

METHOD AND APPARATUS FOR CONTROLLING VIRTUAL OBJECT, TERMINAL, STORAGE MEDIUM, AND PROGRAM PRODUCT

5 **[0001]** This application claims priority to Chinese Patent Application No. 202011497135.5, entitled "METHOD AND APPARATUS FOR CONTROLLING VIRTUAL OBJECT, TERMINAL, AND STORAGE MEDIUM" filed on December 17, 2020, which is incorporated by reference in its entirety.

FIELD OF THE TECHNOLOGY

10 **[0002]** Embodiments of this disclosure relate to the field of human-computer interaction technologies, and in particular, to a method and apparatus for controlling a virtual object, a terminal, a storage medium, and a program product.

BACKGROUND OF THE DISCLOSURE

15 **[0003]** A battle game is a game in which multiple user accounts compete in the same scene. A player can operate a virtual object in a virtual environment to perform motions such as walking, running, climbing, and shooting. In a process that multiplayer teams fight together, the player can actively control a moving direction of the virtual object through a joystick, or control the virtual object to automatically follow a virtual object controlled by another player to move.

20 **[0004]** In the related technology, an automatic following control is disposed on a battle interface, and the player can trigger the automatic following control, so that the virtual object automatically follows another virtual object in the same camp with the virtual object, for example, automatically follows a virtual object of a leader in a team by default, or selects a virtual object intended to be followed from a camp list.

25 **[0005]** However, in the related technology, the method for controlling a virtual object to automatically follow another virtual object requires a user to manually open a virtual object list for the camp of the virtual object, select a following object from the list, and then trigger the following control. These operations are relatively cumbersome, and only the virtual object in the same camp can be followed actively.

SUMMARY

[0006] Embodiments of this disclosure provide a method and apparatus for controlling a virtual object, a terminal, a storage medium, and a program product, to simplify a user operation of controlling a first virtual object to perform automatic following. The technical solutions are as follows.

[0007] According to an aspect, an embodiment of this disclosure provides a method for controlling a virtual object. The method is applicable to a terminal. The method includes: displaying a first virtual object located in a virtual environment; displaying, in response to a moving operation on the first virtual object, one or more candidate object identifiers corresponding to one or more candidate virtual objects respectively in the virtual environment, the candidate virtual objects being located in an orientation region of the first virtual object; and controlling, in response to a trigger operation on a target object identifier in the candidate object identifiers, the first virtual object to automatically follow a target virtual object in the virtual environment, the target virtual object being a virtual object corresponding to the target object identifier.

[0008] According to another aspect, an embodiment of this disclosure provides an apparatus for controlling a virtual object. The apparatus includes: a first display module, configured to display a first virtual object located in a virtual environment; a second display module, configured to display, in response to a moving operation on the first virtual object, one or more candidate object identifiers corresponding to one or more candidate virtual objects respectively in the virtual environment, the candidate virtual objects being located in an orientation region of the first virtual object; and a first control module, configured to control, in response to a trigger operation on a target object identifier in the candidate object identifiers, the first virtual object to automatically follow a target virtual object in the virtual environment, the target virtual object being a virtual object corresponding to the target object identifier.

[0009] According to another aspect, an embodiment of this disclosure provides a terminal, including a processor and a memory, the memory storing at least one instruction, at least one program, a code set or 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 the processor to implement the method for controlling a virtual object in the foregoing aspect.

[0010] According to another aspect, an embodiment of this disclosure provides a computer-readable storage medium, storing at least one computer program, the computer program being loaded and executed by a processor to implement the method for controlling a virtual object described in the foregoing aspect.

[0011] According to an aspect of this disclosure, a computer program product or a computer program is provided, including computer instructions, the computer instructions being stored in a computer-readable storage medium. A processor of a terminal reads the computer instructions from the computer-readable storage medium, and executes the computer instructions, to cause the terminal to implement the method for controlling a virtual object provided in the optional implementations of the foregoing aspect.

[0012] In the embodiments of this disclosure, candidate object identifiers are displayed when a moving operation on a first virtual object is received, so that a user can clearly determine virtual objects in an orientation region of the first virtual object through the candidate object identifiers when controlling the first virtual object to move, and select a target virtual object from candidate virtual objects through a trigger operation. Therefore, the first virtual object automatically follows the target virtual object, and there is no need to select a following object from a virtual object list and perform a following operation, thereby simplifying a user operation, and improving efficiency of making the first virtual object enter an automatic following state. The candidate virtual objects are determined based on an orientation of the first virtual object, so that the determined candidate virtual objects meet actual operation needs of the user better, and validity of the candidate object identifiers are improved.

BRIEF DESCRIPTION OF THE DRAWINGS

[0013] FIG. 1 is a schematic diagram of triggering a first virtual object to perform automatic following in the related technology.

[0014] FIG. 2 is a schematic diagram of an implementation environment according to an exemplary embodiment of this disclosure.

[0015] FIG. 3 is a flowchart of a method for controlling a virtual object according to an exemplary embodiment of this disclosure.

[0016] FIG. 4 is a schematic diagram of a virtual environment picture according to an exemplary embodiment of this disclosure.

[0017] FIG. 5 is a schematic diagram of triggering to display candidate object identifiers according to an exemplary embodiment of this disclosure.

[0018] FIG. 6 is a flowchart of a method for controlling a virtual object according to another exemplary embodiment of this disclosure.

[0019] FIG. 7 is a schematic diagram of determining an orientation region according to an exemplary embodiment of this disclosure.

[0020] FIG. 8 is a schematic diagram of triggering to display candidate object identifiers according to another exemplary embodiment of this disclosure.

[0021] FIG. 9 is a flowchart of a method for controlling a virtual object according to another exemplary embodiment of this disclosure.

5 **[0022]** FIG. 10 is a schematic diagram of triggering to display followable labels according to an exemplary embodiment of this disclosure.

[0023] FIG. 11 is a schematic diagram of updating a display state of a virtual joystick according to an exemplary embodiment of this disclosure.

[0024] FIG. 12 is a flowchart of a method for controlling a virtual object according to
10 another exemplary embodiment of this disclosure.

[0025] FIG. 13 is a structural block diagram of an apparatus for controlling a virtual object according to an exemplary embodiment of this disclosure.

[0026] FIG. 14 is a structural block diagram of a terminal according to an exemplary embodiment of this disclosure.

15 DESCRIPTION OF EMBODIMENTS

[0027] First, terms involved in the embodiments of this disclosure are introduced as follows.

[0028] 1) Virtual environment

[0029] It is a virtual environment displayed (or provided) when an application is run on a terminal. The virtual environment may be a simulated environment of a real world, or may be a
20 semi-simulated semi-fictional three-dimensional environment (3D), or may be an entirely fictional 3D environment. The virtual environment may be any one of a 2D virtual environment, a 2.5-dimensional virtual environment, and a 3D virtual environment, an example in which the virtual environment is a 3D virtual environment is used for description in the following embodiment, but this is not limited. Optionally, the virtual environment is further used for a virtual environment
25 battle between at least two virtual roles. Optionally, there are virtual resources available to the at least two virtual roles in the virtual environment.

[0030] 2) Virtual object

[0031] A virtual object is a movable object in a virtual scene. The movable object may be at least one of a virtual character, a virtual animal, and an animation character. Optionally, when the
30 virtual scene is a 3D virtual scene, the virtual object may be a 3D model. Each virtual object has a shape and a volume in the 3D virtual scene, and occupies some space in the 3D virtual scene.

Optionally, the virtual role is a 3D role constructed based on a 3D human skeleton technology. The virtual role wears different skins to implement different appearances. In some implementations, the virtual role may be alternatively implemented by using a 2.5-dimensional model or a 2D model. This is not limited in this embodiment of this disclosure.

5 **[0032]** 3) User interface (UI) control: It is any visual control or element that can be seen in a UI of an application, such as a picture, an input box, a text box, a button, or a label. Some UI controls respond to an operation of a user.

