

VIRTUAL OBJECT CONTROL METHOD AND APPARATUS, TERMINAL, AND STORAGE MEDIUM

RELATED APPLICATION

[0001] This application claims priority to Chinese Patent Application No. 202010848014.4, entitled "VIRTUAL OBJECT CONTROL METHOD AND APPARATUS, TERMINAL, AND STORAGE MEDIUM", and filed on Friday, August 21, 2020, which is incorporated herein by reference in its entirety.

FIELD OF THE TECHNOLOGY

[0002] This application relates to the field of computer and Internet technologies, and in particular, to a virtual object control method and apparatus, a terminal, and a storage medium.

BACKGROUND OF THE DISCLOSURE

[0003] Currently, skill types of virtual objects in game applications are increasing.

[0004] In the related technology, a user may control, through a skill control in a user interface, a first virtual object to use a skill to initiate an attack on a second virtual object. The second virtual object and the first virtual object are in different camps. Additionally, in a game application, a skill control is displayed in a user interface in the form of a skill icon, and a skill icon represents a unique skill, that is, different skill icons represent different skills. The user may control, through different skill icons, the first virtual object to cast different skills.

[0005] However, in the foregoing related technology, a skill icon represents a unique skill, and when a virtual object has a plurality of skills, the user interface has a limited area. As a result, skill icons have small sizes and close positions, and consequently it is easy to generate a misoperation.

SUMMARY

[0006] Embodiments of this application provide a virtual object control method and apparatus, a terminal, and a storage medium, to reduce a misoperation probability of a user and ensure operation accuracy. The technical solutions are as follows:

[0007] According to an aspect of the embodiments of this application, a virtual object control method is provided, performed by a terminal, the method including:

[0008] displaying a display picture corresponding to a virtual environment and displaying a particular attack control, the particular attack control being used for controlling a first virtual object in the virtual environment to perform at least two different behaviors;

[0009] determining, in response to a touch operation for the particular attack control, a target behavior from the at least two different behaviors according to resources associated with the first virtual object, the resources being consumed items for performing a behavior corresponding to the particular attack control; and

[0010] controlling the first virtual object to perform the target behavior.

[0011] According to an aspect of the embodiments of this application, a virtual object control apparatus is provided, including:

[0012] a control display module, configured to display a display picture corresponding to a virtual environment and to display a particular attack control, the particular attack control being configured for controlling a first virtual object in the virtual environment to perform at least two different behaviors;

[0013] a behavior determining module, configured to determine, in response to a touch operation for the particular attack control, a target behavior from the at least two different behaviors according to resources associated with the first virtual object, the resources being consumed items for performing a behavior corresponding to the particular attack control; and

[0014] a behavior execution module, configured to control the first virtual object to perform the target behavior.

[0015] According to an aspect of the embodiments of this application, a terminal is provided, 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.

[0016] According to an aspect of the embodiments of this application, a computer-readable storage medium is provided, 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 a processor to implement the foregoing virtual object control method.

[0017] According to an aspect of the embodiments of this application, a computer program product or a computer program is provided, the computer program product or the computer program

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 to cause the computer device to perform the foregoing virtual object control method.

[0018] The technical solutions provided in the embodiments of this application may bring the following beneficial effects:

[0019] A first virtual object can be controlled through a particular attack control to perform at least two different behaviors, so that one control controls different behaviors, to avoid interface congestion caused by excessive icons corresponding to operation controls, reduce the misoperation probability of the user, and ensure the operation accuracy; and when behaviors of the first virtual object are increased, no new icon corresponding to the operation controls needs to be added in a user interface, thereby effectively ensuring brevity of the user interface.

[0020] Additionally, when a touch operation for the particular attack control is detected, different behaviors to be performed by the first virtual object are selected from the at least two different behaviors based on resources associated with the first virtual object. That is, when the first virtual object owns different resources, actions of the particular attack control are different, so that the particular attack control is diversified, and the user may control the resources associated with the first virtual object to control the actions of the particular attack control, and then use the particular attack control to implement the different actions, making operations flexible.

BRIEF DESCRIPTION OF THE DRAWINGS

[0021] To describe the technical solutions in the embodiments of this application more clearly, the accompanying drawings required for describing the embodiments are briefly described below. Apparently, the accompanying drawings in the following descriptions show merely some embodiments of this application, and a person of ordinary skill in the art may obtain other accompanying drawings according to these accompanying drawings without creative efforts.

[0022] FIG. 1 is a schematic diagram of an application running environment according to an embodiment of this application.

[0023] FIG. 2 is a flowchart of a virtual object control method according to an embodiment of this application.

[0024] FIG. 3 is a schematic exemplary diagram of a correspondence between quantities and target behaviors.

- [0025]** FIG. 4 is a schematic exemplary diagram of a virtual object control method.
- [0026]** FIG. 5 and FIG. 6 are schematic exemplary diagrams of two other correspondences between quantities and target behaviors.
- [0027]** FIG. 7 is a schematic exemplary diagram of another correspondence between quantities and target behaviors.
- [0028]** FIG. 8 is a schematic exemplary diagram of an operation attribute reminding manner.
- [0029]** FIG. 9 is a flowchart of a virtual object control method according to another embodiment of this application.
- [0030]** FIG. 10 is a block diagram of a virtual object control apparatus according to an embodiment of this application.
- [0031]** FIG. 11 is a block diagram of a virtual object control apparatus according to another embodiment of this application.
- [0032]** FIG. 12 is a structural block diagram of a terminal according to an embodiment of this application.

DESCRIPTION OF EMBODIMENTS

- [0033]** 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.
- [0034]** FIG. 1 is a schematic diagram of an application running environment according to an embodiment of this application. The application running environment may include a terminal 10 and a server 20.
- [0035]** The terminal 10 may be an electronic device such as a mobile phone, a tablet computer, a game console, an ebook reader, a multimedia player, a wearable device, or a personal computer (PC). A client of an application may be installed on the terminal 10. The application may be an application that needs to be downloaded and installed, or may be a click-to-use application, and this is not limited in this embodiment of this application.
- [0036]** In this embodiment of this application, the application may be any application that can provide a virtual environment for a virtual object substituted and operated by a user to perform activities in the virtual environment. Typically, the application is a game application, such as a

multiplayer online battle arena (MOBA) game, a battle royale (BR) game, and a third-person shooting game (TPS), a first-person shooting game (FPS) and a multiplayer gun battle survival game. Certainly, in addition to the game applications, another type of application may also present a virtual object to a user and provide a corresponding function for the virtual object. For example, the foregoing application may be a virtual reality (VR) application, an augmented reality (AR) application, a three-dimensional map application, a military simulation application, a social application, an interactive entertainment application, or the like, and this is not limited in this embodiment 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. This is not limited in the embodiments of this application. Optionally, a client of the foregoing application is run on the terminal 10. In some embodiments, the application is an application developed based on a three-dimensional virtual environment engine. For example, the virtual environment engine is a Unity engine. The virtual environment engine may construct a three-dimensional virtual environment, virtual objects, virtual props, and the like, to bring a more immersive gaming experience to the user.

[0037] The foregoing virtual environment is a scene displayed (or provided) when a client of an application (such as a game application) runs on a terminal. The virtual environment refers to a scene created for a virtual object to perform activities (such as game competition), such as a virtual house, a virtual island, a virtual map, or a virtual building. The virtual environment may be a simulated environment of a real world, or may be a semi-simulated and semi-fictional environment, or may be a completely fictional environment. The virtual environment may be a two-dimensional virtual environment, a 2.5-dimensional virtual environment, or a three-dimensional virtual environment, and this is not limited in this embodiment of this application.

