

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

[0001] This application claims priority to Chinese Patent Application No. 202010018370.3, entitled "VIRTUAL OBJECT CONTROL METHOD AND APPARATUS, TERMINAL, AND
5 STORAGE MEDIUM" filed with China National Intellectual Property Administration on January 8, 2020, which is incorporated by reference in its entirety.

FIELD OF THE TECHNOLOGY

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

BACKGROUND OF THE DISCLOSURE

[0003] At present, shooting games have become more and more popular. In a shooting game, the user controls a virtual object to kill other virtual objects by means of shooting in a virtual environment, thereby winning the game.

15 **[0004]** In the shooting game, the virtual object may hold a virtual firearm, and the virtual object fires virtual bullets to other virtual objects by using the virtual firearm, to cause damage to other virtual objects. When the virtual object fires a plurality of virtual bullets, and cumulative damage to another virtual object reaches a specified value, the other virtual object is switched from an alive state to a death state.

20 **[0005]** However, in the above-mentioned related art, when killing another virtual object, the virtual object needs to fire a plurality of virtual bullets, which takes a long time for a game battle and increases a processing overhead of a server.

SUMMARY

[0006] Embodiments of the present disclosure provide a virtual object control method, a
25 virtual object control apparatus, a terminal, and a storage medium, which can reduce a game battle duration of a shooting game, thereby reducing a processing overhead of a server. The technical solutions are as follows.

[0007] According to an aspect, an embodiment of the present disclosure provides a virtual object control method, applicable to a terminal, the method including:

[0008] displaying a mode selection interface of a shooting game, the mode selection interface including at least one game mode;

[0009] starting, in response to an instruction of selecting a target game mode in the mode selection interface, a game battle of the target game mode, and displaying a battle interface of the game battle;

[0010] controlling a first virtual object to operate in a virtual environment provided for the game battle;

[0011] controlling, in response to a shooting instruction of shooting a virtual firearm used by the first virtual object, the virtual firearm to fire a virtual bullet, a damage value of the virtual bullet being greater than or equal to a hit point value of any virtual object in the game battle; and

[0012] determining, in a case that the virtual bullet hits a second virtual object in the game battle, that the second virtual object is switched from an alive state to a death state.

[0013] According to another aspect, an embodiment of the present disclosure provides a virtual object control apparatus, including:

[0014] an interface display module, configured to display a mode selection interface of a shooting game, the mode selection interface including at least one game mode;

[0015] a battle start module, configured to start, in response to an instruction of selecting a target game mode in the mode selection interface, a game battle of the target game mode, and display a battle interface of the game battle;

[0016] an object control module, configured to control a first virtual object to operate in a virtual environment provided for the game battle;

[0017] a bullet control module, configured to control, in response to a shooting instruction of shooting a virtual firearm used by the first virtual object, the virtual firearm to fire a virtual bullet, a damage value of the virtual bullet being greater than or equal to a hit point value of any virtual object in the game battle; and

[0018] a state switching module, configured to determine, in a case that the virtual bullet hits a second virtual object in the game battle, that the second virtual object is switched from an alive state to a death state.

[0019] According to still another aspect, an embodiment of the present disclosure provides a terminal, including a processor and a memory, the memory storing at least one instruction, at least one program, a code set, or an instruction set, the at least one instruction, where the at least one

program, the code set, or the instruction set is used to, when being loaded and executed by the processor, execute the virtual object control method described above.

[0020] According to yet another aspect, an embodiment of the present disclosure provides 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, where the at least one program, the code set, or the instruction set is used to, when being loaded and executed by a processor, execute the virtual object control method described above.

[0021] According to a further aspect, an embodiment of the present disclosure provides a computer program product, the computer program product, when run on a terminal, causing the mobile terminal to perform the virtual object control method.

