

METHOD FOR CONTROLLING VIRTUAL OBJECT, AND RELATED APPARATUS

RELATED APPLICATION

[0001] This application claims priority to Chinese Patent Application No. 201910812631.6, entitled "VIRTUAL OBJECT CONTROL METHOD AND APPARATUS, TERMINAL, AND STORAGE MEDIUM" and filed with the National Intellectual Property Administration, PRC on August 30, 2019, which is incorporated herein by reference in its entirety.

FIELD OF THE TECHNOLOGY

[0002] Embodiments of this application relate to the field of computer and Internet technologies, and in particular, to control of a virtual object.

BACKGROUND OF THE DISCLOSURE

[0003] Regarding certain shooter games (STGs), at a mobile side, a player may control a virtual object in a game scene provided in a game battle, for example, may control the virtual object to move.

[0004] According to certain technologies, a user interface (UI) of an STG is provided with a button used for controlling a virtual object to move. A user may control, by operating the button, the virtual object to move, for example, control the virtual object to move in a direction in a virtual scene.

SUMMARY

[0005] Embodiments of this application provide a virtual object control method and a related apparatus, to solve the technical problem of complex and inefficient operating when controlling a virtual object to move continuously in the related art. The technical solutions are as follows:

[0006] According to an aspect, an embodiment of this application provides a virtual object control method, including:

[0007] displaying a UI, the UI including a joystick control configured to control a virtual object to move;

[0008] receiving a quick click/tap operation signal acting on a target region corresponding to the joystick control; and

[0009] controlling, according to the quick click/tap operation signal, the virtual object to run in a virtual scene displayed in the UI.

[0010] According to another aspect, an embodiment of this application provides virtual object control method, including:

[0011] displaying a UI, the UI including a joystick control configured to control a virtual object to move;

[0012] controlling, in response to the virtual object being in a non-standing state and receiving a swipe operation signal whose initial position is at the joystick control, the virtual object to switch from the non-standing state to a standing state; and

[0013] controlling, after the virtual object is switched to the standing state, the virtual object to run in a virtual scene displayed in the UI.

[0014] According to another aspect, an embodiment of this application provides a virtual object control apparatus, including:

[0015] an interface display module, configured to display a UI, the UI including a joystick control configured to control a virtual object to move;

[0016] a signal receiving module, configured to receive a quick click/tap operation signal acting on a target region corresponding to the joystick control; and

[0017] a running control module, configured to control, according to the quick click/tap operation signal, the virtual object to run in a virtual scene displayed in the UI.

[0018] According to another aspect, an embodiment of this application provides a virtual object control apparatus, including:

[0019] an interface display module, configured to display a UI, the UI including a joystick control configured to control a virtual object to move;

[0020] a posture switching module, configured to control, in response to the virtual object being in a non-standing state and receiving a swipe operation signal whose initial position is at the joystick control, the virtual object to switch from the non-standing state to a standing state; and

[0021] a running control module, configured to control, after the virtual object is switched to the standing state, the virtual object to run in a virtual scene displayed in the UI.

[0022] According to still another aspect, an embodiment of this application provides a mobile 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 foregoing virtual object control method.

[0023] According to yet another aspect, an embodiment of this application provides a non-transitory computer-readable storage medium, configured to store a computer program, the computer program, when run on a computing device, causing the computing device to perform the virtual object control method according to the foregoing aspects.

[0024] According to still yet another aspect, an embodiment of this application provides a computer program product, comprising instructions, the instructions, when run on a computing device, causing the computing device to perform the foregoing virtual object control methods.

[0025] According to the technical solutions provided in the embodiments of this application, a joystick control is displayed in a UI; and in response to receiving a quick click/tap operation signal acting on a target region corresponding to the joystick control, a virtual object is controlled to automatically run in a virtual environment displayed in the UI, triggering a virtual object to automatically run with one key without requiring the user to continuously click/tap or long press an operation control, thereby improving the operation efficiency.

BRIEF DESCRIPTION OF THE DRAWINGS

[0026] To describe the technical solutions in the embodiments of this application more clearly, the following briefly introduces the accompanying drawings required for describing the embodiments. Apparently, the accompanying drawings in the following description show only some embodiments of this application, and a person of ordinary skill in the art may still derive other accompanying drawings from these accompanying drawings without creative efforts.

[0027] FIG. 1 is a schematic diagram of an implementation environment according to an embodiment of this application.

[0028] FIG. 2 is a schematic structural diagram of a terminal according to an embodiment of this application.

[0029] FIG. 3 is a flowchart of a virtual object control method according to an embodiment of this application.

[0030] FIG. 4 is an exemplary schematic diagram of a UI.

[0031] FIG. 5 is a flowchart of a virtual object control method according to another embodiment of this application.

[0032] FIG. 6 is a flowchart of a virtual object control method according to another embodiment of this application.

[0033] FIG. 7 is an exemplary schematic diagram of another UI.

[0034] FIG. 8 is a flowchart of a swipe operation signal control method according to an embodiment of this application.

[0035] FIG. 9 is a flowchart of a virtual object control method according to yet another embodiment of this application.

[0036] FIG. 10 is a schematic diagram of yet another UI according to this application.

[0037] FIG. 11 is a schematic diagram of still yet another UI according to this application.

[0038] FIG. 12 is a flowchart of a virtual object control method according to yet another embodiment of this application.