[0038] The foregoing virtual object may be a virtual character controlled by a user account in an application, or may be a virtual character controlled by a computer program in the application. For example, the application is a game application. The virtual object may be a game character controlled by the user account in the game application, or may be a game monster controlled by a computer program in the game application. The virtual object may be in a human form or animal, cartoon, or other forms. This 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. This is not limited in the embodiments of this application. Optionally, when the virtual environment is a three-dimensional virtual environment, the virtual object is a three-dimensional model created based on a skeletal animation technology. Each virtual object has a respective shape and size in the

three-dimensional virtual environment, and occupies some space in the three-dimensional virtual environment.

[0039] In a possible implementation, the foregoing virtual objects include virtual objects in the same camp and virtual objects in different camps, that is, in the application, the virtual objects are divided into different camps. A MOBA game is used as an example. Ten users are divided into two teams of a red team and a blue team to play a game, that is, every five persons form a team. In this case, the red team and virtual objects in the red team are in the same camp and are teammates, the blue team and virtual objects in the blue team are in the same camp and are teammates, and the virtual objects in the red team and the blue team are in different camps and are in a hostile relationship.

[0040] The server 20 is configured to provide a backend service for the client of the application in the terminal 10. For example, the server 20 may be a backend server of the application. The server 20 may be one server, a server cluster including a plurality of servers, or a cloud computing service center. Optionally, the server 20 provides backend services for applications in a plurality of terminals 10.

[0041] Optionally, the terminal 10 may communicate with the server 20 through a network.

[0042] FIG. 2 is a flowchart of a virtual object control method according to an embodiment of this application. The method may be applied to a terminal. For example, an execution entity of each step may be the client of the application installed on the terminal 10 in the application running environment shown in FIG. 1. The method may include the following steps (201 to 203):

[0043] Step 201: Display a display picture corresponding to a virtual environment and display a particular attack control.

[0044] The virtual environment is a scene used for a virtual object to perform an activity. The virtual object may be a virtual character controlled by a user account or may be a virtual character controlled by a computer program. Optionally, the virtual environment includes virtual islands, virtual maps, and virtual buildings. This is not limited in the embodiments of this application. In this embodiment of this application, a user may control a first virtual object to interact with the virtual environment through a user account, such as entering or destroying a virtual building in the virtual environment, and attacking other virtual objects in the virtual environment. The foregoing user account may control one or more virtual objects. This is not limited in the embodiments of this application.

[0045] The display picture is a virtual environment picture presented to the user on a user interface. The virtual environment picture may be a picture obtained by a virtual camera from the virtual environment. In a possible implementation, the virtual camera obtains a virtual environment picture from a third visual angle of the first virtual object. Optionally, the virtual camera is set obliquely above the first virtual object. The client uses the virtual camera to observe the virtual environment with the first virtual object as the center, obtains a virtual environment picture with the first virtual object as the center, and generates the display picture corresponding to the virtual environment. In another possible implementation, the virtual camera obtains a virtual environment picture from a first visual angle of the first virtual object. Optionally, the virtual camera is set in front of the first virtual object. The client uses the virtual camera to observe the virtual environment with the first visual angle of the first virtual object, obtains a virtual environment picture with the first visual angle of the first virtual object, and generates the display picture corresponding to the virtual environment. In an exemplary embodiment, the placement position of the virtual camera is adjustable in real time. Optionally, the user may adjust the position of the virtual camera through a control operation for the user interface, and then obtain the display picture corresponding to the virtual environment at different positions. For example, the user drags the display picture corresponding to the virtual environment to adjust the position of the virtual camera. In another example, the user clicks/taps a position in a map presentation control, and uses the position as the adjusted position of the virtual camera to adjust the position of the virtual camera.

[0046] In this embodiment of this application, the client may display the display picture corresponding to the foregoing virtual environment in the user interface. Optionally, after receiving a running instruction of the foregoing application or an enabling instruction of a game battle, the client may obtain a virtual environment picture through the foregoing virtual camera, and display the display picture corresponding to the virtual environment based on the virtual environment picture. In this embodiment of this application, the foregoing user interface further includes a particular attack control. The particular attack control is a control used for controlling a virtual object to perform an attack behavior in the user interface, for example, the control may be a button, a joystick, or in another form. In some embodiments, the particular attack control is an ordinary attack control, the ordinary attack control is used for controlling a virtual object to perform an ordinary attack behavior, and the ordinary attack behavior is an attack behavior that each of a plurality of virtual objects has, for example, a plurality of virtual objects belonging to the same type has the same ordinary attack behavior. For example, for a shooter-type virtual object, an ordinary attack behavior is a long-range physical attack behavior; for a warrior-type virtual object, an ordinary attack behavior is a short-range physical attack behavior; and for a wizard-type virtual

object, an ordinary attack behavior is a long-range spell attack behavior. Optionally, the ordinary attack behavior is a behavior that does not need to consume any virtual resource and that has no cool-down duration; and there is an exclusive attack behavior corresponding to the ordinary attack behavior, the exclusive attack behavior is peculiar to a virtual object, that is, different virtual objects have different exclusive attack behaviors, each exclusive attack behavior has a corresponding exclusive attack control, and the user may control a virtual object through the exclusive attack control to perform a corresponding exclusive attack behavior. Usually, the foregoing ordinary attack behavior is referred to as an ordinary attack, and the foregoing exclusive attack behavior is referred to as a skill. Correspondingly, the ordinary attack control is referred to as an ordinary attack button, the exclusive attack control is referred to as a skill button, and a person skilled in the art may understand meanings thereof.

[0047] In a possible implementation, while displaying the display picture corresponding to the foregoing virtual environment, the client may display the foregoing particular attack control. That is, after receiving the running instruction of the foregoing application or the enabling instruction of the game battle, the client displays both the display picture corresponding to the foregoing virtual environment and the foregoing particular attack control.

[0048] In another possible implementation, to ensure interface brevity, the particular attack control is hidden in the foregoing user interface, and after receiving a display instruction of the foregoing particular attack control, the client displays the particular attack control. Optionally, the display instruction may be generated by trigger of the user in a display region of the particular attack control, and the display region may be any region in the user interface; or the display instruction is generated by trigger of the user in a touch region of the particular attack control, that is, when detecting a touch operation for the particular attack control, the client generates a display instruction of the particular attack control, and then displays the particular attack control in the user interface. The foregoing touch region may be a surrounding region of the particular attack control.

[0049] In this embodiment of this application, the foregoing particular attack control is used for controlling the first virtual object in the virtual environment to perform at least two different behaviors. The behaviors are equivalent to operations, and the operations may be a skill cast operation, a virtual item use operation, a state switching operation, and the like. This is not limited in the embodiments of this application. Optionally, the user may control the foregoing first virtual object through the particular attack control to perform any one of the foregoing two different behaviors.

[0050] In a possible implementation, the foregoing particular attack control is used for controlling the first virtual object in the virtual environment to cast at least two different skills, for example, an attack skill, a defense skill, and a healing skill. Optionally, after detecting a touch operation for the foregoing particular attack control, the client controls the first virtual object to cast a skill, where the skill is any one of the foregoing at least two different skills. In an exemplary embodiment, the foregoing user interface may include a plurality of operation controls for casting skills, and different operation controls are used for casting different skills, and different skills may alternatively be controlled and cast by the same operation control.

[0051] In another possible implementation, the foregoing particular attack control is used for controlling the first virtual object in the virtual environment to use at least two different virtual items, for example, a virtual weapon, a virtual prop, and virtual equipment. Optionally, after detecting a touch operation for the foregoing particular attack control, the client controls the first virtual object to use a virtual item, where the virtual item is any one of the foregoing at least two different virtual items. In an exemplary embodiment, the foregoing user interface may include a plurality of operation controls used for using virtual items, different operation controls are used for using different virtual items, and different virtual items may alternatively be controlled and used by the same operation control.