[0022] The technical solutions provided in the embodiments of the present disclosure may bring the following beneficial effects:

[0023] After entering a game battle of a target game mode, in a case that a virtual bullet fired by a virtual firearm used by a first virtual object hits a second virtual object, it is determined that the second virtual object is switched from an alive state to a death state. In this way, a product function of killing by one shot is achieved, which helps to improve kill efficiency in the game battle, speed up game process, and reduce a game battle duration, thereby reducing a processing overhead of a server.

BRIEF DESCRIPTION OF THE DRAWINGS

[0024] To describe the technical solutions in embodiments of the present disclosure more clearly, the following briefly describes the accompanying drawings required for describing the embodiments. Apparently, the accompanying drawings in the following description show merely some embodiments of the present disclosure, and a person of ordinary skill in the art may still derive other accompanying drawings from these accompanying drawings without creative efforts.

[0025] FIG. 1 is a schematic diagram of a game running environment according to an embodiment of the present disclosure.

[0026] FIG. 2 is a schematic structural diagram of a terminal according to an embodiment of the present disclosure.

[0027] FIG. 3 is a flowchart of a virtual object control method according to an embodiment of the present disclosure.

[0028] FIG. 4 is an exemplary schematic diagram of a selection interface of a game mode.

[0029] FIG. 5 is an exemplary schematic diagram of a game battle interface.

[0030] FIG. 6 is an exemplary schematic diagram of another game battle interface.

[0031] FIG. 7 is an exemplary schematic diagram of a method for determining whether a virtual bullet hits a target.

5 **[0032]** FIG. 8 is a flowchart of a virtual object control method according to another embodiment of the present disclosure.

[0033] FIG. 9 is an exemplary schematic diagram of a virtual object control method.

[0034] FIG. 10 is a block diagram of a virtual object control apparatus according to an embodiment of the present disclosure.

10 **[0035]** FIG. 11 is a block diagram of a virtual object control apparatus according to another embodiment of the present disclosure.

[0036] FIG. 12 is a structural block diagram of a terminal according to an embodiment of the present disclosure.

DESCRIPTION OF EMBODIMENTS

15 **[0037]** To make the objectives, technical solutions, and advantages of the present disclosure clearer, implementations of the present disclosure are further described below in detail with reference to the accompanying drawings.

[0038] FIG. 1 shows a schematic diagram of a game running environment according to an embodiment of the present disclosure. The game running environment may include a terminal 10
20 and a server 20.

[0039] The terminal 10 may be an electronic device such as a mobile phone, a tablet computer, a game console, an e-book reader, a multimedia player, a wearable device, or a personal computer (PC). A client of an application (such as a game application) may be installed on the terminal 10.

25 **[0040]** In embodiments of the present disclosure, the application refers to a shooting game application. The shooting game application can provide a virtual environment, such that a virtual character played and operated by a user to perform activities in the virtual environment, such as walking and shooting. Typically, the shooting game application may be a third-personal shooting (TPS) game, a first-person shooting (FPS) game, a multiplayer online battle arena (MOBA) game, a
30 multiplayer gunfight survival game, or the like. In addition, virtual objects provided in different

applications have different forms and different functions, which may be preset according to an actual requirement, which is not limited in the embodiments of the present disclosure.

[0041] The 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), for example, a virtual house, a virtual island, or a virtual map. 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, which is not limited in the embodiments of the present disclosure.

[0042] The virtual object is a virtual character controlled by a user account in an application. For example, the application is a game application. The virtual object is a game character controlled by the user account in the game application. The virtual object may be in a human form or animal, cartoon, or other forms, which is not limited in the embodiments of the present disclosure. The virtual object may be presented in a three-dimensional form or a two-dimensional form, which is not limited in the embodiments of the present disclosure. Optionally, in a case that 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 shape and a size in the three-dimensional virtual environment, and occupies some space in the three-dimensional virtual environment.

[0043] The server 20 is configured to provide a back-end service to a client of an application in the terminal 10. For example, the server 20 may be a back-end 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 a back-end service to applications in a plurality of terminals 10 at the same time.

[0044] The terminal 10 may communicate with the server 20 through a network 30. The network 30 may be a wired network or a wireless network.