[0039] FIG. 13 is a block diagram of a virtual object control apparatus according to an embodiment of this application.

[0040] FIG. 14 is a block diagram of a virtual object control apparatus according to another embodiment of this application.

[0041] FIG. 15 is a block diagram of a virtual object control apparatus according to yet another embodiment of this application.

[0042] FIG. 16 is a block diagram of a virtual object control apparatus according to still yet another embodiment of this application.

[0043] FIG. 17 is a structural block diagram of a mobile terminal according to an embodiment of this application.

DESCRIPTION OF EMBODIMENTS

[0044] To make objectives, technical solutions, and advantages of this application clearer, the following further describes implementations of this application in detail with reference to the accompanying drawings.

[0045] In certain embodiments, a virtual scene refers to a scene displayed (or provided) when a client of an application (such as a game application) runs on a terminal. The virtual scene is a scene created for a virtual object to perform activities (such as game competition). The virtual

scene may be, for example, a virtual house, a virtual island, or a virtual map. The virtual scene may be a simulated scene of the real world, or may be a semi-simulated semi-fictional scene, or may be an entirely fictional scene. The virtual scene may be a two-dimensional virtual scene, a 2.5-dimensional virtual scene, or a three-dimensional virtual scene, which is not limited in the embodiments of this application.

[0046] In certain embodiments, a virtual object refers to a virtual character controlled by a user account in an application. For example, the application is a game application. The virtual object is a game character controlled by the user account in the game application. The virtual object may be in a human form or animal, cartoon, or other forms, which is not limited in the embodiments of this application. The virtual object may be presented in a three-dimensional form or a two-dimensional form, which is not limited in the embodiments of this application.

[0047] In different game applications, operations that can be performed by the virtual object controlled by the user account may be different. For example, in an STG application, the user account may control the virtual object to perform operations such as shooting, running, jumping, gun picking up, gun replacing, and bullet loading.

[0048] In certain embodiments, in addition to the game applications, the virtual object may also be presented to a user and provided with a corresponding function in applications of another type, for example, an augmented reality (AR) application, a social application, or an interactive entertainment application, which is not limited in the embodiments of this application. Besides, the form and corresponding function of the virtual object vary with different applications, and may be preset according to an actual requirement, which is not limited in the embodiments of this application.

[0049] FIG. 1 is a schematic diagram of an implementation environment according to an embodiment of this application. The implementation environment may include a mobile terminal 10 and a server 20.

[0050] The mobile terminal 10 may be a portable electronic device, such as a mobile phone, a tablet computer, a game console, an ebook reader, a multimedia playback device, and a wearable device. A client of a game application, such as a client of an STG application, may be installed in the mobile terminal 10.

[0051] The server 20 is configured to provide a backend service to the client of the application (for example, a game application) in the mobile terminal 10. For example, the server 20

may be a backend server of the application (for example, the game application). The server 20 may be one server, a server cluster including a plurality of servers, or a cloud computing service center.

[0052] The mobile terminal 10 and the server 20 may communicate with each other through a network 30. The network 30 may be a wired network or a wireless network.

[0053] Steps in method embodiments of this application may be performed by the mobile terminal. FIG. 2 is a schematic structural diagram of a mobile terminal according to an embodiment of this application. The mobile terminal 10 may include a mainboard 110, an external output/input device 120, a memory 130, an external interface 140, a touch system 150, and a power supply 160.

[0054] Processing elements such as a processor and a controller are integrated in the mainboard 110.

[0055] The external output/input device 120 may include a display component (for example, a display screen), a sound playback component (for example, a speaker), a sound collecting component (for example, a microphone), and various buttons.

[0056] The memory 130 stores program code and data.

[0057] The external interface 140 may include an earphone interface, a charging interface, a data interface, and the like.

[0058] The touch system 150 may be integrated in the display component or the buttons of the external output/input device 120, and the touch system 150 is configured to detect touch operations performed by a user on the display component or the buttons.

[0059] The power supply 160 is configured to supply power to other components in the mobile terminal 10.

[0060] In this embodiment of this application, the processor in the mainboard 110 may generate a UI (for example, a game interface) by executing or calling the program code and the data stored in the memory, and present the generated UI (for example, the game interface) by using the external output/input device 120. During presentation of the UI (for example, the game interface), a touch operation performed during interaction between the user and the UI (for example, the game interface) may be detected through the touch system 150, and a response is made to the touch operation.

[0061] FIG. 3 is a flowchart of a virtual object control method according to an embodiment of this application. The method is applicable to the mobile terminal described above, such as a

client of an application (for example, an STG application) in the mobile terminal. The method may include the following steps:

[0062] Step 301: Display a UI, the UI including a joystick control configured to control a virtual object to move.

[0063] Using an STG application as an example, the UI may be a display interface of a game battle. The UI is configured to present a virtual environment of the game battle to a user. For example, the UI may include an element, such as a virtual building, a virtual prop, or a virtual object, in the virtual environment. In certain embodiments, the UI further includes some operation controls, such as buttons, sliders, and icons, for the user to operate.