[0052] In still another possible implementation, the foregoing particular attack control is used for controlling the first virtual object in the virtual environment to switch to at least two different states, for example, an attack state, a defense state, and a cure state. Different states correspond to different talent skills, that is, a virtual object may use the different talent skills in the different states. Optionally, after detecting a touch operation for the foregoing particular attack control, the client controls the first virtual object to perform state switching, where the state is any one of the foregoing at least two different states. In an exemplary embodiment, the foregoing user interface may include a plurality of operation controls for switching states, and different operation controls are used for switching to different states, and different states may alternatively be controlled and switched by the same operation control.

[0053] Additionally, in an exemplary embodiment, a designer may configure behaviors corresponding to the foregoing particular attack control according to an actual situation. This is not limited in the embodiments of this application. In a possible implementation, the foregoing at least two different behaviors are the same type of behaviors, for example, the at least two different behaviors include two different skill cast behaviors. In another possible implementation, the

foregoing at least two different behaviors are different types of behaviors, for example, the at least two different behaviors include two different skill cast behaviors and one virtual item use behavior.

[0054] Step 202. Determine, in response to a touch operation for the particular attack control, a target behavior from the at least two different behaviors according to resources associated with the first virtual object.

[0055] The touch operation is an operation generated by trigger of the user. In a terminal equipped with a touch screen, the touch operation may be an operation of the user for the foregoing user interface. At a PC end, the touch operation may be an operation for the user for a key on a keyboard. In this embodiment of this application, the touch operation may be an operation for the foregoing particular attack control. Optionally, the user may click/tap an icon corresponding to the particular attack control in the user interface to generate the operation for the particular attack control touch; or the user may press a key corresponding to the particular attack control (for example, a key R) to generate the operation for the particular attack control touch.

[0056] Optionally, after displaying the foregoing user interface, the client may detect the user interface, and after detecting a touch operation in the user interface, obtain a trigger position of the touch operation. When the trigger position is in a trigger region of the foregoing particular attack control, it is determined that the touch operation is a touch operation for the foregoing particular attack control.

[0057] In a possible implementation, a position of each control in the user interface is fixed, and a fixed trigger region of each control is set in the user interface. The fixed trigger region may be any position in the user interface, for example, the fixed trigger region of the particular attack control is a surrounding region of the particular attack control. Optionally, the client may determine, according to the trigger position, whether the touch operation is located in the fixed trigger region of the particular attack control. If the touch operation is located in the fixed trigger region of the particular attack control, it is determined that the touch operation is the touch operation of the particular attack control.

[0058] In another possible implementation, the position of each control in the user interface is adjustable, that is, the user may adjust the position of each control in the user interface according to an actual situation. In this case, an activity trigger region of each control is set in the user interface. The activity trigger region may be any region based on the position of the control. For example, the activity trigger region of the particular attack control is a region that is formed by a set of points spaced from a center position of the particular attack control by a distance less than a target value. Optionally, the client may determine a distance between the trigger position and the

center position of the particular attack control according to the trigger position. If the distance is less than the target value, it is determined that the touch operation is the touch operation of the particular attack control.

[0059] In this embodiment of this application, the client determines, when a touch operation for the particular attack control is detected, a target behavior from the at least two different behaviors according to resources associated with the first virtual object. The resources are consumed items required for performing a behavior corresponding to the particular attack control, and the target behavior is a behavior that the first virtual object needs to perform. Optionally, the at least two different behaviors include a first behavior, a second behavior, and a third behavior, and the client may determine the target behavior based on the quantity of resources. Optionally, the foregoing step 202 includes the following several steps:

[0060] 1. Obtain a quantity of the resources associated with the first virtual object.

[0061] 2. Determine the target behavior from the at least two different behaviors according to a value range to which the quantity belongs.

[0062] Optionally, a correspondence between value ranges of quantities and behaviors may be stored in advance in a configuration table. After determining a value range to which the quantity of resources associated with the first virtual object belongs, the client may query the configuration table to determine a corresponding target behavior.

[0063] In this embodiment of this application, after detecting a touch operation for the particular attack control, the client obtains a quantity of resources associated with the first virtual object, and determines a target behavior from the at least two different behaviors according to a value range to which the quantity belongs. Optionally, when the quantity belongs to a first value range, the target behavior is determined as the first behavior; when the quantity belongs to a second value range, the target behavior is determined as the second behavior; and when the quantity belongs to a third value range, the target behavior is determined as the third behavior. The first behavior is configured for fighting back against an attack behavior of a second virtual object; the second behavior is configured for reducing a movement speed of the second virtual object and increasing health points of the first virtual object when health points of the second virtual object are reduced; and the third behavior is configured for clearing the health points of the second virtual object when the health points of the second virtual object is less than a target value. The second virtual object is a virtual object in a camp different from that of the first virtual object; and the first value range, the second value range, and the third value range are pairwise independent and have no intersection.

[0064] For example, with reference to FIG. 3, after the client detects a touch operation for a particular attack control 30, the target behavior is a first behavior 32 when the quantity of resources falls within a first value range 31; the target behavior is a second behavior 34 when the quantity of resources falls within a second value range 33; and the target behavior is a third behavior 36 when the quantity of resources falls within a third value range 35.

[0065] The foregoing descriptions of the first behavior, the second behavior, and the third behavior are only exemplary and explanatory. In an exemplary embodiment, a designer may design a specific function of each operation according to an actual situation. This is not limited in the embodiments of this application.

[0066] Certainly, in an exemplary embodiment, the designer may also flexibly set and adjust a quantity of behaviors corresponding to the particular attack control. This is not limited in the embodiments of this application.

[0067] Further, the foregoing descriptions of the target behavior selecting manners are only exemplary. In an exemplary embodiment, after displaying the display picture corresponding to the virtual environment, the client may detect resources associated with the first virtual object, determine different target behaviors according to different resources, and then directly control, after detecting the touch operation for the particular attack control, the first virtual object to perform the target behavior.

[0068] Optionally, the resources may be collected and obtained through a resource increase operation. In a possible implementation, after displaying the display picture corresponding to the foregoing virtual environment, the client may detect the resource increase operation. The client determines a resource increase quantity of the first virtual object when a resource increase operation is detected; and updates the quantity of the resources associated with the first virtual object according to the resource increase quantity.

[0069] Optionally, the resource increase operation includes at least one of the following: the first virtual object killing a second virtual object, the first virtual object causing health points of the second virtual object to lose, the first virtual object performing a fifth behavior to cause health points of the second virtual object to lose, the first virtual object performing the fifth behavior, a touch operation for an operation control corresponding to the fifth behavior, and the first virtual object obtaining a resource increase prop, where the second virtual object is a virtual object in a camp different from that of the first virtual object.

[0070] That the first virtual object kills the second virtual object means that after the second virtual object receives an attack from the first virtual object, the health points of the second virtual object are zero. An attack value of the first virtual object for the second virtual object may be less than a maximum value of the health points of the second virtual object. Certainly, the attack value of the first virtual object for the second virtual object may alternatively be greater than or equal to the maximum value of the health points of the second virtual object. This is not limited in the embodiments of this application. Optionally, after detecting that the first virtual object kills the second virtual object, the client may determine the resource increase quantity according to the attack value of the first virtual object for the second virtual object, and update the quantity of resources associated with the first virtual object according to the resource increase quantity. For example, the attack value and the resource increase quantity are in a positive correlation.

[0071] That the first virtual object causes health points of the second virtual object to lose means that the first virtual object has successfully attacked the second virtual object. Optionally, after detecting that the first virtual object causes health points of the second virtual object to lose, the client may determine the resource increase quantity according to a health points loss value of the second virtual object, and update the quantity of resources associated with the first virtual object according to the resource increase quantity. For example, the health points loss value and the resource increase quantity are in a positive correlation.

[0072] That the first virtual object performs a fifth behavior to cause health points of the second virtual object to lose means that the first virtual object has successfully attacked the second virtual object by using the fifth behavior. The fifth behavior may be any behavior that the first virtual object can perform, for example, at least two different behaviors corresponding to the particular attack control. Optionally, after detecting that the first virtual object performs a fifth behavior to cause health points of the second virtual object to lose, the client may determine the resource increase quantity according to a health points loss value of the second virtual object, and update the quantity of resources associated with the first virtual object according to the resource increase quantity. For example, the health points loss value and the resource increase quantity are in a positive correlation.