[0045] Optionally, the terminal 10 includes at least a first terminal and a second terminal. In the embodiments of the present disclosure, the first terminal and the second terminal may be installed with clients of the same application. The client in the first terminal is referred to as a first client, and the client in the second terminal is referred to as a second client. A login user account in the first client is recorded as a first user account, and a login user account in the second client is recorded as a second user account. A virtual object controlled by the first client is referred to as a

first virtual object, and a virtual object controlled by the second client is referred to as a second virtual object, where the first virtual object and the second virtual object have a hostile relationship.

[0046] The hostile relationship refers to the relationship between virtual objects in different camps (or groups). For example, there are 100 users participating in the same game battle, and the 100 users may form a plurality of different groups. For example, each group may have at most four users. Virtual objects controlled by users in the current group have a hostile relationship with virtual objects controlled by users in other groups. For another example, ten users participate in a matching game, and are divided into two teams: a red team and a blue team, that is, every five users form a group. At this time, virtual objects in the red team and virtual objects in the blue team have a hostile relationship.

[0047] In the method embodiments of the present disclosure, an entity executing the steps of the method may be a terminal, for example, the client of the application run on the terminal. 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, and the virtual environment engine may construct three-dimensional virtual environments, virtual objects, virtual props, and the like, to bring more immersive experience to the users.

[0048] FIG. 2 shows a schematic structural diagram of a terminal according to an embodiment of the present disclosure. The terminal 10 may include a mainboard 110, an external output/input device 120, a memory 130, an external interface 140, a touch system 150, and a power supply 160.

[0049] Processing elements such as a processor and a controller are integrated in the mainboard 110.

[0050] Optionally, the external output/input device 120 of the terminal may include a display component (for example, a display screen), a sound playback component (for example, a speaker), a sound collecting component (for example, a microphone), and various buttons. The external output/input device 120 of the PC terminal may include a display component (for example, a display screen), a sound playback component (for example, a speaker), a sound collecting component (for example, a microphone), and various buttons (for example, a mouse and a keyboard).

[0051] The memory 130 stores program code and data.

[0052] The external interface 140 may include an earphone interface, a charging interface, a data interface, and the like.

[0053] The touch system 150 may be integrated in the display component or the buttons of the external output/input device 120, and the touch system 150 is configured to detect touch operations performed by a user on the display component or the buttons.

[0054] The power supply 160 is configured to supply power to other components in the terminal 10.

[0055] In the embodiments of the present disclosure, the processor in the mainboard 110 may generate a user interface (UI) (for example, a game interface) by executing or calling the program code and the data stored in the memory, and present the generated UI (for example, the game interface) by using the external output/input device 120. During presentation of the UI (for example, the game interface), a touch operation performed in interaction between the user and the UI (for example, the game interface) may be detected through the touch system 150, and a response is made to the touch operation.

[0056] FIG. 3 shows a flowchart of a virtual object control method according to an embodiment of the present disclosure. The method may be applied to the game running environment of the terminal 10 shown in FIG. 1, for example, an entity executing the steps of the method may be a client of the application installed on the terminal 10 (hereinafter referred to as the "client"). The method may include the following steps (301 to 305):

[0057] Step 301: Display a mode selection interface of a shooting game.

[0058] The shooting game refers to a battle game in which a user is provided with a virtual firearm as a virtual weapon in a game battle. The user may control a virtual object to use the virtual firearm to attack other virtual objects. The mode selection interface is used for presenting the game mode provided for the shooting game. Optionally, the mode selection interface includes at least one game mode. For example, a plurality of game modes are provided for user selection, where different game modes correspond to different game playing methods and rules.

