimensional rendering on the three-dimensional figure model based on the rendering parameter to obtain the updated two-dimensional figure image.

12. The method according to claim 11, wherein before the loading the three-dimensional figure model of the virtual object, the method further comprises:

displaying a rendering parameter setting interface;

obtaining a rendering parameter set for the two-dimensional figure display position in the

rendering parameter setting interface; and

saving the two-dimensional figure display position corresponding to the rendering parameter.

13. The method according to claim 2, wherein before the loading the three-dimensional figure model of the virtual object, the method further comprises:

determining that the two-dimensional figure updating condition is met in a case that an operation for changing a figure of the virtual object is received.

14. An apparatus for displaying a two-dimensional figure of a virtual object, comprising:

an interface display module, configured to display a target interface comprising a two-dimensional figure display position of the virtual object, wherein the virtual object is an object displayed in a virtual scene; and

an image display module, configured to display a two-dimensional figure image of the virtual object at the two-dimensional figure display position, wherein

the image display module is further configured to display, in a case that a two-dimensional figure updating condition is met, an updated two-dimensional figure image of the virtual object at the two-dimensional figure display position, wherein the updated two-dimensional figure image corresponds to a three-dimensional figure model of the virtual object.

15. A computer device, comprising a processor and a memory, the memory storing at least one computer instruction, the at least one computer instruction being loaded and executed by the processor to implement the method for displaying a two-dimensional figure of a virtual object according to any one of claims 1 to 13.

16. A computer-readable storage medium, storing at least one computer instruction, the at least one computer instruction being loaded and executed by a processor to implement the method for displaying a two-dimensional figure of a virtual object according to any one of claims 1 to 13.

17. A computer program product, comprising computer instructions, the computer instructions being executed by a processor to implement the method for displaying a two-dimensional figure of a

virtual object according to any one of claims 1 to 13.