10 **[0033]** In the related technology, a game application usually can provide an automatic following function for a first virtual object, and a user selects a virtual object that the first virtual object needs to follow through a corresponding operation, so that the first virtual object automatically follows the selected virtual object. As shown in FIG. 1, when receiving a trigger operation on a virtual object list 101, a terminal displays a virtual object selection interface 102, and the virtual object selection interface 102 includes object selection controls of virtual objects. When receiving a trigger operation on an object selection control 103 of a virtual object c, the terminal
15 displays an automatic following control 104, and a user may trigger the automatic following control 104, so that a first virtual object automatically follows the virtual object c.

20 **[0034]** In the foregoing related technology, the method for controlling a virtual object to automatically follow another virtual object requires the user to manually open the virtual object list, select a following object from the list, and then trigger the automatic following control. These operations are relatively cumbersome, and the virtual object selection interface blocks a virtual environment picture during a user operation, which affects observation on a virtual environment and other operations of the user.

25 **[0035]** To resolve the foregoing problems, this disclosure provides a method for controlling a virtual object. A terminal determines candidate virtual objects in an orientation of a first virtual object based on a control operation of a user on a virtual joystick, and displays candidate object identifiers, so that the user can rapidly control, by using a simple trigger operation, the first virtual object to automatically follow the target virtual object, thereby simplifying a user operation, and improving efficiency of controlling the first virtual object to perform automatic following.

30 **[0036]** FIG. 2 is a schematic diagram of an implementation environment according to an embodiment of this disclosure. The implementation environment may include: a first terminal 110, a server 120, and a second terminal 130.

[0037] An application 111 supporting a virtual environment is installed and run on the first terminal 110. When the first terminal runs the application 111, a UI of the application 111 is

displayed on a screen of the first terminal 110. The application 111 may be any one of a military simulation program, a multiplayer online battle arena (MOBA) game, an escape shooting game, and a simulation game (SLG). In this embodiment, an example in which the application 111 is a role-playing game (RPG) is used for description. The first terminal 110 is a terminal used by a first user 112. The first user 112 uses the first terminal 110 to control a first virtual object located in the virtual environment to perform activities, and the first virtual object may be referred to as a master virtual object of the first user 112. The activities 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, throwing, and skill releasing. Exemplarily, the first virtual object is a first virtual character such as a simulated character or an animation character. In the server, a user of the application may be referred to with a user account.

[0038] An application 131 supporting a virtual environment is installed and run on the second terminal 130. When the second terminal 130 runs the application 131, a UI of the application 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 game. In this embodiment, an example in which the application 131 is an RPG is used for description. The second terminal 130 is a terminal used by a second user 132. The second user 132 uses the second terminal 130 to control a second virtual object located in the virtual environment to perform activities, and the second virtual object may be referred to as a master virtual role of the second user 132. Exemplarily, the second virtual object is a second virtual character such as a simulated character or an animation character.

[0039] Optionally, the first virtual object and the second virtual object are located in the same virtual world. Optionally, the first virtual object and the second virtual object may belong to the same camp, the same team, or the same organization, have a friend relationship with each other, or have a temporary communication permission. Optionally, the first virtual object and the second virtual object may belong to different camps, different teams, or different organizations, or have a hostile relationship with each other. The first user 112 may use the first terminal 110 to control the first virtual object to automatically follow the second virtual object controlled by the second user 132 in a virtual environment, and similarly, the second user 132 may use the second terminal 130 to control the second virtual object to automatically follow the first virtual object controlled by the first user 112 in the virtual environment.

[0040] Optionally, the applications run on the first terminal 110 and the second terminal 130 are the same, or the applications installed on the two terminals are the same type of applications 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 one of multiple terminals. In this embodiment, only the first terminal 110 and the second terminal 130 are used as an example for description. The first terminal 110 and the second terminal 130 are of the same or different device types. The device type includes at least one of a smartphone, a tablet computer, an e-book reader, a MP3 player, a MP4 player, a laptop, and a desktop computer.

[0041] FIG. 2 shows only two terminals. However, multiple other terminals may access the server 120 in different embodiments. Optionally, one or more terminals are terminals corresponding to a developer. A developing and editing platform for the application supporting a virtual environment is installed on the terminal. The developer may edit and update the application on the terminal and transmit an updated application installation package to the server 120 by using a wired or wireless network. The first terminal 110 and the second terminal 130 may download the application installation package from the server 120 to update the application.

[0042] The first terminal 110, the second terminal 130, and the other terminals are connected to the server 120 through a wireless or wired network.

[0043] The server 120 includes at least one of one server, a server cluster including multiple servers, a cloud computing platform, and a virtualization center. The server 120 is configured to provide backend services for an application supporting a 3D virtual environment. Optionally, the server 120 is responsible for primary computing work, and the terminal is responsible for secondary computing work; the server 120 is responsible for secondary computing work, and the terminal is responsible for primary computing work; or the server 120 and the terminal perform collaborative computing by using a distributed computing architecture between each other.

[0044] In an exemplary example, the server 120 includes a memory 121, 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 120, and process data in the user account database 123 and the battle service module 124. The user account database 123 is configured to store data of user accounts used by the first terminal 110, the second terminal 130, and the another terminal, 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, a 1V1 battle room, a 3V3 battle room, and a 5V5 battle room. The user-oriented I/O interface 125 is configured to establish communication between the first terminal 110 and/or the second terminal 130 through a wireless or wired network for data exchange.

[0045] FIG. 3 is a flowchart of a method for controlling a virtual object according to an exemplary embodiment of this disclosure. In this embodiment, an example in which the method is performed by a terminal supporting a virtual environment is used for description. The method includes the following steps 301 to 303.

5 **[0046]** Step 301. Display a first virtual object in a virtual environment.

[0047] In a possible implementation, the terminal displays the virtual environment through a virtual environment picture. The virtual environment picture is a picture of observing a virtual environment from the perspective of a virtual object. The perspective is an observation angle for observation from a first-person perspective or a third-person perspective of the virtual object in the
10 virtual environment. In this embodiment of this disclosure, the perspective is an angle for observing the virtual object by using a camera model in the virtual environment.

[0048] Optionally, the camera model automatically follows the virtual object in the virtual environment. That is, when a position of the virtual object in the virtual environment changes, a position of the camera model following the virtual object in the virtual environment changes
15 simultaneously, and the camera model is always within a preset distance range of the virtual object in the virtual environment. During automatic following, relative positions between the camera model and the virtual object remain unchanged.

[0049] The camera model is a 3D model located around a virtual object in a virtual environment. When the first-person perspective is used, the camera model is located nearby the
20 head of the virtual object or at the head of the virtual object. When the third-person perspective is used, the camera model may be located behind the virtual object and bound to the virtual object, or may be located at any position away from the virtual object by a preset distance. The virtual object located in the virtual environment may be observed from different angles through the camera model. Optionally, when the third-person perspective is a first-person over-shoulder perspective, the
25 camera model is located behind the virtual object (for example, the head and shoulders of the virtual character). Optionally, in addition to the first-person perspective and the third-person perspective, the perspective also includes other perspectives, such as a top perspective. When the top perspective is used, the camera model may be located above the head of the virtual object. The top perspective is a perspective for observing the virtual environment at an angle from the air. Optionally, the
30 camera model is not actually displayed in the virtual environment. In other words, the camera model is not displayed in the virtual environment displayed in the UI. An example in which the camera model is located at any position away from the virtual object by the preset distance is used for description. Optionally, one virtual object corresponds to one camera model, and the camera

model may rotate with the virtual object as a rotation center. For example, the camera model is rotated with any point of the virtual object as the rotation center. During rotation, the camera model is not only rotated at an angle, but also displaced. During rotation, a distance between the camera model and the rotation center remains unchanged, that is, the camera model is rotated on a surface of a sphere with the rotation center as a sphere center. Any point of the virtual object may be the head or the torso of the virtual object, or any point around the virtual object. This is not limited in this embodiment of this disclosure. Optionally, when the camera model observes the virtual object, a center of the perspective of the camera model points to a direction from a point on the spherical surface at which the camera model is located to the sphere center.

[0050] Optionally, the camera model may alternatively observe the virtual object at a preset angle in different directions of the virtual object.