[0059] In the embodiments of the present disclosure, when a user starts a game, the client displays the mode selection interface of the shooting game. Further, the client starts a game battle in response to obtaining a selection instruction of the user, and displays a battle interface of the game battle. Optionally, the mode selection interface further includes: game mode types, game modes, playing methods and rules, and the like. Exemplarily, referring to FIG. 4, game mode types 41 is displayed in a mode selection interface 40, and the game mode types 41 includes type-A game modes, type-B game modes, and type-C game modes. The type-B game modes include a game mode a, a game mode b, a game mode c and a game mode d. When the user clicks the game mode

d, the mode selection interface 40 displays a game rule 42 corresponding to the game mode d. Optionally, the game rule 42 includes a recommended number of players and a game playing method, and the game playing method is used for prompting the user how to win the game.

[0060] Step 302: Start, in response to a selection instruction of selecting a target game mode in the mode selection interface, a game battle of the target game mode, and display a battle interface of the game battle.

[0061] The selection instruction is an operation instruction for selecting the target game mode from at least one game mode provided in the mode selection interface. Optionally, the generation of the selection instruction may be triggered by the user. For example, the user may click an option corresponding to the target game mode in the mode selection interface, to trigger to generate the selection instruction of selecting the target game mode.

[0062] The game battle refers to an interactive game environment corresponding to the target game mode. A game battle usually takes a certain time. In a game battle of a shooting game, a virtual object may attack other virtual objects through a virtual firearm, to win the game battle. The battle interface refers to a UI used to present a virtual environment of the game battle to the user. The battle interface may include a display screen of the virtual environment, which includes, for example, elements in the virtual environment, such as virtual buildings, virtual props, virtual objects, and the like. Optionally, the battle interface further includes some operation controls such as buttons, sliders, and icons for the user to operate. Optionally, in the embodiments of the present disclosure, the battle interface may further include virtual objects and virtual firearms used by the virtual objects.

[0063] In the embodiments of the present disclosure, the user may select the target game mode in the mode selection interface, to start the game battle. Optionally, the user clicks the target game mode, to trigger to generate the selection instruction of selecting the target game mode.

Further, after the selection instruction is received, the client may start the game battle of the target game mode according to the selection instruction, and display the battle interface of the game battle. Exemplarily, referring to FIG. 4 and FIG. 5 and using a first-person shooting game as an example, by clicking the game mode d in FIG. 4, the user triggers to generate a selection instruction of selecting the game mode d. The client starts a game battle of the game mode d in response to the selection instruction, and displays a battle interface 50 of the game battle. Optionally, the battle interface 50 may display a game rule 51 in the game battle, to prompt the user how to win the game battle.

[0064] In the embodiments of the present disclosure, when the game battle in the target game mode is started, a server may allocate a virtual weapon to each virtual object in the game battle, and the virtual weapon may be a virtual firearm. Optionally, after step 302, the method further includes the following sub-steps:

5 **[0065]** 1. Receive firearm configuration information transmitted by the server.

[0066] 2. Determine a virtual firearm used by a first virtual object according to the firearm configuration information.

[0067] The firearm configuration information refers to related information that the virtual firearm is configured for the corresponding virtual object. For example, the firearm configuration information transmitted by the server to the client corresponding to the first virtual object is used for
10 designating the virtual firearm used by the first virtual object in the game battle. Optionally, the firearm configuration information includes identification information of the virtual firearm. After receiving the firearm configuration information, the client may determine the virtual firearm of the first virtual object according to the firearm configuration information, and display the virtual
15 firearm used by the first virtual object in the battle interface, for example, a virtual firearm 52 in FIG. 5.

[0068] In the embodiments of the present disclosure, the server may configure a virtual firearm for each virtual object in the game battle. In a possible implementation, each virtual object in the game battle is configured to use the same virtual firearm. In this way, each virtual object
20 participating in the same game battle is configured to use the same virtual firearm, which helps to ensure fairness of the game. In another possible implementation, at least two virtual objects in the game battle are configured to use different virtual firearms. In this way, the first virtual object may kill a second virtual object and snatch a virtual firearm used by the second virtual object, thereby making the game more diversified and strategic. The first virtual object and the second virtual
25 object have a hostile relationship.