[0064] In this embodiment of this application, as shown in FIG. 4, a UI 40 includes a joystick control 41, and the joystick control 41 is an operation control configured to control a virtual object to move. No display parameter of the joystick control 41 is limited in this embodiment of this application, that is, none of a shape, a size, a position, a style, and the like of the joystick control 41 is limited. For example, the joystick control 41 may be round or square. An area of the joystick control 41 may be 1/10 of an area of the UI 40, or may be 1/20 of the area of the UI 40. The joystick control 41 may be located at a lower left corner of the UI 40, or may be located at a lower right corner of the UI 40. The joystick control 41 may be displayed in a transparent form, or may be displayed in a color filled form. In this embodiment of this application, the display of the joystick control 41 does not block main display elements of the UI 40.

[0065] In certain embodiments, the UI includes a first view layer and a second view layer. A display level of the first view layer is higher than a display level of the second view layer. The joystick control is located at the first view layer, and a game picture used for displaying the virtual environment of the game battle is located at the second view layer. In certain embodiments, in addition to the joystick control described above, the first view layer may further include other operation controls, such as an operation control for controlling a posture of the virtual object, and an operation control for controlling virtual equipment assembled by the virtual object. This is not limited in this embodiment of this application.

[0066] Step 302: Receive a quick click/tap operation signal acting on a target region corresponding to the joystick control.

[0067] The target region corresponding to the joystick control is a region that overlaps with the joystick control. In the target region, the mobile terminal can respond to a touch operation of the

user, for example, a click/tap or press operation. After the user performs the click/tap or press operation acting on the target region, the mobile terminal receives a corresponding operation signal.

[0068] In an example, the target region corresponding to the joystick control and the joystick control totally overlap with each other, that is, a size and shape of the target region corresponding to the joystick control are totally the same as a size and shape of the joystick control, and a center position of the joystick control and a center position of the target region overlap with each other.

[0069] In another example, the target region corresponding to the joystick control includes the joystick control, and a size of the target region corresponding to the joystick control is greater than a size of the joystick control. For example, as shown in FIG. 4, a target region 42 is a rectangular region with a size greater than a size of the joystick control 41. In certain embodiments, a center position of the joystick control and a center position of the target region overlap with each other.

[0070] In still another example, a size of the target region corresponding to the joystick control is less than a size of the joystick control. In certain embodiments, a center position of the joystick control and a center position of the target region overlap with each other.

[0071] In addition, the target region may be a region visible to the user in the UI, or may be a region invisible to the user in the UI. For example, the target region may be a completely transparent region. This is not limited in this embodiment of this application.

[0072] In this embodiment of this application, the quick click/tap operation signal is a signal generated by triggering of a plurality of consecutive click/tap operations, and in the plurality of consecutive click/tap operations, a time interval between two adjacent click/tap operations is less than a preset threshold. In certain embodiments, the quick click/tap operation signal may be a double-click/tap operation signal. The double-click/tap operation signal is an operation signal of two consecutive clicks/taps between which a time interval is less than a preset threshold. In certain embodiments, in other examples, the quick click/tap operation signal may be a triple-click/tap operation signal, a quadra-click/tap operation signal, or the like. This is not limited in this embodiment of this application.

[0073] Step 303: Control, according to the quick click/tap operation signal, the virtual object to run in a virtual scene displayed in the UI.

[0074] In an example, when the quick click/tap operation signal is received, if the virtual object is in a standing state, the virtual object is controlled to automatically run in the virtual scene in the standing state.

[0075] In another example, when the quick click/tap operation signal is received, if the virtual object is in a non-standing state, the virtual object is controlled to automatically run in the virtual scene in the non-standing state.

[0076] In certain embodiments, to provide diversified virtual object control manners to the user and improve the man-machine interaction experience, after the foregoing step 303, the method further includes: receiving a posture switching instruction corresponding to the virtual object. The posture switching instruction is an operation instruction for controlling a posture of a virtual object to switch. The posture switching instruction may be triggered by the user through an operation control, a voice, or a gesture. If the virtual object is in a non-standing state, according to the posture switching instruction, the virtual object is controlled to switch from a non-standing state to a standing state, and then the virtual object is controlled to automatically run in the virtual scene in the standing state. If the virtual object is in a standing state, according to the posture switching instruction, the virtual object is controlled to switch from a standing state to a non-standing state, and then the virtual object is controlled to automatically run in the virtual scene in the non-standing state. In certain embodiments, referring to FIG. 4, a posture switching icon 43 is displayed in the UI. When the user clicks/taps the posture switching icon 43, the mobile terminal receives a posture switching instruction. In certain embodiments, the posture switching icon 43 may be a squatting icon.

[0077] In certain embodiments, to get closer to a running manner of an object in an actual environment, a speed when automatically running in a non-standing state is lower than a speed when automatically running in a standing state. For example, when the virtual object automatically runs in a virtual scene in a squatting state, a speed when automatically running is 1 m/s in this situation. If the mobile terminal receives a posture switching instruction, the virtual object is controlled to switch from the squatting state to a standing state, and the virtual object is controlled to automatically run in the virtual scene in a standing state. In this situation, a speed when automatically running is 3 m/s.

[0078] In certain embodiments, to improve the man-machine interaction experience and to help the user learn a state of the virtual object conveniently, in response to the virtual object being in a non-standing state, first prompt information is displayed in a UI, the first prompt information being used for indicating that the virtual object is in the non-standing state; and in response to the

virtual object being in a standing state, second prompt information is displayed in the UI, the second prompt information being used for indicating that the virtual object is in the standing state. In certain embodiments, the first prompt information and the second prompt information may be displayed based on the same icon. For example, as shown in FIG. 4, both the first prompt information and the second prompt information are displayed based on the posture switching icon 43 in the UI. The posture switching icon 43 is used for indicating whether the virtual object is in a standing state or a squatting state. When the virtual object is in a squatting state, the first prompt information is displayed, for example, the posture switching icon 43 is highlighted. When the virtual object is in a standing state, the second prompt information is displayed, for example, the posture switching icon 43 is not highlighted.