[0051] Optionally, the virtual environment picture in this embodiment of this disclosure is a picture of observing the virtual environment from the perspective of a first virtual object. The first virtual object is a virtual object corresponding to an account that the terminal logs in to a game application. A user may control, through different trigger operations, the first virtual object to move, jump, drive, pick, perform skill releasing, and the like, and the trigger operations may be implemented through trigger gestures, voice instructions, control operations for function controls, and the like. In a possible implementation, the user controls a virtual joystick to move the first virtual object in the virtual environment. The first virtual object is a virtual object corresponding to a currently login account of the application.

[0052] Exemplarily, as shown in FIG. 4, a virtual environment picture and a virtual joystick 402 are displayed in a virtual environment interface, the virtual environment picture includes a first virtual object 401, and the virtual joystick 402 is located at a lower left corner of the virtual environment interface. When the terminal does not receive a trigger operation on the virtual joystick 402, and the first virtual object is not in an automatic following state, the virtual joystick 402 is displayed in a default state. The virtual environment interface shown in FIG. 4 further includes a map display control, a skill releasing control, an ordinary attack control, and other controls, and the virtual environment picture further includes other virtual objects in addition to the first virtual object, and virtual things such as roads and plants in a certain range.

[0053] Step 302. Display, in response to a moving operation on the first virtual object, one or more candidate object identifiers corresponding to one or more candidate virtual objects respectively in the virtual environment, the candidate virtual objects being located in an orientation region of the first virtual object.

[0054] An orientation is a direction in which a terminal controls a virtual object to move in a virtual environment, and the orientation region is a region formed along an orientation of the first virtual object. The user performs a control operation on the virtual joystick, the terminal controls, based on the control operation, the first virtual object to move in the virtual environment, and a moving direction is a direction indicated by the control operation. Exemplarily, the control operation may be a drag operation on the virtual joystick, a trigger operation on a direction control in the virtual joystick, or the like.

[0055] Optionally, the moving operation on the first virtual object is a touch operation on a virtual joystick in a virtual environment interface, or a control operation on a movement joystick in a control device (for example, a game handle) connected to the terminal, or the like. This is not limited in this embodiment of this disclosure.

[0056] For example, the terminal determines an orientation of the first virtual object based on the control operation on the virtual joystick, detects virtual objects in the orientation region in a process of controlling the first virtual object to move along the orientation, and displays candidate object identifiers corresponding to candidate virtual objects when the candidate virtual objects exist in the orientation region. The terminal updates the candidate object identifiers in real time during the control operation, so that the user can observe the candidate object identifiers to learn about the candidate virtual objects existing in the orientation region while controlling the first virtual object to move.

[0057] Optionally, the candidate virtual object and the first virtual object belong to the same camp, or the candidate virtual object and the first virtual object belong to different camps, or the candidate virtual object may be a virtual object belonging to the same camp as the first virtual object or a virtual object belonging to different camps from the first virtual object, that is, the candidate virtual objects include a virtual object of the same camp and a virtual object of other camps that meet a preset condition.

[0058] Optionally, at least one of a model thumbnail of the candidate virtual object and an account identifier of an account corresponding to the candidate virtual object is displayed on the candidate object identifier.

[0059] Exemplarily, as shown in FIG. 5, after receiving a control operation on a virtual joystick 501, the terminal determines that a candidate virtual object 503a and a candidate virtual object 503b exist in an orientation region corresponding to a first virtual object 502, and displays a candidate object identifier 504a corresponding to the candidate virtual object 503a and a candidate

object identifier 504b corresponding to the candidate virtual object 503b outside the virtual joystick 501.

[0060] Optionally, after receiving a moving operation on the first virtual object, the terminal determines the candidate virtual objects based on the moving operation. The terminal reports
5 control information (for example, an orientation of the first virtual object) corresponding to the control operation to a server, and the server is configured to determine the orientation region based on the orientation of the first virtual object and position information of virtual objects in the virtual environment, determine the candidate virtual objects based on the orientation region, and feed back the candidate virtual objects to the terminal; and the process of determining the candidate virtual
10 objects may alternatively be collaboratively completed by the terminal and the server. For example, the terminal determines the orientation of the first virtual object based on the moving operation on the first virtual object, and reports related information of the candidate virtual objects and the control operation to the server after determining the candidate virtual objects in the orientation region; and the server verifies the candidate virtual objects based on the control operation and the
15 position information of the virtual objects in the virtual environment, and sends feedback information indicating that the verification succeeds to the terminal after the verification succeeds, so that the terminal continues to perform subsequent operations. This is not limited in this embodiment of this disclosure.

[0061] Step 303. Control, in response to a trigger operation on a target object identifier in
20 the candidate object identifiers, the first virtual object to automatically follow a target virtual object in the virtual environment, the target virtual object being a virtual object corresponding to the target object identifier.

[0062] The target object identifier is an identifier of the trigger operation received in the candidate object identifiers displayed in the terminal. When receiving the trigger operation on the
25 target object identifier, the terminal controls the first virtual object to automatically follow the target virtual object in the virtual environment. For example, the terminal controls the first virtual object to move toward the target virtual object, to reduce or maintain a distance between the first virtual object and the target virtual object. When the target virtual object moves in a direction, the terminal automatically controls the first virtual object to move toward the target virtual object. When the
30 target virtual object stops at a position, and the distance between the first virtual object and the target virtual object is less than or equal to a preset distance, the terminal controls the first virtual object to stop at a current position; or when the distance between the first virtual object and the target virtual object is greater than the preset distance, the terminal still controls the first virtual

object to move toward the target virtual object, until the distance between the first virtual object and the target virtual object is equal to the preset distance.

[0063] Optionally, the trigger operation includes a touch operation (for example, a click operation, a press operation, or a slide operation) on the candidate object identifier in the virtual environment interface, or the trigger operation is implemented by a trigger operation on a corresponding key in a control device. The trigger operation on the target object identifier and the moving operation on the first virtual object are two independent operations, or the trigger operation on the target object identifier is a continuation of the moving operation on the first virtual object, for example, the moving operation on the first virtual object is a drag operation on the virtual joystick. When the drag operation continues to a candidate object identifier and ends, a candidate virtual object corresponding to the candidate object identifier is determined as the target virtual object, thereby improving convenience of the trigger operation, so that the user can quickly switch from a control operation on the virtual joystick to the trigger operation on the target object identifier. This is not limited in this embodiment of this disclosure.

[0064] Exemplarily, as shown in FIG. 5, when receiving a trigger operation on a candidate object identifier 504a, the terminal determines the candidate object identifier 504a as the target object identifier, determines a candidate virtual object 503a as the target virtual object, and controls the first virtual object 502 to automatically follow the target virtual object in the virtual environment.

[0065] In conclusion, in the embodiments of this disclosure, candidate object identifiers are displayed when a moving operation on a first virtual object is received, so that a user can clearly determine virtual objects in an orientation region of the first virtual object through the candidate object identifiers when controlling the first virtual object to move, and select a target virtual object from candidate virtual objects through a trigger operation. Therefore, the first virtual object automatically follows the target virtual object, and there is no need to select a following object from a virtual object list and perform a following operation, thereby simplifying a user operation, and improving efficiency of making the first virtual object enter an automatic following state. The candidate virtual objects are determined based on an orientation of the first virtual object, so that the determined candidate virtual objects meet actual operation needs of the user better, and validity of the candidate object identifiers are improved.

[0066] In a possible implementation, the terminal determines the orientation region based on the position of the first virtual object and the orientation, and determines the candidate virtual objects from the orientation region. FIG. 6 is a flowchart of a method for controlling a virtual object

according to another exemplary embodiment of this disclosure. In this embodiment, an example in which the method is performed by a terminal supporting a virtual environment is used for description. The method includes the following steps 601 to 607.

[0067] Step 601. Display a first virtual object in a virtual environment.

5 **[0068]** For the specific implementation of step 601, reference may be made to step 301, and details are not described again in this embodiment of this disclosure.

[0069] Step 602. In response to a moving operation on the first virtual object, determine an orientation of the first virtual object based on a moving direction indicated by the moving operation.

[0070] In a possible implementation, a user controls the moving direction of the first virtual
10 object through the moving operation on the first virtual object, and the terminal determines the orientation of the first virtual object based on the moving direction indicated by the moving operation, and controls the first virtual object to move along the orientation.