[0069] Descriptions about virtual weapons configured by the server are only exemplary and explanatory. During actual application, the client may alternatively configure one or more other virtual weapons for the virtual objects in the game battle, such as virtual daggers, virtual explosives, and the like.

30 **[0070]** Step 303: Control the first virtual object to operate in the virtual environment provided for the game battle.

[0071] The first virtual object refers to a user-controlled virtual object that can interact with the virtual environment. Optionally, in the embodiments of the present disclosure, after the client starts the game battle of the target game mode, the user may operate an operation control in the battle interface to generate a corresponding touch operation instruction. Further, the client controls the first virtual object to operate in the virtual environment provided for the game battle according to the touch operation instruction.

[0072] Exemplarily, referring to FIG. 5, the battle interface 50 includes an operation control A 53, an operation control B 54, an operation control C 55, and an operation control D 56. The operation control A 53, the operation control B 54, and the operation control C 55 are skill operation controls of the first virtual object, and the user may click a corresponding operation control, to control the first virtual object to cast a corresponding skill in the virtual environment. The operation control D 56 is a movement operation control of the first virtual object, and the user may slide the operation control D 56, to control the first virtual object to move in the virtual environment.

[0073] Step 304: Control, in response to a shooting instruction for the virtual firearm used by the first virtual object, the virtual firearm to fire a virtual bullet.

[0074] The virtual firearm refers to an offensive virtual prop used by a virtual object. The virtual bullet refers to a virtual prop equipped on the virtual firearm and used for attacking or damaging virtual objects other than the virtual object using the virtual firearm. The shooting instruction refers to a control instruction used for triggering the virtual firearm used by the virtual object to fire the virtual bullet. Optionally, the generation of the shooting instruction is triggered by the user. For example, the user may click an operation control corresponding to the shooting instruction in the battle interface to trigger to generate the shooting instruction. For another example, the user may alternatively press a corresponding key (such as an R key) on a keyboard to trigger to generate the shooting instruction. The key refers to a key corresponding to the operation control corresponding to the shooting instruction. Optionally, the key may be set according to personal habits of the user.

[0075] In the embodiments of the present disclosure, the user may click the operation control corresponding to the shooting instruction to trigger to generate the shooting instruction. Further, after receiving the shooting instruction, the client may control the first virtual object to use the virtual firearm to fire the virtual bullet according to the shooting instruction. In the embodiments of the present disclosure, a damage value of the virtual bullet is greater than or equal to a hit point value of any virtual object in the game battle. Optionally, the damage value of the virtual bullet is greater than or equal to a hit point value of any virtual object in the game battle. In other words, in a

case that each virtual object in the game battle has the same hit point value, which is referred to as a first hit point value, the damage value of the virtual bullet is greater than or equal to the first hit point value; and in a case that the virtual objects in the game battle have different hit point values, a highest hit point value among which is referred to as a second hit point value, the damage value of the virtual bullet is greater than or equal to the second hit point value. In other words, in a case that the virtual bullet hits the second virtual object in the game battle, a hit point value of the second virtual object is reset to zero, that is, the second virtual object is killed. In addition, a kill quantity of the first virtual object in the game battle is increased. The second virtual object may be any virtual object other than the first virtual object in the game battle, and the second virtual object has a hostile relationship with the first virtual object.

[0076] The virtual bullet is equipped to the virtual firearm by the client when configuring the virtual firearm for the first virtual object at the beginning of the game battle. In a possible implementation, a quantity of virtual bullets equipped on the virtual firearm is limited, which may be 1, 2, 3, 4, or the like, and is not limited in the embodiments of the present disclosure.

[0077] In a possible implementation, a condition for responding to the shooting instruction is that a quantity of available virtual bullets of the virtual firearm is not zero. Optionally, after receiving the shooting instruction, the client may detect whether the quantity of available virtual bullets of the virtual firearm is zero. In a case that the quantity of available virtual bullets is not zero, the client responds to the shooting instruction; and in a case that the quantity of available virtual bullets is zero, the client does not respond to the shooting instruction.