[0079] In addition, in this embodiment of this application, a standing state is a state in which a virtual object stands, and a non-standing state is a state in which a virtual object does not stand. In a possible implementation, the non-standing state may be a squatting state, that is, a state in which the virtual object squats. In this embodiment of this application, a direction and speed of the virtual object when automatically running may be preset. For example, a direction of the virtual object when automatically running may be preset to a front direction, and a speed of the virtual object when automatically running may be preset to 3 m/s, the front direction being a direction that the virtual object is facing.

[0080] In summary, according to the technical solution provided in this embodiment of this application, a joystick control is displayed in a UI; and in response to receiving a quick click/tap operation signal acting on a target region corresponding to the joystick control, a virtual object is controlled to automatically run in a virtual environment displayed in the UI, triggering a virtual object to automatically run with one key without requiring the user to continuously click/tap or long press an operation control, thereby improving the operation efficiency. Moreover, after the user triggers the virtual object to start to automatically run, a finger of the user may be released, and then some other operations are performed with the released finger, for example, observing a virtual environment during running, changing equipment during running, and communicating with other users during running, thereby bringing more diversified interaction functions.

[0081] In addition, before receiving the quick click/tap operation signal acting on the target region, if the virtual object is in a non-standing state, after the virtual object is triggered to start to automatically run in a non-standing state, the virtual object may be further controlled, through a posture switching instruction, to switch from the non-standing state to a standing state, and then the virtual object is controlled to automatically run in the standing state, thereby providing diversified

manners of controlling the virtual object to automatically run to the user, and further improving the man-machine interaction experience.

[0082] In a possible implementation, to improve the man-machine interaction experience and the flexibility of controlling the virtual object to automatically run, the foregoing step 303 includes: obtaining an attribute value corresponding to the quick click/tap operation signal; determining a running speed of the virtual object according to the attribute value; and controlling the virtual object to automatically run in the virtual scene according to the running speed.

[0083] In this embodiment of this application, the attribute value is an operation parameter corresponding to the quick click/tap operation signal. For example, the attribute value may be an operation time interval or a quantity of operations. This is not limited in this embodiment of this application.

[0084] For example, when the quick click/tap operation signal is a double-click/tap operation signal, the attribute value is a click/tap time interval of the double-click/tap operation signal. In certain embodiments, the click/tap time interval of the double-click/tap operation signal has a negative correlation with the running speed of the virtual object. That is, a shorter click/tap time interval indicates a higher running speed, and a longer click/tap time interval indicates a lower running speed.

[0085] In certain embodiments, the attribute value of the quick click/tap operation signal may alternatively be a quantity of clicks/taps. In certain embodiments, the quantity of clicks/taps of the quick click/tap operation signal has a positive correlation with the running speed of the virtual object. That is, a larger quantity of clicks/taps indicates a higher running speed, and a smaller quantity of clicks/taps indicates a lower running speed.

[0086] In another possible implementation, to further improve the man-machine interaction experience and the flexibility of controlling the virtual object to automatically run, the foregoing step 303 includes that the mobile terminal may detect an operation position of the quick click/tap operation signal; determine a running direction of the virtual object according to the operation position; and control the virtual object to automatically run in the virtual scene according to the running direction. For example, the target region is evenly divided into four regions, which are recorded as a region 1, a region 2, a region 3, and a region 4. A direction corresponding to the region 1 is east, a direction corresponding to the region 2 is south, a direction corresponding to the region 3 is west, and a direction corresponding to the region 4 is north. In response to detecting the operation position of the quick click/tap operation signal to be in the region 2, that is, the running direction of the virtual object is determined as south, the virtual object is controlled to automatically

run in the virtual scene according to a south direction. In this embodiment of this application, neither a quantity of regions into which the target region is specifically divided, nor a direction corresponding to each region is limited.

[0087] In actual implementation, the foregoing two manners may be combined, to control a running speed and a running direction of the virtual object simultaneously.

[0088] In still another possible implementation, referring to FIG. 5, a virtual object control method provided in an embodiment of this application may include the following steps:

[0089] Step 501: Determine whether a virtual object is in a standing state. Step 502 is performed if the virtual object is in a standing state; and step 501 continues to be performed if the virtual object is not in a standing state.

[0090] Step 502: Determine whether a quick click/tap operation signal is received. Step 503 is performed if a quick click/tap operation signal is received; and step 502 continues to be performed if a quick click/tap operation signal is not received.

[0091] Step 503: Determine whether the quick click/tap operation signal is located in a target region. Step 504 is performed if the quick click/tap operation signal is located in a target region; and step 503 continues to be performed if the quick click/tap operation signal is not located in a target region.

[0092] Step 504: Determine a running speed and a running direction of the virtual object according to an attribute value and an operation position of the quick click/tap operation signal.

[0093] Step 505: Control the virtual object to run according to the running speed and the running direction.