[0071] Optionally, the moving operation on the first virtual object is a touch operation on a virtual joystick in a virtual environment interface, or a control operation on a movement joystick in
15 a control device (for example, a game handle) connected to the terminal, or the like. This is not limited in this embodiment of this disclosure.

[0072] Exemplarily, the virtual joystick is a rotatable control, and is provided with a joystick direction, and the user changes the joystick direction by rotating the virtual joystick, so that the terminal determines the orientation of the first virtual object based on the joystick direction after
20 rotation; or, the virtual joystick includes a movable joystick portion and an immovable wheel portion, and the user controls movement of the joystick portion in the wheel portion, so that the terminal determines the orientation of the first virtual object. For example, a straight line in which a center of the joystick portion and a center of the wheel portion are located is parallel to a straight line in which the orientation of the first virtual object is located, and the orientation is a direction
25 from the center of the wheel portion to the center of the joystick portion.

[0073] Exemplarily, as shown in FIG. 7, the user controls a moving direction of a first virtual object 702 by dragging a joystick portion in a virtual joystick 701, and the terminal determines, based on the moving direction, an orientation of the first virtual object 702, namely, a direction indicated by a ray 703.

30 **[0074]** Step 603. Determine an orientation region based on a position of the first virtual object in the virtual environment and the orientation.

[0075] In a possible implementation, the terminal determines the orientation region based on the position of the first virtual object in the virtual environment and the orientation, and determines candidate virtual objects from the orientation region, so that the user may know the candidate virtual objects in the orientation region of the first virtual object in real time when controlling the first virtual object to move. Step 603 includes the following steps 603a and 603b.

[0076] Step 603a. Determine an orientation framing region based on the position of the first virtual object and the orientation, the orientation framing region being a rectangular region formed along the orientation, a width of the rectangular region being a preset width, the first virtual object being located at a shorter side of the rectangular region, and a length of the rectangular region being a distance from the first virtual object to a virtual environment boundary in a direction of the orientation.

[0077] In a possible implementation, the terminal determines the orientation framing region based on the orientation of the first virtual object, and the orientation framing region is used for selecting the orientation region from a virtual environment picture. The terminal takes a line segment which passes through the position of the first virtual object and is perpendicular to the orientation as a shorter side of the orientation framing region, and takes a line segment which is parallel to the orientation and whose length is equal to the distance from the first virtual object to the virtual environment boundary in the direction of the orientation as a longer side, to determine the rectangular orientation framing region. A length of the shorter side is a preset length, a length of the longer side is equal to the distance from the first virtual object to the virtual environment boundary in the direction of the orientation (namely, a distance from the position of the first virtual object to an intersection of the orientation and the virtual environment boundary). Optionally, the position of the first virtual object is a middle point of the shorter side of the orientation framing region, or the position of the first virtual object is any position located in the shorter side of the orientation framing region.

[0078] Exemplarily, as shown in FIG. 7, the terminal takes a line segment 704 in which a position of the first virtual object 702 is located as a shorter side, and takes a line segment 705 parallel to the ray 703 in which the orientation is located as a longer side, to obtain an orientation framing region 706. A length of the line segment 705 is equal to a distance from the position of the first virtual object 702 to a virtual environment boundary 707 along an orientation, and a length of the line segment 704 is the preset length.

[0079] Optionally, after receiving the moving operation on the first virtual object, the terminal determines the orientation region based on the control operation; the terminal reports

control information (for example, the orientation of the first virtual object) corresponding to the moving operation to a server, and the server is configured to determine the orientation region based on the orientation of the first virtual object and position information, and feed back position information of the orientation region to the terminal; and the process of determining the orientation region may alternatively be collaboratively completed by the terminal and the server. For example, the terminal reports the position information of the orientation region and information about the position and orientation of the first virtual object to the server after determining the orientation region based on the position the first virtual object and the orientation; and the server verifies, based on the position of the first virtual object, and the orientation, the orientation region determined by the terminal, and sends feedback information indicating that the verification succeeds to the terminal after the verification succeeds, so that the terminal continues to perform subsequent operations. This is not limited in this embodiment of this disclosure.

[0080] Step 603b. Perform a Boolean operation on the orientation framing region and a virtual environment picture, to obtain the orientation region, the orientation region being an intersection region of the orientation framing region and the virtual environment picture.

[0081] The virtual environment picture displayed by the terminal is a part of the virtual environment. To facilitate the user's observation and prevent the first virtual object from automatically following virtual objects outside the virtual environment picture causing that the user cannot predict a travel direction and determine a following object, the orientation region in this embodiment of this disclosure is a region in the virtual environment picture currently displayed by the terminal, that is, candidate virtual objects are all virtual objects in the virtual environment picture. In a possible implementation, to ensure that the orientation region is located in the virtual environment picture displayed by the terminal, the terminal performs the Boolean operation on the orientation framing region and the virtual environment picture, to obtain the orientation region. That is, the orientation region is the intersection region of the orientation framing region and the virtual environment picture.

[0082] Exemplarily, as shown in FIG. 7, the terminal performs the Boolean operation on the orientation framing region 706 and a virtual environment picture 708 displayed by the terminal, to obtain an orientation region 709 of the first virtual object 702.

[0083] Optionally, after determining the orientation region, the terminal displays the orientation region in the virtual environment picture (for example, highlights the orientation region), so that the user knows a range of the current orientation region and adjusts the moving direction of the first virtual object according to an object that the user intends to make the first virtual object

follow, to change the orientation region and make the object that the user intends to follow be in the orientation region.

[0084] Step 604. Determine candidate virtual objects from virtual objects in the orientation region.

5 **[0085]** The terminal determines the candidate virtual objects based on the orientation indicated by the moving operation on the first virtual object, and displays object identifiers corresponding to the candidate virtual objects, so that the user knows in real time virtual objects existing in a virtual environment in the front of the first virtual object. In addition, the orientation region is a region formed along the current orientation of the first virtual object, and the user is
10 more likely to control the first virtual object to move to the region, and select the candidate virtual objects from the orientation region, which meets automatic following needs of the user better.

[0086] Optionally, the terminal determines the virtual objects in the orientation region as the candidate virtual objects, or to avoid the interference of the candidate object identifiers displayed by the terminal to the user operation when there are a large quantity of virtual objects in the orientation
15 region, the terminal selects the candidate virtual objects from the virtual objects in the orientation region. In a possible implementation, step 604 includes the following steps 604a and 604b.

[0087] Step 604a. Determine, for each of the virtual objects in the orientation region, a distance between the first virtual object and the virtual object.

[0088] In a possible implementation, the terminal determines the candidate virtual objects
20 based on the distance between the first virtual object and each virtual object in the orientation region. After determining the orientation region, the terminal determines the distance between the first virtual object and each virtual object in the orientation region based on a position of the virtual object in the orientation region and the position of the first virtual object.

[0089] Step 604b. Determine k virtual objects closest to the first virtual object as the
25 candidate virtual objects, k being a positive integer.

[0090] In this embodiment of this disclosure, considering that the user usually intends to follow a virtual object nearby the first virtual object, the terminal determines the k virtual objects closest to the first virtual object as the candidate virtual objects. This ensures accuracy of selecting the candidate virtual objects and also controls a quantity of candidate virtual objects, to prevent
30 excessive candidate object identifiers displayed in the virtual environment picture from affecting the user operation. If a quantity of virtual objects in the orientation region is less than k, the terminal determines all virtual objects in the orientation region as the candidate virtual objects. In another

possible implementation, the terminal may alternatively determine k virtual objects farthest away from the first virtual object in the orientation region as the candidate virtual objects.

[0091] As shown in FIG. 7, the terminal determines that four virtual objects, namely, a teammate A, a teammate B, an enemy A, and an enemy D exist in the orientation region 709, and the terminal determines three virtual objects closest to the first virtual object 702 as the candidate virtual objects, namely, the teammate A, the teammate B, and the enemy A.