[0073] That the first virtual object performs a fifth behavior means that the first virtual object performs a corresponding behavior. The fifth behavior may be any behavior that the first virtual object can perform, for example, at least two different behaviors corresponding to the particular attack control; or the fifth behavior may alternatively be a particular behavior of the first virtual object, for example, a skill cast behavior. This is not limited in the embodiments of this

application. Optionally, after detecting that the first virtual object performs a fifth behavior, the client determines the resource increase quantity according to the fifth behavior, and updates the quantity of resources associated with the first virtual object according to the resource increase quantity. Resource increase quantities corresponding to different behaviors may be the same or different.

[0074] The touch operation for the operation control corresponding to the fifth behavior is a touch operation of a behavior. The fifth behavior may be any behavior that the first virtual object can perform, for example, at least two different behaviors corresponding to the particular attack control; or the fifth behavior may alternatively be a particular behavior of the first virtual object, for example, a skill cast behavior. This is not limited in the embodiments of this application. Optionally, after detecting a touch operation for an operation control corresponding to the fifth behavior, the client determines the resource increase quantity according to the fifth behavior, and updates the quantity of resources associated with the first virtual object according to the resource increase quantity. Resource increase quantities corresponding to different behaviors may be the same or different.

[0075] That the first virtual object obtains a resource increase prop means that the first virtual object collects the resource increase prop in the virtual environment. The resource increase prop may be placed at a particular position in the virtual environment when the display picture corresponding to the virtual environment is displayed, and may alternatively be a reward prop dropped when the first virtual object completes a particular behavior. This is not limited in the embodiments of this application. Optionally, after detecting that the first virtual object obtains a resource increase prop, the client determines the resource increase quantity according to the resource increase prop, and updates the quantity of resources associated with the first virtual object according to the resource increase quantity. Resource increase quantities corresponding to different resource increase props may be the same or different.

[0076] The resources are only items for the at least two behaviors. That is to say, the resources are consumed only when the first virtual object performs any behavior of the at least two behaviors other than the ordinary attack behavior; and when the first virtual object performs a behavior other than the at least two behaviors, the resources do not need to be consumed, and a corresponding behavior is not determined according to the resources either.

[0077] Step 203. Control the first virtual object to perform the target behavior.

[0078] The target behavior is a behavior that the first virtual object needs to perform. Optionally, the target behavior may be any one of the at least two different behaviors. The target

behavior may be a skill cast behavior, a virtual item use behavior, a state switching behavior, or the like. This is not limited in the embodiments of this application.

[0079] In this embodiment of this application, after determining the target behavior from the at least two different behaviors, the client controls the first virtual object to perform the target behavior. Optionally, when controlling the first virtual object to perform the target behavior, the client may determine the action target of the target behavior based on the action range of the target behavior, and then control the first virtual object to perform the target behavior for the action target. The action range is a behavior validation range of the target behavior, for example, an attack range of an attack skill, a throw range of a virtual item, or a jumping impact range of a virtual object. Optionally, different behaviors correspond to different action ranges.

[0080] Optionally, after obtaining the action range of the target behavior, the client may obtain attribute information of a target virtual object in the action range, and select an action target corresponding to the target behavior according to the attribute information. The target virtual object is an object to which the target behavior is directed. Optionally, when the target behavior is an attack behavior, the target virtual object is the second virtual object in a camp different from that of the first virtual object; and when the target behavior is a gaining behavior, the target virtual object is a virtual object in a camp the same as that of the first virtual object.

[0081] In a possible implementation, the attribute information includes health points of the target virtual object. Optionally, the client may sort target virtual objects in ascending order of health points, and then select a top-ranked target virtual object as an action target corresponding to the target behavior according to a quantity of action targets of the target behavior.

[0082] In another possible implementation, the attribute information includes a distance between the target virtual object and the first virtual object. Optionally, the client sorts target virtual objects in ascending order of distances, and then selects a top-ranked target virtual object as an action target corresponding to the target behavior according to a quantity of action targets of the target behavior.

[0083] Optionally, after determining the action target of the target behavior, the client controls the first virtual object to perform the target behavior for the action target.

[0084] The foregoing selection of the action target is only exemplary and explanatory. In an exemplary embodiment, the client may alternatively determine the action target corresponding to the target behavior according to other content in the attribute information, for example, a movement

speed of the target virtual object, a defense value of the target virtual object, or an attack speed of the target virtual object.

[0085] To sum up, in the technical solution provided in this embodiment of this application, a first virtual object can be controlled through a particular attack control to perform at least two different behaviors, so that one control controls different behaviors, to avoid interface congestion caused by excessive icons corresponding to operation controls, reduce the misoperation probability of the user, and ensure the operation accuracy; and when behaviors of the first virtual object are increased, no new icon corresponding to the operation controls needs to be added in a user interface, thereby effectively ensuring brevity of the user interface.

[0086] Additionally, when a touch operation for the particular attack control is detected, different behaviors to be performed by the first virtual object are selected from the at least two different behaviors based on resources associated with the first virtual object. That is, when the first virtual object owns different resources, actions of the particular attack control are different, so that the particular attack control is diversified, and the user may control the resources associated with the first virtual object to control the actions of the particular attack control, and then use the particular attack control to implement the different actions, making operations flexible.

[0087] This application is completely described below with reference to FIG. 4.

[0088] Step 401. A client detects a skill cast instruction for a first virtual object.

[0089] Optionally, the user interface includes a plurality of skill operation controls, the skill operation controls are used for controlling the first virtual object to cast skills, and different skill operation controls correspond to different skills.

[0090] In this embodiment of this application, after the user interface is displayed, the client detects the user interface. After a touch operation in the user interface is detected, a skill operation control to which the touch operation is directed is determined according to a position of the touch operation, and then a skill cast instruction for the first virtual object is generated according to the skill operation control.

[0091] Step 402. The client controls, according to the skill cast instruction, the first virtual object to cast a skill, and increases resources associated with the first virtual object.

[0092] In this embodiment of this application, after detecting the skill cast instruction, the client controls, according to the skill cast instruction, the first virtual object to cast a corresponding skill, and increases resources associated with the first virtual object. The resources are consumed items required for performing a behavior corresponding to the particular attack control.

[0093] Optionally, when controlling the first virtual object to cast a skill, the client obtains a resource increase quantity corresponding to the skill; and further, with the resource increase quantity as a reference, increases resources associated with the first virtual object, and updates the quantity of resources currently associated with the first virtual object.

[0094] Optionally, resource increase quantities corresponding to different skills may be the same or different. This is not limited in the embodiments of this application.

[0095] In a possible implementation, when different skills correspond to the same resource increase quantity, the client may predict the resource increase quantity, and then may obtain, when controlling the first virtual object to cast a skill, the resource increase quantity without considering the skill, thereby improving resource quantity update efficiency.

[0096] In another possible implementation, when different skills correspond to different resource increase quantities, the client needs to first determine a skill when controlling the first virtual object to cast the skill, and then determine a resource increase quantity corresponding to the skill.

[0097] Step 403. The client detects a touch operation for a particular attack control.

[0098] In this embodiment of this application, the user interface includes the particular attack control, and the particular attack control is an attack control that each virtual object has. After the user interface is displayed, the client detects the user interface. After a touch operation in the user interface is detected, whether the touch operation is an operation for the particular attack control is determined according to position of the touch operation. Then, when the touch operation is an operation for the particular attack control, the client determines that the touch operation for the particular attack control is detected.

[0099] Step 404. The client determines a quantity of the resources associated with the first virtual object.

[0100] In this embodiment of this application, after detecting a touch operation for the particular attack control, the client obtains a quantity of resources associated with the first virtual object, and then determines, according to a value range to which the quantity belongs, a behavior that the first virtual object needs to perform.