[0094] In summary, according to the technical solution provided in this embodiment of this application, a running speed and a running direction when a virtual object automatically runs are determined according to an attribute value and an operation position of a quick click/tap operation signal, so as to provide more virtual object control manners to a user, and to better satisfy operation requirements of the user.

[0095] FIG. 6 is a flowchart of a virtual object control method according to another embodiment of this application. The method is applicable to the mobile terminal described above, such as a client of an application (for example, an STG application) in the mobile terminal. The method may include the following steps:

[0096] Step 601: Display a UI, the UI including a joystick control configured to control a virtual object to move.

[0100] Step 602. Control, in response to the virtual object being in a non-standing state and receiving a swipe operation signal whose initial position is at the joystick control, the virtual object to switch from the non-standing state to a standing state.

[0101] In an exemplary embodiment, before the controlling the virtual object to switch from the non-standing state to a standing state, the method further includes: obtaining a touch position of the swipe operation signal; and performing, in response to the touch position of the swipe operation signal being in a display position of a quick stand-up icon, the operation of controlling the virtual object to switch from the non-standing state to a standing state. In certain embodiments, when the touch position of the swipe operation signal is in the display position of the quick stand-up icon, the quick stand-up icon may be highlighted to serve as a prompt that switching to a triggered state succeeds, thereby improving the man-machine interaction efficiency.

[0102] In an exemplary embodiment, after the obtaining a touch position of the swipe operation signal, the method further includes: obtaining a distance between the touch position of the swipe operation signal and a target position; and displaying the quick stand-up icon in the UI in response to detecting the distance to be greater than a distance threshold. The target position is a center position of the joystick control. In this embodiment of this application, a size of the distance threshold is set according to a size of a UI of a used mobile terminal. For example, when the mobile terminal is a tablet computer, the distance threshold is set to 5 cm, and when the mobile terminal is a mobile phone, the distance threshold is set to 3 cm. Through the foregoing settings, the distance threshold may vary flexibly according to different sizes of UIs of used mobile terminals, further improving the man-machine interaction experience.

[0103] In an exemplary embodiment, to provide more virtual object control manners and to further improve the man-machine interaction experience, after the obtaining a distance between the touch position of the swipe operation signal and a target position, the method further includes: determining a moving speed of the virtual object according to the distance, the distance having a negative correlation with the moving speed. That is, a longer distance indicates a lower moving speed, and a shorter distance indicates a higher moving speed. That is, as the touch position of the swipe operation signal gets increasingly close to the target position, the moving speed of the virtual object is increasingly high.

[0104] Step 603: Control, after the virtual object is switched to the standing state, the virtual object to run in a virtual scene displayed in the UI.

[0105] In an exemplary embodiment, as shown in FIG. 7(a), a UI 70 includes a joystick control 71 and a quick stand-up icon 73. The joystick control 71 is configured to control a virtual object to move, and the quick stand-up icon 73 is configured to switch a state of the virtual object and control the virtual object to run in a virtual scene after the state switching. In certain embodiments, the joystick control 71 includes a drag icon 72, and the drag icon 72 is used for changing a moving speed and direction of a virtual object. In certain embodiments, to improve the man-machine interaction experience, the drag icon 72 may change its position in real time according to the swipe operation signal, and position changing of the drag icon 72 reflects position changing of the virtual object. In certain embodiments, a center position of the drag icon coincides with a center position of the touch position of the swipe operation signal. For example, when the swipe operation signal is to quickly swipe to an upper left corner of the UI, the drag icon 72 also quickly moves to the upper left corner of the UI, and a situation that the virtual object quickly moves toward northwest is reflected in the virtual scene displayed in the UI. In addition, when a distance between the center position of the drag icon and the center position of the joystick control is greater than a distance threshold, the quick stand-up icon 73 is displayed in the UI 70. As shown in FIG. 7(b), when the center position of the drag icon 72 is in the display position of the quick stand-up icon 73, the virtual object is controlled to quickly stand up and then automatically run, and the quick stand-up icon 73 is controlled to be highlighted. In certain embodiments, as shown in FIG. 7(a), the UI 70 includes a squatting icon 74, and when the virtual object is in a squatting state, the squatting icon 74 is highlighted. In certain embodiments, as shown in FIG. 7(b), when the center position of the drag icon 72 is in the display position of the quick stand-up icon 73, the virtual object is controlled to quickly stand up and then automatically run, and the squatting icon 74 is not highlighted.

[0106] In summary, according to the technical solution provided in this embodiment of this application, when a virtual object is in a non-standing state, the virtual object is controlled to quickly stand up first and then automatically run, which avoids a situation where after the virtual object in a non-standing state is controlled to run, some operations are further required to control the virtual object to stand up and automatically run, so as to further improve the man-machine interaction efficiency and the man-machine interaction experience.

[0107] In a possible implementation, referring to FIG. 8, a swipe operation signal control method provided in an embodiment of this application may include the following steps:

[0108] Step 801: Receive a swipe operation signal.

[0109] Step 802: Calculate a distance between a center position of the swipe operation signal and a center position of a joystick control.

[0110] Step 803: Display a quick running icon in response to the distance being greater than a distance threshold.

[0111] Step 804: Control the center position of the swipe operation signal to be in a display position of the quick running icon.

[0112] In a possible implementation, referring to FIG. 9, a virtual object control method provided in an embodiment of this application may include the following steps:

[0113] Step 901: Control a virtual object to be in a non-standing state.