[0092] Optionally, after determining the orientation region, the terminal determines the candidate virtual objects based on the orientation region; the terminal reports position information of the orientation region and position information of the first virtual object to a server, and the server is configured to determine the candidate virtual objects, and feed back the candidate virtual objects to the terminal; and the process of determining the candidate virtual objects may alternatively be collaboratively completed by the terminal and the server. For example, the terminal reports the candidate virtual objects to the server after determining the candidate virtual objects based on the position of the first virtual object and the orientation region; and the server verifies, based on the position of the first virtual object and a position of the orientation region, the candidate virtual objects determined by the terminal, and sends feedback information indicating that the verification succeeds to the terminal after the verification succeeds, so that the terminal continues to perform subsequent operations. This is not limited in this embodiment of this disclosure.

[0093] Step 605. Display candidate object identifiers corresponding to the candidate virtual objects.

[0094] In a possible implementation, the moving operation on the first virtual object is a control operation on a virtual joystick in a virtual environment interface. The virtual joystick is a control configured to control the first virtual object to move in the virtual environment, and the user controls the orientation of the first virtual object by controlling a joystick direction of the virtual joystick and makes the first virtual object move along the orientation.

[0095] Optionally, the virtual joystick displayed in the virtual environment interface has a label used for indicating the orientation (for example, an arrow pointing to the orientation); and/or the virtual joystick includes a joystick portion and a wheel portion. The terminal movably displays the joystick portion in the wheel portion based on a touch position of the user, and a direction of a connecting line between a center of the joystick portion and a center of the wheel portion is consistent with the orientation of the first virtual object. A style of the virtual joystick is not limited in this embodiment of this disclosure.

[0096] When the user controls, through the virtual joystick, the first virtual object to move, to facilitate the user to rapidly control automatic following and simplify an operation procedure of the automatic following, the terminal displays the candidate object identifiers nearby the virtual joystick, so that the user completes selection of a target object identifier through the control operation continuously on the virtual joystick. Step 605 further includes the following steps 605a to 605c.

[0097] Step 605a. Determine a display orientation of the candidate object identifiers relative to the virtual joystick based on a moving direction indicated by the control operation, the virtual joystick and the candidate object identifiers being located on a same straight line, and a direction from the virtual joystick to the candidate object identifiers being consistent with the moving direction.

[0098] When the terminal displays the candidate object identifiers, to facilitate the user to know current candidate virtual objects in real time, the terminal displays the candidate object identifiers nearby the virtual joystick, and the terminal enables the display orientation of the candidate object identifiers relative to the virtual joystick to be consistent with the moving direction according to the moving direction indicated by the control operation. As shown in FIG. 7, the terminal determines a display orientation of candidate object identifiers 710a, 710b, and 710c based on the moving direction indicated by the control operation, so that the candidate object identifiers 710a, 710b, and 710c and a center of the virtual joystick 701 are located in a same straight line, and a direction of the straight line is consistent with the moving direction.

[0099] Step 605b. Determine, for each of the candidate object identifiers at the display orientation, a display position of the candidate object identifier based on a distance between the first virtual object and the candidate virtual object corresponding to the candidate object identifier, the distance between the first virtual object and the candidate virtual object corresponding to the candidate object identifier being in a positive correlation with a distance between the virtual joystick and the candidate object identifier.

[0100] Step 605c. Display the candidate object identifiers according to the display positions.

[0101] To further simplify a user operation, and improve efficiency of controlling the first virtual object to perform automatic following, the terminal determines the distance between the candidate object identifier and the virtual joystick based on the distance between the first virtual object and the candidate virtual object, and the distance between the first virtual object and the candidate virtual object is in a positive correlation with the distance between the virtual joystick and the candidate object identifier, so that the display position of the candidate object identifier can

reflect a rough position relationship between the candidate virtual and the first virtual object, to help the user rapidly determine, by observing the virtual joystick and the candidate object identifiers, a target virtual object that needs to follow, and there is no need to view a specific position of each candidate virtual object in the orientation region.

5 **[0102]** As shown in FIG. 7, in the orientation region, if candidate virtual objects having distances from the first virtual object 702 from small to large are respectively the teammate A, the teammate B, and the enemy A, the terminal determines display positions of the candidate object identifiers at the display orientation for display according to a position relationship in which a distance between the candidate object identifier 710a corresponding to the teammate A and the
10 virtual joystick 701 is shortest and a distance between the candidate object identifier 710c corresponding to the enemy A and the virtual joystick 701 is longest.

[0103] In a possible implementation, the candidate virtual object and the first virtual object may belong to different camps, and different candidate virtual objects may belong to different camps. Because the user may need to control the first virtual object to automatically follow a virtual
15 object of a specified camp, the terminal may further need to display the candidate object identifier at the display position based on a camp to which the candidate virtual object belongs. The process of displaying the candidate object identifiers in step 605 further includes the following steps 605d and 605e.

[0104] Step 605d. Determine, for each of the candidate virtual objects, a candidate camp to
20 which the candidate virtual object belongs.

[0105] For example, as shown in FIG. 7, in the candidate virtual objects, the teammate A and the teammate B belong to a camp of the first virtual object 702, and the enemy A belongs to an opposing camp.

[0106] Step 605e. Display the candidate object identifier corresponding to the candidate
25 virtual object according to a display manner corresponding to the candidate camp, display manners corresponding to different camps being different.

[0107] In a possible implementation, a correspondence between a camp and the preset display manner is preset in an application, and the terminal determines the preset display manner corresponding to the candidate object identifier based on the candidate camp, and displays the
30 candidate object identifier at the display position according to the preset display manner.

[0108] For example, at least one of a color and a shape of the candidate object identifiers corresponding to the candidate virtual objects of different camps is different.

[0109] In this embodiment of this disclosure, the terminal selects followable objects based on the orientation of the first virtual object and the orientations of the virtual objects, and is not limited by types of the virtual objects, and the terminal can control the first virtual object to automatically follow a virtual object of a different camp. Compared with a manner of selecting a following object from a camp list of a camp to which the first virtual object belongs in the related technology, on the one hand, a user operation on automatic following is simplified; and on the other hand, types of the candidate virtual objects are not limited, and the user can automatically follow virtual objects of the same camp and an opposing camp, which meets different following needs of the user (for example, the user intends to follow a virtual object of the same camp to jointly complete a task, or intends to follow a virtual object of the opposing camp for battle).

[0110] Step 606. Control, in response to a trigger operation on a target object identifier in the candidate object identifiers, the first virtual object to automatically follow a target virtual object in the virtual environment, the target virtual object being a virtual object corresponding to the target object identifier.

[0111] For the specific implementation of step 606, reference may be made to step 303, and details are not described again in this embodiment of this disclosure.

[0112] Step 607. Cancel automatic following of the first virtual object to the target virtual object in response to the target virtual object leaving the orientation region or in response to the control operation on the first virtual joystick.

[0113] In a process that the first virtual object automatically follows the target virtual object, the target virtual object may be defeated, or rapidly moves to another region in a preset manner, to leave the orientation region. In this case, the terminal cancels the automatic following of the first virtual object to the target virtual object.

[0114] Optionally, after canceling the automatic following of the first virtual object to the target virtual object, the terminal automatically determines the candidate virtual objects; or after the terminal cancels the automatic following of the first virtual object to the target virtual object, the virtual joystick is restored to a default state and the candidate object identifiers are not displayed, until the control operation on the virtual joystick is received again.

[0115] In an automatic following state, if the user triggers the virtual joystick, it indicates that the user intends to control the first virtual object to move toward another direction, and does not continue to follow the target virtual object. Therefore, in another possible implementation, when the first virtual object is in the automatic following state, the user may further trigger the virtual joystick again, so that the terminal cancels the automatic following of the first virtual object to the

target virtual object. For example, the terminal performs the control operation on the virtual joystick again, to cancel the automatic following of the first virtual object to the target virtual object, and consequently the terminal redetermines a candidate virtual object based on the control operation and updates a candidate object identifier. As shown in FIG. 8, when a first virtual object 802 is in a state of automatically following an enemy A, if receiving a control operation on a virtual joystick 801 again, the terminal cancels the automatic following state of the first virtual object 802, determines an enemy C in an orientation region 803 as a candidate virtual object based on a moving direction indicated by the current control operation, and displays a candidate object identifier 804 based on the moving direction.