[0078] In another possible implementation, the condition for responding to the shooting instruction is that a time length between a moment at which a previous virtual bullet is fired from the virtual firearm and a moment at which the current shooting instruction is received is greater than a preset time length. Optionally, after receiving the shooting instruction, the client may obtain a time length between the moment at which the shooting instruction is received and the moment at which the previous virtual bullet is fired. In a case that the time length is greater than the preset time length, the step of the responding to the shooting instruction is performed, that is, the client controls, in response to the shooting instruction for the virtual firearm used by the first virtual object, the virtual firearm to fire the virtual bullet. That is, each time a shooting instruction is triggered, only one virtual bullet can be fired, i.e., a plurality of virtual bullets cannot be fired without a time interval. In a case that the time length is less than or equal to the preset time length, the client does not respond to the shooting instruction. The preset time length may be 1s, 2s, or 3s,

which is not limited in the embodiments of the present disclosure. The preset time length is less than a time length of automatic recovery of virtual bullets.

[0079] Certainly, in other possible embodiments, the condition for responding to the shooting instruction may be that the quantity of available virtual bullets of the virtual firearm is not zero, and the time length between the moment at which the previous virtual bullet is fired and the moment at which the current shooting instruction is received is greater than the preset time length. Optionally, after receiving the shooting instruction, the client detects whether the quantity of available virtual bullets of the virtual firearm is zero, and obtains the time length between the moment at which the previous virtual bullet is fired and the moment at which the current shooting instruction is received. In a case that the quantity of available virtual bullets is not zero and the time length is greater than the preset time length, the client responds to the shooting instruction; and in a case that the quantity of available virtual bullets is zero or the time length is less than or equal to the preset time length, the client does not respond to the shooting instruction.

[0080] Optionally, in a case that the client does not respond to the shooting instruction, the client may display first prompt information in the battle interface, and the first prompt information is used for prompting the user that use of a virtual bullet for attack is disallowed currently. Optionally, the first prompt information may include a condition for allowing use of a virtual bullet, for example, "an attack using a virtual bullet is allowed n minutes later".

[0081] Step 305: Determine, in a case that the virtual bullet hits the second virtual object in the game battle, that the second virtual object is switched from an alive state to a death state.

[0082] The alive state refers to a state in which a hit point value of a virtual object in a game battle is not zero. Optionally, by using the first virtual object as an example, when the first virtual object is in the alive state, the user may control the first virtual object to perform interactive operations on the virtual environment or the second virtual object. The death state refers to a state in which the hit point value of the virtual object in the game battle is zero. Optionally, by using the first virtual object as an example, when the first virtual object is in the death state, the user cannot control the first virtual object to perform interactive operations. Optionally, under the death state, the user may freely switch a perspective to observe the game battle.

[0083] In the embodiments of the present disclosure, the damage value of the virtual bullet is not less than the hit point value of the virtual object. Therefore, for example, in the case that the virtual bullet hits the second virtual object in the game battle, the client may determine that the second virtual object is switched from the alive state to the death state. Exemplarily, referring to FIG. 6, in a battle interface 60, a second virtual object 61 is in the alive state, the user clicks the

operation control A 53 to trigger to generate a shooting instruction, and the client controls the first virtual object to use the virtual firearm 52 to fire a virtual bullet according to the shooting instruction. In a case that the virtual bullet hits the second virtual object 61, the battle interface 60 is switched to a battle interface 62. The second virtual object 61 is in the death state in the battle interface 62.

[0084] Optionally, the client may determine whether the second virtual object is hit by the virtual bullet by transmitting a detection ray. In the embodiments of the present disclosure, a collision box is disposed on or around the second virtual object, and the collision box is used for detecting whether the virtual object is attacked. The collision box may be cuboidal, cylindrical, spherical, or the like, which is not limited in the embodiments of the present disclosure. Optionally, the client may control the first virtual object to transmit the detection ray from a designated position, where the designated position may be a position from which the virtual bullet is fired, for example, a muzzle of the virtual firearm used by the first virtual object, and a trajectory of the detection ray overlaps with a movement trajectory of the virtual bullet. In a possible implementation, after the client receives the shooting instruction, in a case that the detection ray intersects the collision box of the second virtual object, it is determined that the second virtual object is hit by the virtual bullet.