[0114] Step 902: Determine whether a swipe operation signal is received. Step 903 is performed if a swipe operation signal is received; and step 902 continues to be performed if a swipe operation signal is not received.

[0115] Step 903: Obtain a distance between a touch position of the swipe operation signal and a target position.

[0116] Step 904: Determine whether the distance is greater than a distance threshold. Step 905 is performed if the distance is greater than a distance threshold; and step 903 is performed if the distance is not greater than a distance threshold.

[0117] Step 905: Display a quick stand-up icon.

[0118] Step 906: Determine whether the touch position of the swipe operation signal is in a display position of the quick stand-up icon. Step 907 is performed if the touch position of the swipe operation signal is in a display position of the quick stand-up icon; and step 906 continues to be performed if the touch position of the swipe operation signal is not in a display position of the quick stand-up icon.

[0119] Step 907: Control the virtual object to switch from the non-standing state to a standing state.

[0120] Step 908: Control the virtual object to run.

[0121] In a possible implementation, the foregoing method may further include the following steps:

[0122] displaying a UI, the UI including an automatic running icon used for controlling a virtual object to run; receiving a triggering signal acting on the automatic running icon; and controlling, according to the triggering signal, the virtual object to run.

[0123] For example, as shown in FIG. 10, an automatic running icon 101 is displayed in a UI 100, and when a triggering signal corresponding to the automatic running icon 101 is received, a virtual object is controlled to run. In certain embodiments, after the virtual object automatically runs, the automatic running icon 101 is highlighted to serve as a prompt that the virtual object is in an automatic running state. In certain embodiments, the triggering signal may be a click/tap signal, or may be a press signal. This is not limited in this embodiment of this application.

[0124] In another possible implementation, the foregoing method may further include the following steps:

[0125] obtaining a touch position of the swipe operation signal in response to receiving a swipe operation signal whose initial position is at the joystick control; displaying an automatic running icon in response to a distance between a center position of the touch position of the swipe operation signal and a center position of the joystick control being greater than a distance threshold; and controlling, in response to the touch position of the swipe operation signal being in a display position of the automatic running icon, the virtual object to run.

[0126] In certain embodiments, the joystick control includes a drag icon used for controlling a virtual object to move, and a center position of the drag icon coincides with the center position of the touch position of the swipe operation signal.

[0127] In certain embodiments, after the virtual object automatically runs, the automatic running icon is highlighted to serve as a prompt that the virtual object is in an automatic running state.

[0128] For example, as shown in FIG. 11(a), a joystick control 111 is displayed in a UI 110, the joystick control 111 includes a drag icon 112, and a position of the drag icon 112 may be changed by receiving a swipe operation signal. As shown in FIG. 11(b), when the center position of the touch position of the swipe operation signal coincides with a center position of the joystick control 111, that is, the center position of the touch position of the swipe operation signal coincides with a center position of the drag icon 112, an automatic running icon 113 is displayed in the UI. When the center position of the drag icon 112 is in a display position of the automatic running icon 113, a virtual object is controlled to run, and the automatic running icon 113 is controlled to be highlighted.

[0129] In a possible implementation, referring to FIG. 12, a virtual object control method provided in an embodiment of this application may include the following steps:

[0130] Step 1201: Control a virtual object to be in a standing state.

[0131] Step 1202: Determine whether a swipe operation signal is received. Step 1203 is performed if a swipe operation signal is received; and step 1202 continues to be performed if a swipe operation signal is not received.

[0132] Step 1103: Obtain a distance between a touch position of the swipe operation signal and a target position.

[0133] Step 1104: Determine whether the distance is greater than a distance threshold. Step 1205 is performed if the distance is greater than a distance threshold; and step 1203 is performed if the distance is not greater than a distance threshold.

[0134] Step 1205: Display an automatic running icon.

[0135] Step 1206: Determine whether the touch position of the swipe operation signal is in a display position of the automatic running icon. Step 1207 is performed if the touch position of the swipe operation signal is in a display position of the automatic running icon; and step 1206 continues to be performed if the touch position of the swipe operation signal is not in a display position of the automatic running icon.

[0136] Step 1207: Control the virtual object to run.

[0137] The following are apparatus embodiments of this application, which can be used to perform the method embodiments of this application. For details not disclosed in the apparatus embodiments of this application, reference may be made to the method embodiments of this application.

[0138] FIG. 13 is a block diagram of a virtual object control apparatus according to an embodiment of this application. The apparatus has functions of implementing the foregoing method embodiments. The functions may be implemented by hardware, or may be implemented by hardware executing corresponding software. The apparatus may be a mobile terminal or set in a mobile terminal. The apparatus 1300 may include: an interface display module 1310, a signal receiving module 1320, and a running control module 1330.

[0139] The interface display module 1310 is configured to display a UI, the UI including a joystick control configured to control a virtual object to move.

[0140] The signal receiving module 1320 is configured to receive a quick click/tap operation signal acting on a target region corresponding to the joystick control.

[0141] The running control module 1330 is configured to control, according to the quick click/tap operation signal, the virtual object to run in a virtual scene displayed in the UI.

[0142] In an exemplary embodiment, as shown in FIG. 14, the signal receiving module 1320 is configured to: receive a quick click/tap operation signal acting on a target region corresponding to the joystick control; or receive a press operation signal acting on a target region corresponding to the joystick control.