[0116] After the target virtual object leaves the orientation region, the terminal automatically cancels the automatic following, which can prompt the user that the virtual object is not in the orientation region, so that the user continues to control the first virtual object to move, or search for a more suitable automatic following object, and the user does not need to manually cancel the automatic following. When the user actively triggers the virtual joystick (or triggers the moving operation on the first virtual object in another manner), the terminal cancels the automatic following similarly, and combines the cancel operation on the automatic following with an operation procedure of the moving operation, to further simplify a user operation.

[0117] In this embodiment of this disclosure, a terminal determines an orientation region based on a position and an orientation of a first virtual object, and determines candidate virtual objects from the orientation region. When the terminal displays candidate object identifiers, a relative orientation of the candidate object identifiers and a virtual joystick is consistent with the orientation of the first virtual object, and the distance between the candidate object identifier and the virtual joystick is proportional to a distance between the first virtual object and the candidate virtual object, so that the user can rapidly learn about candidate virtual objects in the orientation region of the first virtual object and respective position relationships by observing the virtual joystick and the candidate object identifiers, and for candidate virtual objects of different camps, the candidate object identifiers are displayed in different preset display manners, to facilitate the user to rapidly select the candidate virtual objects according to needs, thereby improving efficiency of controlling the first virtual object to perform automatic following.

[0118] FIG. 9 is a flowchart of a method for controlling a virtual object according to another exemplary embodiment of this disclosure. In this embodiment, an example in which the method is applied to a terminal supporting a virtual environment is used for description. The method includes the following steps 901 to 908.

[0119] Step 901. Display a first virtual object located in a virtual environment.

[0120] For the specific implementation of step 901, reference may be made to step 301, and details are not described again in this embodiment of this disclosure.

[0121] Step 902. Display candidate object identifiers of candidate virtual objects in the virtual environment in response to a drag operation on a virtual joystick, the candidate virtual objects being located in an orientation region of the first virtual object.

[0122] In a possible implementation, a control operation on the virtual joystick is the drag operation, and the user may control a moving direction through the drag operation, for example, drag the virtual joystick in a preset interface range, or drag a movable joystick portion in a fixed wheel of the virtual joystick. In addition, display positions of the candidate object identifiers change with a drag direction of the drag operation in the virtual joystick.

[0123] Step 903. Display, in response to a drag position of the drag operation being located at the candidate object identifier, a followable label at the candidate virtual object corresponding to the drag position and/or at a position identifier of the candidate virtual object corresponding to the drag position in a map display control.

[0124] In a possible implementation, the user may move the drag position of the drag operation to a candidate object identifier, and control the first virtual object to automatically follow a virtual object corresponding to the candidate object identifier after raising a hand. When the drag operation on the virtual joystick continues from the virtual joystick outward to the candidate object identifier (the drag operation does not end in this case), the terminal displays a followable label at a corresponding candidate virtual object based on the candidate object identifier corresponding to the drag position, and/or displays the followable label at a position identifier of the corresponding candidate virtual object in the map display control, so that the user clearly determines the candidate virtual object corresponding to the current drag position, that is, the first virtual object can automatically follow the object after the hand raises. The user can determine, according to the identifier, whether to end the drag operation or change an end position of the drag operation, to facilitate the user to operate while observing, thereby improving the visibility and convenience of the automatic following operation.

[0125] Exemplarily, as shown in FIG. 10, when receiving a drag operation on a virtual joystick 1001, the terminal displays candidate object identifiers based on a moving direction indicated by the drag operation. When the terminal detects that a drag position of the drag operation is located at a candidate object identifier 1002 corresponding to a candidate virtual object 1003, the terminal displays a followable label 1004 above the candidate virtual object 1003, and displays the

followable label 1004 at a position identifier corresponding to the candidate virtual object 1003 in a map control 1005.

[0126] Step 904. Determine, in response to the drag operation ending, a candidate object identifier at which a drag end point of the drag operation is located at the target object identifier, and determining that the trigger operation on the target object identifier is received.

[0127] In a possible implementation, the user performs the drag operation on the virtual joystick, and continues the drag operation to a target object identifier and raises a hand, so that the terminal determines that the trigger operation on the target object identifier is received (that is, when the user performs the drag operation on the virtual joystick and raises the hand to end the drag operation at a candidate object identifier, the terminal determines the candidate object identifier as the target object identifier, and determines that the trigger operation on the target object identifier is received). As shown in FIG. 10, if the user moves the drag position of the drag operation to the candidate object identifier 1002 and raises the hand, the terminal determines the candidate object identifier 1002 as the target object identifier.

[0128] The user may perform the drag operation on the virtual joystick, so that the terminal displays the candidate object identifier while controlling the first virtual object to move. If the user needs to control the first virtual object to automatically follow the candidate virtual object, the drag operation ends at the target object identifier. The moving operation on the first virtual object and an operation procedure of the automatic following operation are combined, and the moving operation is continued, to trigger the automatic following operation, thereby further improving the convenience of controlling the first virtual object.

[0129] Step 905. Control the first virtual object to automatically follow the target virtual object in the virtual environment.

[0130] For the specific implementation of step 905, reference may be made to step 303, and details are not described again in this embodiment of this disclosure.

[0131] Step 906. Display prompt information in the virtual joystick in response to the first virtual object being in an automatic following state, the prompt information being used for indicating that the first virtual object is in the automatic following state.

[0132] In a possible implementation, in a process that the first virtual object automatically follows the target virtual object, the terminal displays the prompt information in the virtual joystick, to facilitate the user to determine that the first virtual object is in the automatic following state. Optionally, the prompt information is dynamically displayed in the virtual joystick, and a display

position of the prompt information changes with the orientation of the first virtual object, so that the user rapidly knows the moving direction of the first virtual object through movement of the prompt information, and there is no need to search a virtual environment picture for the first virtual object or the target virtual object.

5 **[0133]** Exemplarily, as shown in FIG. 11, a first virtual object 1102 is automatically following a target virtual object 1103, and in this case, prompt information of "following" is displayed in a virtual joystick 1101. When canceling an automatic following state of the first virtual object, the terminal also cancels display of the prompt information. A direction from a center of the virtual joystick 1101 to a center of a display region of the prompt information is consistent with an
10 orientation of the first virtual object 1102, that is, the prompt information moves with the first virtual object, thereby enhancing a prompt effect of automatic following, and facilitating the user to observe a change situation of the moving direction.

[0134] Step 907. Update a display state of the virtual joystick based on the moving direction of the first virtual object, a joystick direction of the virtual joystick after the display state is updated
15 being consistent with the moving direction.

[0135] In a possible implementation, to further prompt the user the moving direction of the first virtual object during automatic following, the terminal updates the joystick direction of the virtual joystick in real time, so that the virtual joystick can dynamically display the moving direction of the first virtual object.

20 **[0136]** As shown in FIG. 11, if the first virtual object 1102 is automatically following the target virtual object 1103, a joystick direction of the virtual joystick 1101 is consistent with a moving direction of the first virtual object 1102. That is, the terminal updates, according to the moving direction, a label used for indicating the joystick direction and a display position of a movable joystick portion in the virtual joystick 1101. As shown in the figure, when the target virtual
25 object 1103 moves from an upper right corner of a virtual environment picture to a position directly above the virtual environment picture, the moving direction of the first virtual object 1102 also changes, and the terminal updates a display state of the virtual joystick 1101 based on the moving direction of the first virtual object 1102.

[0137] Step 908. Display, in response to the first virtual object being in an automatic
30 following state, automatic following connecting lines between the first virtual object and the target virtual object and/or between a position identifier of the first virtual object and a position identifier of the target virtual object in a map display control.

[0138] To further prompt the user that the first virtual object is in the automatic following state, and also indicate the target virtual object followed by the first virtual object and the moving direction of the first virtual object, the terminal displays the automatic following connecting lines between the first virtual object and the target virtual object in the virtual environment and/or between the position identifier of the first virtual object and the position identifier of the target virtual object in the map display control. The movement of the first virtual object is not controlled by the user in the automatic following state, and it is difficult for the user to predict a movement path of the target virtual object, while the automatic following connecting line is more obvious than other display content in the virtual environment. Therefore, the user can rapidly position positions of the first virtual object and the target virtual object and the moving direction according to the automatic following connecting line. When canceling the automatic following of the first virtual object to the target virtual object, the terminal cancels the automatic following connecting line.