[0101] Step 405. When the quantity belongs to a first value range, the client controls the first virtual object to perform a first behavior.

[0102] Step 406. When the quantity belongs to a second value range, the client controls the first virtual object to perform a second behavior.

[0103] Step 407. When the quantity belongs to a third value range, the client controls the first virtual object to perform a third behavior.

[0104] Step 408. When the quantity is less than a lower limit value of the first value range, the client controls the first virtual object to perform an ordinary attack behavior.

[0105] The foregoing is a simple description of the target behavior selecting manner. In this embodiment of this application, the client may determine, based on the resources associated with the first virtual object, the target behavior with reference to an operation attribute of the touch operation for the particular attack control.

[0106] In an exemplary embodiment, the foregoing step 202 includes the following several steps:

[0107] 1. Obtain a quantity of the resources associated with the first virtual object.

[0108] 2. Determine an operation attribute of the touch operation.

[0109] 3. Determine the target behavior from the at least two different behaviors based on the quantity and the operation attribute.

[0110] The operation attribute is used for representing an attribute of the touch operation. Different touch operations have different operation attributes. Optionally, the operation attribute includes but is not limited to at least one of the following: a touch duration, a touch pressure, a quantity of touch times, and a touch direction.

[0111] The touch duration is a duration of the touch operation. Optionally, the client may begin timing from a moment at which the touch operation is detected, until the touch operation disappears, to obtain the touch duration of the touch operation. The touch pressure is a value of a pressure generated by the touch operation of the user and detected through a pressure sensor. The quantity of touch times is a quantity of times of continuously touching a control by the user in a time threshold. For example, a quantity of touch times corresponding to a click operation is 1, a quantity of touch times corresponding to a double-click operation is 2, and a quantity of touch times corresponding to a triple-click operation is 3. The touch direction is a direction of a slide track when the touch operation is a slide operation.

[0112] In this embodiment of this application, after detecting a touch operation for the ordinary attack control, the client obtains a quantity of resources associated with the first virtual object and the operation attribute of the touch operation, and then determines a target behavior from the at least two different behaviors based on the quantity and the operation attribute.

[0113] In a possible implementation, the client determines, according to a range to which the quantity of resources associated with the first virtual object belongs, the target behavior with reference to the operation attribute of the touch operation. Optionally, the target behavior is determined as the first behavior when the quantity belongs to a first value range and the operation attribute is a first operation attribute. The target behavior is determined as the second behavior when the quantity belongs to a second value range and the operation attribute is the first operation attribute; and the target behavior is determined as the first behavior when the quantity belongs to the second value range and the operation attribute is a second operation attribute. The target behavior is determined as the third behavior when the quantity belongs to a third value range and the operation attribute is the first operation attribute; the target behavior is determined as the first behavior when the quantity belongs to the third value range and the operation attribute is the second operation attribute; and the target behavior is determined as the second behavior when the quantity belongs to the third value range and the operation attribute is a third operation attribute.

[0114] Settings about the first value range, the second value range, and the third value range are not limited in the embodiments of this application. Optionally, the first value range, the second value range, and the third value range may be mutually consecutive value ranges. For example, the first value range is from 10 to 20, the second value range is from 20 to 30, and the third value range is from 30 to 40. Alternatively, the first value range, the second value range, and the third value range may be mutually non-consecutive value ranges. For example, the first value range is from 10 to 20, the second value range is from 30 to 40, and the third value range is from 50 to 60.

[0115] Further, settings about the first operation attribute, the second operation attribute, and the third operation attribute are not limited in the embodiments of this application. Optionally, different touch operations correspond to different operation attributes. For example, when the touch operation is a click/tap operation, the operation attribute may be a quantity of touch times, for example, the first operation attribute is 1 time of touch, the second operation attribute is 2 times of touch, and the third operation attribute is 3 times of touch; or the operation attribute may be a touch duration, for example, the first operation attribute is 1s of touch, the second operation is 2s of touch, and the third operation attribute is 3s of touch. In another example, when the touch operation is a slide operation, the operation attribute is a slide direction, for example, the first operation attribute is upward slide, the second operation attribute is downward slide, and the third operation attribute is left slide.

[0116] For example, with reference to FIG. 5, using an example in which a touch operation is a click/tap operation and an operation attribute is a quantity of touch times, after the client detects a

touch operation for a particular attack control 50, when the quantity belongs to a first value range 51, if the touch operation is a click operation 52, a target behavior is a first behavior 53. When the quantity belongs to a second value range 54, if the touch operation is the click operation 52, the target behavior is a second behavior 55; and if the touch operation is a double-click operation 56, the target behavior is the first behavior 53. When the quantity belongs to a third value range 57, if the touch operation is the click operation 52, the target behavior is a third behavior 58; if the touch operation is the double-click operation 56, the target behavior is the first behavior 53; and if the touch operation is a triple-click operation 59, the target behavior is the second behavior 55.

[0117] In another possible implementation, the client determines, according to the operation attribute of the touch operation, the target behavior with reference to a relationship between a quantity of resources associated with the first virtual object and a threshold. Optionally, the target behavior is determined as the first behavior when the operation attribute is a first operation attribute and the quantity is greater than a first threshold. The target behavior is determined as the second behavior when the operation attribute is a second operation attribute and the quantity is greater than a second threshold; and the target behavior is determined as the first behavior when the operation attribute is the second operation attribute and the quantity is less than the second threshold and greater than the first threshold. The target behavior is determined as the third behavior when the operation attribute is a third operation attribute and the quantity is greater than a third threshold; the target behavior is determined as the second behavior when the operation attribute is the third operation attribute and the quantity is less than the third threshold and greater than the second threshold; and the target behavior is determined as the first behavior when the operation attribute is the third operation attribute and the quantity is less than the second threshold and greater than the first threshold.

[0118] Settings about the first threshold, the second threshold, and the third threshold are not limited in the embodiments of this application. Optionally, the first threshold, the second threshold, and the third threshold may be any value. For example, the first threshold is 10, the second threshold is 20, and the third threshold is 30.

[0119] For example, with reference to FIG. 6, using an example in which a touch operation is a click/tap operation and an operation attribute is a quantity of touch times, after the client detects a touch operation for a particular attack control 60, when the touch operation is a click operation 61, if the quantity is greater than a first threshold 62, a target behavior is a first behavior 63. When the touch operation is a double-click operation 64, if the quantity is greater than a second threshold 65, the target behavior is a second behavior 66; and if the quantity is less than the second threshold 65

and greater than the first threshold 62, the target behavior is the first behavior 63. When the touch operation is a triple-click operation 67, if the quantity is greater than a third threshold 68, the target behavior is a third behavior 69; if the quantity is less than the third threshold 68 and greater than the second threshold 65, the target behavior is the second behavior 66; and if the quantity is less than the second threshold 65 and greater than the first threshold 62, the target behavior is the first behavior 63.

[0120] In this embodiment of this application, the particular attack control may be further used for controlling the first virtual object to perform an ordinary attack behavior. The ordinary attack behavior is a physical attack behavior of the first virtual object. Optionally, the ordinary attack behavior may be a behavior that does not need to consume any virtual resource and that has no cool-down duration. The cool-down duration is a minimum value of a cast time interval between the same behaviors, and the virtual resource may be the foregoing resource, and may alternatively be energy that another behavior needs to consume. This is not limited in the embodiments of this application. In an exemplary embodiment, the ordinary attack behavior may be a behavior that each of a plurality of virtual objects has. Optionally, the same type of virtual objects corresponds to similar ordinary attack behaviors, for example, similar attack distances. However, other behaviors (for example, referred to as skills) of a virtual object different from the ordinary attack behavior are unique, and virtual objects have different other behaviors (that is, exclusive attack behaviors described above). That is, any behavior of the other behaviors can only be exclusive to a particular virtual object, and other virtual objects cannot have the any behavior. That is to say, in this embodiment of this application, each of the first behavior, the second behavior, and the third behavior of the first virtual object is a behavior exclusive to the first virtual object, and other virtual objects cannot have the first behavior, the second behavior, or the third behavior; and the ordinary attack behavior of the first virtual object is not a behavior exclusive to the first virtual object, and other virtual objects in a type the same as that of the first virtual object may also have the same ordinary attack behavior.