[0143] In an exemplary embodiment, as shown in FIG. 14, the running control module 1330 includes a speed control submodule 1331, configured to: obtain an attribute value corresponding to the quick click/tap operation signal; determine a running speed of the virtual object according to the attribute value; and control the virtual object to run in the virtual scene according to the running speed.

[0144] In an exemplary embodiment, as shown in FIG. 14, the running control module 1330 includes a direction control submodule 1332, configured to: detect an operation position corresponding to the quick click/tap operation signal; determine a running direction of the virtual object according to the operation position; and control the virtual object to run in the virtual scene according to the running direction.

[0145] In an exemplary embodiment, the running control module 1330 is further configured to: control, in response to receiving the quick click/tap operation signal and the virtual object being in a non-standing state, the virtual object to run in the virtual scene in the non-standing state.

[0146] In an exemplary embodiment, the running control module 1330 is further configured to: receive a posture switching instruction corresponding to the virtual object; control, according to the posture switching instruction, the virtual object to switch from the non-standing state to a standing state; and control the virtual object to run in the virtual scene in the standing state.

[0147] In summary, according to the technical solution provided in this embodiment of this application, a joystick control is displayed in a UI; and in response to receiving a quick click/tap operation signal acting on a target region corresponding to the joystick control, a virtual object is controlled to run in a virtual environment displayed in the UI, triggering a virtual object to run with one key without requiring the user to continuously click/tap or long press an operation control, thereby improving the operation efficiency. Moreover, after the user triggers the virtual object to start to run, a finger of the user may be released, and then some other operations are performed with

the released finger, for example, observing a virtual environment during running, changing equipment during running, and communicating with other users during running, thereby bringing more diversified interaction functions.

[0148] In addition, before receiving the quick click/tap operation signal acting on the target region, if the virtual object is in a non-standing state, after the virtual object is triggered to start to run in a non-standing state, the virtual object may be further controlled, through a posture switching instruction, to switch from the non-standing state to a standing state, and then the virtual object is controlled to run in the standing state, thereby providing diversified manners of controlling the virtual object to run to the user, and further improving the man-machine interaction experience.

[0149] FIG. 15 is a block diagram of a virtual object control apparatus according to another embodiment of this application. The apparatus has functions of implementing the foregoing method embodiments. The functions may be implemented by hardware, or may be implemented by hardware executing corresponding software. The apparatus may be a mobile terminal or set in a mobile terminal. The apparatus 1500 may include: an interface display module 1510, a posture switching module 1520, and a running control module 1530.

[0150] The interface display module 1510 is configured to display a UI, the UI including a joystick control configured to control a virtual object to move.

[0151] The posture switching module 1520 is configured to control, in response to the virtual object being in a non-standing state and receiving a swipe operation signal whose initial position is at the joystick control, the virtual object to switch from the non-standing state to a standing state.

[0152] The running control module 1530 is configured to control, after the virtual object is switched to the standing state, the virtual object to run in a virtual scene displayed in the UI.

[0153] In an exemplary embodiment, the posture switching module 1520 is further configured to: obtain a touch position of the swipe operation signal; perform, in response to the touch position of the swipe operation signal being in a display position of a quick stand-up icon, the operation of controlling the virtual object to switch from the non-standing state to a standing state.

[0154] In an exemplary embodiment, the posture switching module 1520 is further configured to: obtain a distance between the touch position of the swipe operation signal and a target position; and display the quick stand-up icon in the UI in response to the distance being detected to be greater than a distance threshold.

[0155] In an exemplary embodiment, the posture switching module 1520 is further configured to: determine a moving speed of the virtual object according to the distance, the distance having a negative correlation with the moving speed.

[0156] In an exemplary embodiment, as shown in FIG. 16, the apparatus 1500 further includes an information display module 1540, configured to: display first prompt information in the UI in response to the virtual object being in the non-standing state, the first prompt information being used for indicating that the virtual object is in the non-standing state; and display second prompt information in the UI in response to the virtual object being in the standing state, the second prompt information being used for indicating that the virtual object is in the standing state.

[0157] In summary, according to the technical solution provided in this embodiment of this application, when a virtual object is in a non-standing state, the virtual object is controlled to quickly stand up first and then automatically run, which avoids a situation where after the virtual object in a non-standing state is controlled to run, some operations are further required to control the virtual object to stand up and automatically run, so as to further improve the man-machine interaction efficiency and the man-machine interaction experience.

[0158] When the apparatus provided in the foregoing embodiment implements its functions, a description is given only by using the foregoing division of function modules as an example. In actual implementations, the functions may be allocated to and implemented by different function modules according to the requirements, that is, the internal structure of the device may be divided into different function modules, to implement all or some of the functions described above. In addition, the apparatus and method embodiments provided in the foregoing embodiments belong to the same conception. For the specific implementation process, refer to the method embodiments, so the details are not described herein again.

[0159] FIG. 17 is a structural block diagram of a mobile terminal 1700 according to an embodiment of this application. The mobile terminal 1700 may be a portable electronic device, such as a mobile phone, a tablet computer, a game console, an ebook reader, a multimedia playback device, and a wearable device. The mobile terminal is configured to implement the virtual object control method provided in the foregoing embodiments. The mobile terminal may be the mobile terminal 10 in the implementation environment shown in FIG. 1.