[0139] As shown in FIG.11, the first virtual object 1102 is automatically following the target virtual object 1103, and the terminal displays an automatic following connecting line 1104 between the first virtual object 1102 and the target virtual object 1103, and displays an automatic following connecting line 1105 between a position identifier of the first virtual object 1102 and a position identifier of the target virtual object 1103 in a map display control 1105.

[0140] In this embodiment of this disclosure, when a drag position of a drag operation on a virtual joystick is moved to candidate object identifiers, followable labels are displayed at corresponding candidate virtual objects, to facilitate a user to clearly determine positions of the candidate virtual objects. The user may perform the drag operation on the virtual joystick, so that a terminal displays the candidate object identifiers while controlling a first virtual object to move. If the user needs to control the first virtual object to automatically follow a candidate virtual object, the drag operation ends at a target object identifier. The operation is simple and rapid, and the user operation is further simplified. In a process that the first virtual object is in an automatic following state, an automatic following connecting line is displayed, and a joystick direction of the virtual joystick is updated based on a moving direction of the first virtual object, to facilitate the user to timely learn about the moving direction of the first virtual object and a position of the target virtual object.

[0141] With reference to the embodiments, in an exemplary example, a procedure of controlling a virtual object is shown in FIG. 12.

[0142] Step 1201. Receive a drag operation on a virtual joystick.

[0143] Step 1202. Determine whether candidate virtual objects exist in an orientation region. In a positive determination, perform step 1203.

[0144] Step 1203. Display candidate object identifiers.

[0145] Step 1204. Determine whether an end position of the drag operation is located at the candidate object identifier. In a positive determination, perform step 1205. In a negative determination, return to step 1203.

[0146] Step 1205. Determine a target object identifier, and control a first virtual object to automatically follow the target virtual object.

[0147] Step 1206. Determine whether the target virtual object leaves an orientation range. In a positive determination, perform step 1208. In a negative determination, perform step 1207.

[0148] Step 1207. Determine whether to receive a control operation on the virtual joystick again. In a positive determination, perform step 1208; In a negative determination, return to step 1205.

[0149] Step 1208. Cancel automatic following of the first virtual object to the target virtual object.

[0150] FIG. 13 is a structural block diagram of an apparatus for controlling a virtual object according to an exemplary embodiment of this disclosure. The apparatus includes: a first display module 1301, configured to display a first virtual object located in a virtual environment; a second display module 1302, configured to display, in response to a moving operation on the first virtual object, one or more candidate object identifiers corresponding to one or more candidate virtual objects respectively in the virtual environment, the candidate virtual objects being located in an orientation region of the first virtual object; and a first control module 1303, configured to control, in response to a trigger operation on a target object identifier in the candidate object identifiers, the first virtual object to automatically follow a target virtual object in the virtual environment, the target virtual object being a virtual object corresponding to the target object identifier.

[0151] Optionally, the second display module 1302 includes: a first determining unit, configured to determine an orientation of the first virtual object based on a moving direction indicated by the moving operation; a second determining unit, configured to determine the orientation region based on a position of the first virtual object in the virtual environment and the orientation; a third determining unit, configured to determine the candidate virtual objects from virtual objects in the orientation region; and a first display unit, configured to display the candidate object identifiers corresponding to the candidate virtual objects.

[0152] Optionally, the third determining unit is further configured to: determine, for each of the virtual objects in the orientation region, a distance between the first virtual object and the virtual object; and determine k virtual objects closest to the first virtual object as the candidate virtual objects, k being a positive integer.

5 **[0153]** Optionally, the moving operation includes a control operation on a virtual joystick in a virtual environment interface; and the first display unit is further configured to: determine a display orientation of the candidate object identifiers relative to the virtual joystick based on a moving direction indicated by the control operation, the virtual joystick and the candidate object identifiers being located on a same straight line, and a direction from the virtual joystick to the
10 candidate object identifiers being consistent with the moving direction; determine, for each of the candidate object identifiers at the display orientation, a display position of the candidate object identifier based on a distance between the first virtual object and the candidate virtual object corresponding to the candidate object identifier, the distance between the first virtual object and the candidate virtual object corresponding to the candidate object identifier being in a positive
15 correlation with a distance between the virtual joystick and the candidate object identifier; and display the candidate object identifiers according to the display positions.

[0154] Optionally, the second determining unit is further configured to: determine an orientation framing region based on the position of the first virtual object and the orientation, the orientation framing region being a rectangular region formed along the orientation, a width of the
20 rectangular region being a preset width, the first virtual object being located at a shorter side of the rectangular region, and a length of the rectangular region being a distance from the first virtual object to a virtual environment boundary in a direction of the orientation; and perform a Boolean operation on the orientation framing region and a virtual environment picture, to obtain the orientation region, the orientation region being an intersection region of the orientation framing
25 region and the virtual environment picture.

[0155] Optionally, the second display module 1302 further includes: a fourth determining unit, configured to determine, for each of the candidate virtual objects, a candidate camp to which the candidate virtual object belongs; and a second display unit, configured to display the candidate object identifier corresponding to the candidate virtual object according to a display manner
30 corresponding to the candidate camp, display manners corresponding to different camps being different.

[0156] Optionally, the moving operation is a drag operation on the virtual joystick in the virtual environment interface; and the first control module 1303 includes: a fifth determining unit,

configured to determine, in response to the drag operation ending, a candidate object identifier at which a drag end point of the drag operation is located as the target object identifier, and determining that the trigger operation on the target object identifier is received; and a control unit, configured to control the first virtual object to automatically follow the target virtual object in the virtual environment.

[0157] Optionally, the apparatus further includes a third display module, configured to display, in response to a drag position of the drag operation being located at the candidate object identifier, a followable label at the candidate virtual object corresponding to the drag position and/or at a position identifier of the candidate virtual object corresponding to the drag position in a map display control.

[0158] Optionally, the moving operation includes a control operation on a virtual joystick in a virtual environment interface; and the apparatus further includes: a fourth display module, configured to display prompt information in the virtual joystick in response to the first virtual object being in an automatic following state, the prompt information being used for indicating that the first virtual object is in the automatic following state; and a fifth display module, configured to update a display state of the virtual joystick based on a moving direction of the first virtual object, a joystick direction of the virtual joystick after the display state is updated being consistent with the moving direction.

[0159] Optionally, the apparatus further includes a sixth display module, configured to display, in response to the first virtual object being in an automatic following state, automatic following connecting lines between the first virtual object and the target virtual object and/or between a position identifier of the first virtual object and a position identifier of the target virtual object in a map display control.

[0160] Optionally, the apparatus further includes a second control module, configured to cancel automatic following of the first virtual object to the target virtual object in response to the target virtual object leaving the orientation region or in response to the control operation on the virtual joystick.

[0161] In conclusion, in the embodiments of this disclosure, candidate object identifiers are displayed when a control operation on a virtual joystick is received, so that a user can clearly determine virtual objects in an orientation region of the first virtual object through the candidate object identifiers when controlling the first virtual object to move, and select a target virtual object from candidate virtual objects through a simple operation. Therefore, the first virtual object automatically follows the target virtual object, and there is no need to select a following object from

a virtual object list and perform a following operation, thereby simplifying a user operation, and improving efficiency of making the first virtual object enter an automatic following state. The candidate virtual objects are determined based on an orientation of the first virtual object, so that the determined candidate virtual objects meet actual operation needs of the user better, and validity of the candidate object identifiers are improved.

[0162] FIG. 14 is a structural block diagram of a terminal 1400 according to an exemplary embodiment of this disclosure. The terminal 1400 may be a portable mobile terminal, for example, a smartphone, a tablet computer, an MP3, or an MP4 player. The terminal 1400 may also be referred to as other names such as user equipment and a portable terminal.

[0163] Generally, the terminal 1400 includes a processor 1401 and a memory 1402.