[0121] Optionally, the at least two different behaviors further include a fourth behavior, and the fourth behavior is an ordinary attack behavior.

[0122] In a possible implementation, after the client detects the touch operation for the particular attack control, when the quantity of resources associated with the first virtual object is less than the first threshold, the target behavior is determined as the fourth behavior. The first threshold may be less than or equal to the lower limit value of the first value range. As shown in FIG. 7, after the client detects a touch operation for a particular attack control 70, when the quantity

of resources associated with the first virtual object is less than a lower limit value 71 of the first value range 51, the target behavior is an ordinary attack behavior 72.

[0123] In another possible implementation, after the client detects the touch operation for the particular attack control, when the quantity of resources associated with the first virtual object is greater than the first threshold and the touch operation is a fourth operation attribute, the target behavior is determined as the ordinary attack behavior. The fourth operation attribute is any operation attribute different from each of the first operation attribute, the second operation attribute, and the third operation attribute.

[0124] Optionally, in this embodiment of this application, to enable the user to accurately grasp the operation attribute of the touch operation, to ensure that the target behavior performed by the first virtual object is a behavior required by the user, a reminding item may be included around the particular attack control. For different operation attributes, the reminding items are in different display styles. For example, with reference to FIG. 8, a first reminding item 81, a second reminding item 82, and a third reminding item 83 are disposed around a particular attack control 80. When the touch duration corresponding to the touch operation is 1s, the first reminding item 81 is highlighted; when the touch duration corresponding to the touch operation is 2s, the first reminding item 81 and the second reminding item 82 are highlighted; and when the touch duration corresponding to the touch operation is 3s, the first reminding item 81, the second reminding item 82, and the third reminding item 83 are highlighted.

[0125] The target behavior is determined above in a manner of combining the quantity and the operation attribute. Certainly, in another possible implementation, to ensure operation flexibility and convenience, the client may alternatively determine the target behavior according to only the quantity of resources associated with the first virtual object. Optionally, the user interface may include a function switch control, the function switch control being used for enabling or disabling a function of collecting the resources. The user may control enabling or disabling of the function of collecting the resources, to control the quantity of resources associated with the first virtual object, and then control the target behavior corresponding to the particular attack control. Optionally, specific steps are as follows:

[0126] 1. Display a function switch control.

[0127] 2. Disable, when the function of collecting the resources is in an enabled state, the function of collecting the resources for the first virtual object in response to a touch operation for the function switch control.

[0128] 3. Enable, when the function of collecting the resources is in a disabled state, the function of collecting the resources for the first virtual object in response to a touch operation for the function switch control.

[0129] The function switch control is a control used for enabling or disabling the function of collecting the resources. Optionally, the function switch control may be set on any position in the user interface. In a possible implementation, while displaying the display picture corresponding to the foregoing virtual environment, the client may display the function switch control. In another possible implementation, to ensure interface brevity, the function switch control is hidden in the foregoing user interface, and after receiving a display instruction of the foregoing function switch control, the client displays the function switch control. Optionally, the display instruction may be generated by trigger of the user in a display region of the function switch control, and the display region may be any region in the user interface. In another possible implementation, the display instruction is generated by trigger of the user in a touch region of the function switch control, that is, when detecting a touch operation for the function switch control, the client generates a display instruction of the function switch control, and then displays the function switch control in the user interface. The foregoing touch region may be a surrounding region of the function switch control.

[0130] In this embodiment of this application, when the function of collecting the resources is in an enabled state, the function of collecting the resources for the first virtual object is disabled if the client detects a touch operation for the function switch control. In this case, the first virtual object cannot collect the resources, and the target behavior corresponding to the particular attack control is in a fixed state. When the function of collecting the resources is in a disabled state, the function of collecting the resources for the first virtual object is enabled if the client detects a touch operation for the function switch control. In this case, the first virtual object can collect the resources, and the target behavior corresponding to the particular attack control is in a changing state.

[0131] FIG. 9 is a flowchart of a virtual object control method according to another embodiment of this application. The method may be applied to a terminal. For example, an execution entity of each step may be the terminal 10 (referred to as the "client" below) in the running environment of the application shown in FIG. 1. The method may include the following steps (901 to 905):

[0132] Step 901: Display a display picture corresponding to a virtual environment and display a particular attack control.

[0133] Step 902. Determine, in response to a touch operation for the particular attack control, a target behavior from the at least two different behaviors according to resources associated with the first virtual object.

[0134] Step 903. Control the first virtual object to perform the target behavior.

[0135] Step 901 to step 903 are the same as step 201 to step 203 in the embodiment of FIG. 2. For details, reference is made to the embodiment of FIG. 2. Details are not described herein again.

[0136] Step 904. Determine a resource decrease quantity of the first virtual object.

[0137] The resource decrease quantity is used for indicating resources consumed by the first virtual object to perform the target behavior. Optionally, resource decrease quantities corresponding to different target behaviors may be the same or different. In this embodiment of this application, after controlling the first virtual object to perform the target behavior, the client obtains the resource decrease quantity of the first virtual object.

[0138] In a possible implementation, the client obtains the resource decrease quantity according to the target behavior, that is, the resource decrease quantity corresponding to the same target behavior is unchanged.

[0139] In another possible implementation, the client obtains the resource decrease quantity according to an action effect of the target behavior, that is, the resource decrease quantity corresponding to the same target behavior is a changing value. The action effect is used for indicating an effect on the virtual environment or virtual object after the target behavior is performed, and the action effect includes at least one of the following: a quantity of action targets of the target behavior, combat power of an action target of the target behavior, virtual resources associated with an action target of the target behavior, a health points loss value caused by the target behavior to an action target, and the like.

[0140] Step 905. Update a quantity of the resources associated with the first virtual object according to the resource decrease quantity.

[0141] In this embodiment of this application, after obtaining the resource decrease quantity, the client updates the quantity of the resources associated with the first virtual object according to the resource decrease quantity.

[0142] To sum up, in the technical solution provided in this embodiment of this application, a first virtual object can be controlled through a particular attack control to perform at least two different behaviors, so that one control controls different behaviors, improving interface display

brevity. When the first virtual object owns different resources, actions of the particular attack control are different, so that the particular attack control is diversified, and the user may control the resources associated with the first virtual object to control the actions of the particular attack control, making operations flexible. Moreover, after a target behavior is performed, the resources associated with the first virtual object are consumed, to ensure game battle fairness.

[0143] The following is an apparatus embodiment of this application, which can be used to execute the method embodiments of this application. For details not disclosed in the apparatus embodiments of this application, reference is made to the method embodiments of this application.

[0144] FIG. 10 is a block diagram of a virtual object control apparatus according to an embodiment of this application. The apparatus has a function of implementing the foregoing virtual object control methods, and the function may be implemented by hardware or by hardware executing corresponding software. The apparatus may be a terminal or may be disposed in a terminal. The apparatus 1000 may include: a control display module 1010, a behavior determining module 1020, and a behavior execution module 1030.

[0145] The control display module 1010 is configured to display a display picture corresponding to a virtual environment and to display a particular attack control, the particular attack control being used for controlling a first virtual object in the virtual environment to perform at least two different behaviors.

[0146] The behavior determining module 1020 is configured to determine, in response to a touch operation for the particular attack control, a target behavior from the at least two different behaviors according to resources associated with the first virtual object, the resources being consumed items required for performing a behavior corresponding to the particular attack control.

[0147] The behavior execution module 1030 is configured to control the first virtual object to perform the target behavior.