[0085] Exemplarily, referring to FIG. 7, a detection ray 73 is transmitted in real time, where a muzzle of a virtual firearm 72 used by a first virtual object 71 is used as a start point. A collision box 75 is disposed on a second virtual object 74. When the client receives the shooting instruction, in a case that the detection ray 73 intersects the collision box 75, it is determined that the second virtual object 74 is hit by a virtual bullet, and the second virtual object 74 is controlled to be switched from an alive state to a death state.

[0086] A size of the collision box may be set by the server. In a possible implementation, in a case that any part of the second virtual object is hit by the virtual bullet, the second virtual object is switched from the alive state to the death state. Optionally, the size of the collision box is large enough to cover all body parts of the second virtual object. Alternatively, a quantity of collision boxes is sufficient to cover all body parts of the second virtual object. In another possible implementation, in a case that a specific part of the second virtual object is hit by the virtual bullet, the second virtual object is switched from the alive state to the death state. Optionally, the collision box only covers the specific part, and the specific part may be a head, chest, abdomen, or the like of the second virtual object.

[0087] Optionally, in the embodiments of the present disclosure, the virtual bullet has penetrability. The penetrability means that the virtual bullet may penetrate the second virtual object and continue to fly after hitting the second virtual object. Optionally, after the virtual bullet hits the second virtual object, the client may control the virtual bullet to penetrate the second virtual object, and continuously move in a designated direction; and control the virtual bullet to stop moving in a case that a moving state of the virtual bullet meets a condition.

[0088] The condition refers to a limitation condition of determining whether to stop the virtual bullet. In a possible embodiment, the condition refers to a flying distance of the virtual bullet. Optionally, the flying distance may be set by the server. After the virtual bullet hits the second virtual object, the client may control the virtual bullet to continuously move in the designated direction; and in a case that a moving distance of the virtual object reaches the flying distance, the client controls the virtual bullet to stop flying.

[0089] In another possible embodiment, the condition means that the virtual bullet hits a virtual obstacle, for example, a virtual tree, a virtual building, a virtual wall, or the like. Optionally, after the virtual bullet hits the second virtual object, the client may control the virtual bullet to continuously move in the designated direction; and in a case that the virtual bullet hits the virtual obstacle, the client controls the virtual bullet to stop flying.

[0090] In still another possible implementation, the condition means that the virtual bullet hits n second virtual objects in the flying distance. Optionally, n may be 2, 3, 4, or the like. After the virtual bullet hits the second virtual object, the client may control the virtual bullet to continuously move in the designated direction; and in a case that the virtual object continuously hits the n second virtual objects, the client controls the virtual bullet to stop flying.

[0091] In conclusion, in the technical solution provided in the embodiments of the present disclosure, after entering a game battle of a target game mode, in a case that a virtual bullet fired by a virtual firearm used by a first virtual object hits a second virtual object, it is determined that the second virtual object is switched from an alive state to a death state. In this way, a function of killing by one shot is achieved, which helps to improve kill efficiency in the game battle, speed up game process, and reduce a game battle duration, thereby reducing a processing overhead of a server.

[0092] In addition, a damage value of the virtual bullet is set to be greater than or equal to a hit point value of each virtual object in the game battle, which achieves a product function of killing by one shot (namely, "one-hit kill") without changing other code logic of the game application, thereby significantly reducing development costs of new modes.

[0093] In the embodiments of the present disclosure, a quantity of virtual bullets equipped by the client while configuring the virtual firearm for the first virtual object is limited. Optionally, the virtual bullets may be automatically recovered after consumed. A recovery method of the virtual bullets is introduced below.