[0160] Generally, the mobile terminal 1700 includes a processor 1701 and a memory 1702.

[0161] The processor 1701 may include one or more processing cores. For example, the processor may be a 4-core processor or an 8-core processor. The processor 1701 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 1701 may alternatively include a main processor and a coprocessor. The main processor is 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, a graphics processing unit (GPU) may be integrated into the processor 1701. The GPU is configured to be responsible for rendering and drawing content to be displayed on a display screen. In some embodiments, the processor 1701 may further include an AI processor. The AI processor is configured to process computing operations related to machine learning.

[0162] The memory 1702 may include one or more computer-readable storage media. The computer-readable storage media may be non-transitory. The memory 1702 may further include a high-speed random access memory (RAM), and a non-volatile memory such as one or more magnetic disk storage devices or flash storage devices. In some embodiments, the non-transitory computer-readable storage medium in the memory 1702 is configured to store 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 configured to be executed by one or more processors to implement the virtual object control method

[0163] In some embodiments, the mobile terminal 1700 may include: a peripheral interface 1703 and at least one peripheral. The processor 1701, the memory 1702, and the peripheral interface 1703 may be connected by a bus or a signal cable. Each peripheral may be connected to the peripheral interface 1703 through a bus, a signal cable, or a circuit board. Specifically, the peripheral includes: at least one of a radio frequency (RF) circuit 1704, a display screen 1705, a camera component 1706, an audio circuit 1707, a positioning component 1408, and a power supply 1709.

[0164] A person skilled in the art may understand that the structure shown in FIG. 17 does not constitute a limitation to the mobile terminal 1700, and the mobile terminal may include more or fewer components than those shown in the figure, or some components may be combined, or a different component arrangement may be used.

[0165] An exemplary embodiment of this application further provides a storage medium, the storage medium being configured to store a computer program, the computer program being configured to perform the virtual object control method.

[0166] In certain embodiments, the computer-readable storage medium may include: a read-only memory (ROM), a RAM, a solid state drive (SSD), an optical disc, or the like. The RAM may include a resistive RAM (ReRAM) and a dynamic RAM (DRAM).

[0167] In an exemplary embodiment, a computer program product is further provided, the computer program product, when run on a terminal device, causing the terminal device to perform the foregoing virtual object control method.

[0168] In certain embodiments, the term "a plurality of" mentioned means two or more. "And/or" describes an association relationship for associated objects and represents that three relationships may exist. For example, A and/or B may represent the following three situations: only A exists, both A and B exist, and only B exists. The character "/" generally indicates an "or" relationship between the associated objects. In addition, the step numbers described in this specification merely exemplarily show a possible execution sequence of the steps. In some other embodiments, the steps may not be performed according to the number sequence. For example, two steps with different numbers may be performed simultaneously, or two steps with different numbers may be performed according to a sequence contrary to the sequence shown in the figure. This is not limited in the embodiments of this application.

[0169] The foregoing descriptions are merely exemplary embodiments of this application, but are not intended to limit this application. Any modification, equivalent replacement, or improvement made within the spirit and principle of this application shall fall within the protection scope of this application.

CLAIMS

What is claimed is:

1. A virtual object control method, performed by a terminal device, the method comprising:
 - displaying a user interface (UI), the UI comprising a joystick control configured to control a virtual object to move;
 - receiving a quick click/tap operation signal acting on a target region corresponding to the joystick control;
 - obtaining an operation time interval corresponding to the quick click/tap operation signal, and determining a first running speed of the virtual object according to the operation time interval, wherein a click/tap time interval of a double-click/tap operation signal has a negative correlation with the first running speed;
 - controlling, in response to receiving the quick click/tap operation signal and the virtual object being in a non-standing state, the virtual object to run in the non-standing state according to the first running speed in a virtual scene displayed in the UI;
 - receiving a posture switching instruction corresponding to the virtual object, and controlling, according to the posture switching instruction, the virtual object to switch from the non-standing state to a standing state; and
 - controlling the virtual object to run in the virtual scene in the standing state according to a second running speed, wherein the first running speed is lower than the second running speed.
2. A virtual object control apparatus, comprising:
 - an interface display module, configured to display a user interface (UI), the UI comprising a joystick control configured to control a virtual object to move;
 - a signal receiving module, configured to receive a quick click/tap operation signal acting on a target region corresponding to the joystick control;
 - a running control module, configured to
 - obtain an operation time interval corresponding to the quick click/tap operation signal, and determine a first running speed of the virtual object according to the operation time

interval, wherein a click/tap time interval of a double-click/tap operation signal has a negative correlation with the first running speed;

control, in response to receiving the quick click/tap operation signal and the virtual object being in a non-standing state, the virtual object to run in the non-standing state according to the first running speed in a virtual scene displayed in the UI;

receive a posture switching instruction corresponding to the virtual object, and control, according to the posture switching instruction, the virtual object to switch from the non-standing state to a standing state; and

control the virtual object to run in the virtual scene in the standing state according to a second running speed, wherein the first running speed is lower than the second running speed.

3. A mobile 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 virtual object control method according to claim 1.

4. A non-transitory computer-readable storage medium, configured to store a computer program, the computer program, when run on a computing device, causing the computing device to perform the virtual object control method according to claim 1.

5. A computer program product, comprising instructions, the instructions, when run on a computing device, causing the computing device to perform the virtual object control method according to claim 1.