[0148] In an exemplary embodiment, as shown in FIG. 11, the behavior determining module 1020 further includes: a quantity obtaining unit 1021 and a behavior determining unit 1022.

[0149] The quantity obtaining unit 1021 is configured to obtain a quantity of the resources associated with the first virtual object.

[0150] The behavior determining unit 1022 is configured to determine the target behavior from the at least two different behaviors according to a value range to which the quantity belongs.

[0151] In an exemplary embodiment, the at least two different behaviors include a first behavior, a second behavior, and a third behavior; and the behavior determining unit 1022 is

configured to determine the target behavior as the first behavior when the quantity belongs to a first value range; determine the target behavior as the second behavior when the quantity belongs to a second value range; and determine the target behavior as the third behavior when the quantity belongs to a third value range, where the first value range, the second value range, and the third value range are pairwise independent and have no intersection.

[0152] In an exemplary embodiment, as shown in FIG. 11, the behavior determining module 1020 further includes: an attribute determining unit 1023.

[0153] The quantity obtaining unit 1021 is configured to obtain a quantity of the resources associated with the first virtual object.

[0154] The attribute determining unit 1023 is configured to determine an operation attribute of the touch operation.

[0155] The behavior determining unit 1022 is further configured to determine the target behavior from the at least two different behaviors based on the quantity and the operation attribute.

[0156] In an exemplary embodiment, the at least two different behaviors include a first behavior, a second behavior, and a third behavior; and the behavior determining unit 1023 is configured to determine the target behavior as the first behavior when the quantity belongs to a first value range and the operation attribute is a first operation attribute; determining the target behavior as the second behavior when the quantity belongs to a second value range and the operation attribute is the first operation attribute; and determine the target behavior as the first behavior when the quantity belongs to the second value range and the operation attribute is the second operation attribute; determining the target behavior as the third behavior when the quantity belongs to a third value range and the operation attribute is the first operation attribute; determining the target behavior as the first behavior when the quantity belongs to the third value range and the operation attribute is the second operation attribute; and determining the target behavior as the second behavior when the quantity belongs to the third value range and the operation attribute is a third operation attribute.

[0157] In an exemplary embodiment, the at least two different behaviors include a first behavior, a second behavior, and a third behavior; and the behavior determining unit 1023 is configured to determine the target behavior as the first behavior when the operation attribute is a first operation attribute and the quantity is greater than a first threshold; determine the target behavior as the second behavior when the operation attribute is a second operation attribute and the quantity is greater than a second threshold; and determine the target behavior as the first behavior

when the operation attribute is the second operation attribute and the quantity is less than the second threshold and greater than the first threshold; and determine the target behavior as the third behavior when the operation attribute is a third operation attribute and the quantity is greater than a third threshold; determine the target behavior as the second behavior when the operation attribute is the third operation attribute and the quantity is less than the third threshold and greater than the second threshold; and determine the target behavior as the first behavior when the operation attribute is the third operation attribute and the quantity is less than the second threshold and greater than the first threshold.

[0158] In an exemplary embodiment, the first behavior is configured for fighting back against an attack behavior of a second virtual object; the second behavior is configured for reducing a movement speed of the second virtual object and increasing health points of the first virtual object when health points of the second virtual object are reduced; and the third behavior is configured for clearing the health points of the second virtual object when the health points of the second virtual object is less than a target value, where the second virtual object is a virtual object in a camp different from that of the first virtual object.

[0159] In an exemplary embodiment, the at least two different behaviors further include a fourth behavior, and the fourth behavior is an ordinary attack behavior; and the operation determining module 1020 is further configured to determine the target behavior as the fourth behavior when the quantity is less than the first threshold; or determine the target behavior as the fourth behavior when the quantity is greater than the first threshold and the operation attribute of the touch operation is a fourth operation attribute.

[0160] In an exemplary embodiment, as shown in FIG. 11, the apparatus 1000 further includes: a function disabling module 1040 and a function enabling module 1050.

[0161] The control display module 1010 is further configured to display a function switch control, the function switch control being used for enabling or disabling a function of collecting the resources.

[0162] The function disabling module 1040 is configured to disable, when the function of collecting the resources is in an enabled state, the function of collecting the resources for the first virtual object in response to a touch operation for the function switch control.

[0163] The function enabling module 1050 is configured to enable, when the function of collecting the resources is in a disabled state, the function of collecting the resources for the first virtual object in response to a touch operation for the function switch control.

[0164] In an exemplary embodiment, as shown in FIG. 11, the apparatus 1000 further includes: a resource update module 1060.

[0165] The resource update module 1060 is configured to determine a resource increase quantity of the first virtual object when a resource increase operation is detected; and update the quantity of the resources associated with the first virtual object according to the resource increase quantity.

[0166] In an exemplary embodiment, the resource increase operation includes at least one of the following: the first virtual object killing a second virtual object, the first virtual object causing health points of the second virtual object to lose, the first virtual object performing a fifth behavior to cause health points of the second virtual object to lose, the first virtual object performing the fifth behavior, a touch operation for an operation control corresponding to the fifth behavior, and the first virtual object obtaining a resource increase prop, where the second virtual object is a virtual object in a camp different from that of the first virtual object.

[0167] In an exemplary embodiment, the resource update module 1060 is configured to determine a resource decrease quantity of the first virtual object; and update the quantity of the resources associated with the first virtual object according to the resource decrease quantity.

[0168] To sum up, in the technical solution provided in this embodiment of this application, a first virtual object can be controlled through a particular attack control to perform at least two different behaviors, so that one control controls different behaviors. Therefore, when behaviors of the first virtual object are increased, no new icon corresponding to the operation controls needs to be added in a user interface, to avoid interface congestion caused by excessive icons, reduce the misoperation probability of the user, and ensure the operation accuracy.

[0169] Additionally, when a touch operation for the particular attack control is detected, different behaviors to be performed by the first virtual object are selected from the at least two different behaviors based on resources associated with the first virtual object. That is, when the first virtual object owns different resources, actions of the particular attack control are different, so that the particular attack control is diversified, and the user may control the resources associated with the first virtual object to control the actions of the particular attack control, and then use the particular attack control to implement the different actions, making operations flexible.

[0170] When the apparatus provided in the foregoing embodiment implements the functions of the apparatus, only division of the foregoing function modules is used as an example for description. In the practical application, the functions may be allocated to and completed by

different function modules according to requirements. That is, an internal structure of the device is divided into different function modules, to complete 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, reference may be made to the method embodiments, and details are not described herein again.

[0171] FIG. 12 is a structural block diagram of a terminal 1200 according to an embodiment of this application. The terminal 1200 may be an electronic device such as a mobile phone, a tablet computer, a game console, an ebook reader, a multimedia player, a wearable device, or a PC. The terminal is configured to implement the virtual object control method provided in the foregoing embodiments. The terminal may be the terminal 10 in the game running environment shown in FIG. 1. Specifically:

[0172] Generally, the terminal 1200 includes a processor 1201 and a memory 1202.

[0173] The processor 1201 may include one or more processing cores, for example, a 4-core processor or an 8-core processor. The processor 1201 may be implemented in 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 1201 may alternatively include a main processor and a coprocessor. The main processor is configured to process data in an awake state, also referred to as a central processing unit (CPU); and the coprocessor is a low-power processor configured to process data in a standby state. In some embodiments, the processor 1201 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 1201 may further include an artificial intelligence (AI) processor. The AI processor is configured to process a calculation operation related to machine learning.

[0174] The memory 1202 may include one or more computer-readable storage media. The computer-readable storage medium may be non-transient. The memory 1202 may further include a high-speed random access memory and a non-volatile memory such as one or more magnetic disk storage devices and a flash storage device. In some embodiments, the non-transient computer-readable storage medium in the memory 1202 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.