[0164] The processor 1401 may include one or more processing cores, for example, a 4-core processor or an 8-core processor. The processor 1401 may be implemented by using at least one hardware form of a digital signal processor (DSP), a field-programmable gate array (FPGA), and a programmable logic array (PLA). The processor 1401 may alternatively include a main processor and a coprocessor. The main processor is a processor configured to process data in an active state, also referred to as a central processing unit (CPU). The coprocessor is a low-power processor configured to process data in a standby state. In some embodiments, the processor 1401 may be integrated with a graphics processing unit (GPU). The GPU is configured to render and draw content that needs to be displayed on a display screen. In some embodiments, the processor 1401 may further include an artificial intelligence (AI) processor. The AI processor is configured to process a computing operation related to machine learning.

[0165] The memory 1402 may include one or more computer-readable storage media. The computer-readable storage medium may be tangible and non-transient. The memory 1402 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-transient computer-readable storage medium in the memory 1402 is configured to store at least one instruction, the at least one instruction being configured to be executed by the processor 1401 to implement the method provided in the embodiments of this disclosure.

[0166] A person skilled in the art may understand that a structure shown in FIG. 14 constitutes no limitation on the terminal 1400, and the terminal may include more or fewer components than those shown in the figure, or some components may be combined, or a different component deployment may be used.

[0167] An embodiment of this disclosure further provides a computer-readable storage medium, storing at least one instruction, the at least one instruction being loaded and executed by a processor to implement the method for controlling a virtual object described in the foregoing embodiments.

5 **[0168]** According to an aspect of this disclosure, a computer program product or a computer program is provided, including computer instructions, the computer instructions being 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, so that the computer device performs the method for controlling a virtual object provided in the
10 optional implementations in the foregoing aspect.

CLAIMS

What is claimed is:

1. A method for controlling a virtual object, the method being applicable to a terminal, the method comprising:

5 displaying a first virtual object located in a virtual environment;

displaying, in response to a moving operation on the first virtual object, one or more candidate object identifiers corresponding to one or more candidate virtual objects respectively in the virtual environment, the candidate virtual objects being located in an orientation region of the first virtual object; and

10 controlling, in response to a trigger operation on a target object identifier in the candidate object identifiers, the first virtual object to automatically follow a target virtual object in the virtual environment, the target virtual object being a virtual object corresponding to the target object identifier.

2. The method according to claim 1, wherein the displaying, in response to a moving operation
15 on the first virtual object, one or more candidate object identifiers corresponding to one or more candidate virtual objects respectively in the virtual environment comprises:

determining, in response to the moving operation on the first virtual object, an orientation of the first virtual object based on a moving direction indicated by the moving operation;

20 determining the orientation region based on a position of the first virtual object in the virtual environment and the orientation;

determining the candidate virtual objects from virtual objects in the orientation region; and

displaying the candidate object identifiers corresponding to the candidate virtual objects.

3. The method according to claim 2, wherein the determining the candidate virtual objects from virtual objects in the orientation region comprises:

25 determining, for each of the virtual objects in the orientation region, a distance between the first virtual object and the virtual object; and

determining k virtual objects closest to the first virtual object as the candidate virtual objects, k being a positive integer.

4. The method according to claim 3, wherein the moving operation comprises a control
30 operation on a virtual joystick in a virtual environment interface; and the displaying the candidate

object identifiers corresponding to the candidate virtual objects comprises:

determining a display orientation of the candidate object identifiers relative to the virtual joystick based on a moving direction indicated by the control operation, the virtual joystick and the candidate object identifiers being located on a same straight line, and a direction from the virtual joystick to the candidate object identifiers being consistent with the moving direction;

determining, for each of the candidate object identifiers at the display orientation, a display position of the candidate object identifier based on a distance between the first virtual object and the candidate virtual object corresponding to the candidate object identifier, the distance between the first virtual object and the candidate virtual object corresponding to the candidate object identifier being in a positive correlation with a distance between the virtual joystick and the candidate object identifier; and

displaying the candidate object identifiers according to the display positions.

5. The method according to any one of claims 2 to 4, wherein the determining the orientation region based on a position of the first virtual object in the virtual environment and the orientation comprises:

determining an orientation framing region based on the position of the first virtual object and the orientation, the orientation framing region being a rectangular region formed along the orientation, a width of the rectangular region being a preset width, the first virtual object being located at a shorter side of the rectangular region, and a length of the rectangular region being a distance from the first virtual object to a virtual environment boundary in a direction of the orientation; and

performing a Boolean operation on the orientation framing region and a virtual environment picture, to obtain the orientation region, the orientation region being an intersection region of the orientation framing region and the virtual environment picture.

6. The method according to any one of claims 1 to 4, wherein the displaying one or more candidate object identifiers corresponding to one or more candidate virtual objects respectively in the virtual environment comprises:

determining, for each of the candidate virtual objects, a candidate camp to which the candidate virtual object belongs; and

displaying the candidate object identifier corresponding to the candidate virtual object according to a display manner corresponding to the candidate camp, display manners corresponding to different camps being different.

7. The method according to any one of claims 1 to 4, wherein the moving operation is a drag operation on a virtual joystick in a virtual environment interface; and

the controlling, in response to a trigger operation on a target object identifier in the candidate object identifiers, the first virtual object to automatically follow a target virtual object in the virtual environment comprises:

determining, in response to the drag operation ending, a candidate object identifier at which a drag end point of the drag operation is located as the target object identifier, and determining that the trigger operation on the target object identifier is received; and

controlling the first virtual object to automatically follow the target virtual object in the virtual environment.

8. The method according to claim 7, wherein before the controlling, in response to a trigger operation on a target object identifier in the candidate object identifiers, the first virtual object to automatically follow a target virtual object in the virtual environment, the method further comprises:

displaying, in response to a drag position of the drag operation being located at the candidate object identifier, a followable label at the candidate virtual object corresponding to the drag position and/or at a position identifier of the candidate virtual object corresponding to the drag position in a map display control.

9. The method according to any one of claims 1 to 4, wherein the moving operation comprises a control operation on a virtual joystick in a virtual environment interface; and after the controlling, in response to a trigger operation on a target object identifier in the candidate object identifiers, the first virtual object to automatically follow a target virtual object in the virtual environment, the method further comprises:

displaying prompt information in the virtual joystick in response to the first virtual object being in an automatic following state, the prompt information being used for indicating that the first virtual object is in the automatic following state; and

updating a display state of the virtual joystick based on a moving direction of the first virtual object, a joystick direction of the virtual joystick after the display state is updated being consistent with the moving direction.

10. The method according to any one of claims 1 to 4, wherein after the controlling, in response to a trigger operation on a target object identifier in the candidate object identifiers, the first virtual object to automatically follow a target virtual object in the virtual environment, the

method further comprises:

displaying, in response to the first virtual object being in an automatic following state, automatic following connecting lines between the first virtual object and the target virtual object and/or between a position identifier of the first virtual object and a position identifier of the target virtual object in a map display control.

11. The method according to any one of claims 1 to 4, wherein after the controlling, in response to a trigger operation on a target object identifier in the candidate object identifiers, the first virtual object to automatically follow a target virtual object in the virtual environment, the method further comprises:

canceling automatic following of the first virtual object to the target virtual object in response to the target virtual object leaving the orientation region or in response to the moving operation on the first virtual object.

12. An apparatus for controlling a virtual object, comprising:

a first display module, configured to display a first virtual object located in a virtual environment;

a second display module, configured to display, in response to a moving operation on the first virtual object, one or more candidate object identifiers corresponding to one or more candidate virtual objects respectively in the virtual environment, the candidate virtual objects being located in an orientation region of the first virtual object; and

a first control module, configured to control, in response to a trigger operation on a target object identifier in the candidate object identifiers, the first virtual object to automatically follow a target virtual object in the virtual environment, the target virtual object being a virtual object corresponding to the target object identifier.

13. A terminal, comprising a processor and a memory, the memory storing at least one instruction, at least one program, a code set or 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 the processor to implement the method for controlling a virtual object according to any one of claims 1 to 11.

14. A computer-readable storage medium, storing at least one computer program, the computer program being loaded and executed by a processor to implement the method for controlling a virtual object according to any one of claims 1 to 11.

15. A computer program product, comprising computer instructions, the computer instructions being stored in a computer-readable storage medium, a processor of a terminal reading the computer

instructions from the computer-readable storage medium, and the processor executing the computer instructions to implement the method for controlling a virtual object according to any one of claims 1 to 11.