[0175] In some embodiments, the terminal 1200 may optionally include a peripheral interface 1203 and at least one peripheral. The processor 1201, the memory 1202, and the peripheral

interface 1203 may be connected through a bus or a signal cable. Each peripheral may be connected to the peripheral interface 1203 through a bus, a signal cable, or a circuit board. Specifically, the peripheral includes at least one of a radio frequency circuit 1204, a display screen 1205, a camera assembly 1206, an audio circuit 1207, a positioning assembly 1208, and a power supply 1209.

[0176] A person skilled in the art may understand that the structure shown in FIG. 12 does not constitute a limitation on the terminal 1200, 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.

[0177] In an exemplary embodiment, a computer-readable storage medium is further provided, 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 executed by a processor to implement the foregoing virtual object control method.

[0178] Optionally, 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 resistance random access memory (ReRAM) and a dynamic random access memory (DRAM).

[0179] In an exemplary embodiment, a computer program product or a computer program is further provided. The computer program product or the computer program includes computer instructions, and the computer instructions are stored in a computer-readable storage medium. A processor of a computer device reads the computer instructions from the computer-readable storage medium and executes the computer instructions to cause the computer device to perform the foregoing virtual object control method.

[0180] It is to be understood that "plurality of" mentioned in this specification means two or more. The "and/or" describes an association relationship for describing associated objects and represents that three relationships may exist. For example, A and/or B may represent the following three cases: Only A exists, both A and B exist, and only B exists. The character "/" generally indicates an "or" relationship between the associated objects. In addition, the step numbers described in this specification merely schematically 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.

[0181] 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

1. A virtual object control method, performed by a terminal, the method comprising:

displaying a display picture corresponding to a virtual environment and displaying a particular attack control, wherein the particular attack control is configured for controlling a first virtual object in the virtual environment to perform at least two different behaviors;

determining, in response to a touch operation for the particular attack control, a target behavior from the at least two different behaviors according to resources associated with the first virtual object, wherein the resources are consumed items for performing a behavior corresponding to the particular attack control; and

controlling the first virtual object to perform the target behavior,

wherein the determining the target behavior from the at least two different behaviors according to resources associated with the first virtual object comprises:

obtaining a quantity of the resources associated with the first virtual object; and determining the target behavior from the at least two different behaviors according to a value range to which the quantity belongs; or

obtaining a quantity of the resources associated with the first virtual object; determining an operation attribute of the touch operation; and determining the target behavior from the at least two different behaviors based on the quantity and the operation attribute.

2. The method according to claim 1, wherein the at least two different behaviors comprise a first behavior, a second behavior, and a third behavior; and

the determining the target behavior from the at least two different behaviors according to the value range to which the quantity belongs comprises:

determining the target behavior as the first behavior when the quantity belongs to a first value range;

determining the target behavior as the second behavior when the quantity belongs to a second value range; and

determining the target behavior as the third behavior when the quantity belongs to a third value range, wherein the first value range, the second value range, and the third value range are pairwise independent and have no intersection.

3. The method according to claim 1, wherein the at least two different behaviors comprise a first behavior, a second behavior, and a third behavior; and

the determining the target behavior from the at least two different behaviors based on the quantity and the operation attribute comprises:

determining the target behavior as the first behavior when the quantity belongs to a first value range and the operation attribute is a first operation attribute;

determining the target behavior as the second behavior when the quantity belongs to a second value range and the operation attribute is the first operation attribute; and determining the target behavior as the first behavior when the quantity belongs to the second value range and the operation attribute is a second operation attribute; and

determining the target behavior as the third behavior when the quantity belongs to a third value range and the operation attribute is the first operation attribute;

determining the target behavior as the first behavior when the quantity belongs to the third value range and the operation attribute is the second operation attribute; and

determining the target behavior as the second behavior when the quantity belongs to the third value range and the operation attribute is a third operation attribute.

4. The method according to claim 1, wherein the at least two different behaviors comprise a first behavior, a second behavior, and a third behavior; and

the determining the target behavior from the at least two different behaviors based on the quantity and the operation attribute comprises:

determining the target behavior as the first behavior when the operation attribute is a first operation attribute and the quantity is greater than a first threshold;

determining the target behavior as the second behavior when the operation attribute is a second operation attribute and the quantity is greater than a second threshold; and determining the target behavior as the first behavior when the operation attribute is the second operation attribute and the quantity is less than the second threshold and greater than the first threshold; and

determining the target behavior as the third behavior when the operation attribute is a third operation attribute and the quantity is greater than a third threshold; determining the target behavior as the second behavior when the operation attribute is the third operation attribute and the quantity is less than the third threshold and greater than the second threshold; and determining the target behavior as the first behavior when the operation attribute is the third operation attribute and the

quantity is less than the second threshold and greater than the first threshold.

5. The method according to claim 2, 3, or 4, wherein

the first behavior is configured for fighting back against an attack behavior of a second virtual object;

the second behavior is configured for reducing a movement speed of the second virtual object and increasing health points of the first virtual object when health points of the second virtual object are reduced; and

the third behavior is configured for clearing the health points of the second virtual object when the health points of the second virtual object is less than a target value, wherein

the second virtual object is a virtual object in a camp different from that of the first virtual object.

6. The method according to claim 2, 3, or 4, wherein the at least two different behaviors further comprise a fourth behavior, and the fourth behavior is an ordinary attack behavior; and

the determining the target behavior from the at least two different behaviors according to resources associated with the first virtual object comprises:

determining the target behavior as the fourth behavior when the quantity is less than the first threshold;

or

determining the target behavior as the fourth behavior when the quantity is greater than the first threshold and the operation attribute of the touch operation is a fourth operation attribute.

7. The method according to any one of claims 1 to 4, further comprising:

displaying a function switch control, the function switch control being configured for enabling or disabling a function of collecting the resources;

disabling, when the function of collecting the resources is in an enabled state, the function of collecting the resources for the first virtual object in response to a touch operation for the function switch control; and

enabling, when the function of collecting the resources is in a disabled state, the function of collecting the resources for the first virtual object in response to a touch operation for the function switch control.

8. The method according to any one of claims 1 to 4, further comprising:

determining a resource increase quantity of the first virtual object when a resource increase operation is detected; and

updating the quantity of the resources associated with the first virtual object according to the resource increase quantity.

9. The method according to claim 8, wherein the resource increase operation comprises at least one of the following:

the first virtual object killing a second virtual object;

the first virtual object causing health points of the second virtual object to lose;

the first virtual object performing a fifth behavior;

a touch operation for an operation control corresponding to the fifth behavior; or

the first virtual object obtaining a resource increase prop,

wherein the second virtual object is a virtual object in a camp different from that of the first virtual object.

10. The method according to any one of claims 1 to 4, wherein after the controlling the first virtual object to perform the target behavior, the method further comprises:

determining a resource decrease quantity of the first virtual object; and

updating the quantity of the resources associated with the first virtual object according to the resource decrease quantity.

11. A virtual object control apparatus, comprising:

a control display module, configured to display a display picture corresponding to a virtual environment and to display a particular attack control, wherein the particular attack control is configured for controlling a first virtual object in the virtual environment to perform at least two different behaviors;

a behavior determining module, configured to determine, in response to a touch operation for the particular attack control, a target behavior from the at least two different behaviors according to resources associated with the first virtual object, wherein the resources are consumed items for performing a behavior corresponding to the particular attack control; and

a behavior execution module, configured to control the first virtual object to perform the target behavior,

wherein the behavior determining module is further configured to:

obtain a quantity of the resources associated with the first virtual object; and determine the target behavior from the at least two different behaviors according to a value range to which the quantity belongs; or

obtain a quantity of the resources associated with the first virtual object; determine an operation attribute of the touch operation; and determine the target behavior from the at least two different behaviors based on the quantity and the operation attribute.

12. 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, and 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 any one of claims 1 to 10.

13. A computer-readable storage medium, 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 a processor to implement the virtual object control method according to any one of claims 1 to 10.