5 **[0094]** In an exemplary embodiment, after step 304, the method further includes the following sub-steps:

[0095] 1. Control, in a case that the virtual bullet hits a second virtual object in the game battle, a quantity of available virtual bullets equipped on the virtual firearm to be increased by a first value after a first time period.

10 **[0096]** The first time period refers to a time period for recovering the quantity of available virtual bullets after the client receives a shooting instruction or the first virtual object consumes a virtual bullet to successfully kill the second virtual object. The first value refers to a recovery quantity of virtual bullets.

[0097] Optionally, in the embodiments of the present disclosure, in a case that the first
15 virtual object consumes a virtual bullet and the virtual bullet hits the second virtual object in the game battle, the client may control the quantity of available virtual bullets equipped on the virtual firearm used by the first virtual object to be increased by the first value after the first time period. The first time period may be 0.050s, 0.066s, 0.080s, or the like, and the first value may be 1, 2, 3, or the like. The first value is less than an upper limit value of the quantity of available virtual bullets
20 equipped on the virtual firearm. The upper limit value may be set by the server to 3, 4, 5, or the like, which is not limited in the embodiments of the present disclosure.

[0098] 2. Control, in a case that the virtual bullet does not hit the second virtual object in the game battle, the quantity of available virtual bullets equipped on the virtual firearm to be increased by a second value after a second time period.

25 **[0099]** The second time period refers to a time period for recovering the quantity of available virtual bullets after the client receives the shooting instruction. The second value refers to a recovery quantity of virtual bullets. The second time period is greater than the first time period.

[0100] Optionally, in the embodiments of the present disclosure, in a case that the first
30 virtual object consumes the virtual bullet and the virtual bullet does not hit the second virtual object in the game battle, the client may control the quantity of available virtual bullets equipped on the virtual firearm used by the first virtual object to be increased by the second value after the second time period. The second time period may be 1 min, 3 min, 5 min, or the like, and the second value

may be 1, 2, 3, or the like. The second value is less than the upper limit value of the quantity of available virtual bullets equipped on the virtual firearm. In a possible implementation, the second value is equal to the first value. Optionally, both the second value and the first value are 1.

[0101] In an example, the first time period is 0 and the first value is 1, namely, after the first virtual object uses the virtual firearm to fire a virtual bullet, in a case that the virtual bullet hits the second virtual object, the quantity of available virtual bullets of the virtual firearm is immediately increased by 1. For example, the quantity of available virtual bullets of the virtual firearm is 3, and in a case that the first virtual object hits the second virtual object after a first shooting instruction is generated, the quantity of available virtual bullets is immediately recovered to 3. In a case that the first virtual object does not hit the second virtual object after a second shooting instruction is generated, the quantity of available virtual bullets of the virtual firearm is recovered to 3 when the time period of recovering the virtual bullets reaches the second time period.

[0102] Optionally, a progress icon is displayed in the battle interface during a process of increasing the quantity of available virtual bullets, where the progress icon is used for indicating increment progress of the quantity of available virtual bullets. Exemplarily, referring to FIG. 6, the upper limit value of the quantity of available virtual bullets equipped on the virtual firearm being 3 is used as an example. In the battle interface 60, the quantity of available virtual bullets is 3, and a progress icon 63 is in the form of a full bar at this time. After the first virtual object consumes 3 virtual bullets to kill the second virtual object 61, the quantity of available virtual bullets may be recovered to 1 after the first time period in the battle interface 62, and the form of the progress icon 63 changes with the second time period at this time. In addition, as shown in FIG. 6, the form of a virtual bullet icon may change with the quantity of available virtual bullets. In a case that the quantity of available virtual bullets equipped on the virtual firearm 52 reaches the upper limit value, the quantity of available virtual bullets stops being increased.

[0103] In conclusion, in the embodiments of the present disclosure, a time period for recovering virtual bullets is determined according to whether a virtual bullet hits a virtual object. In a case that the virtual object is hit, the recovery time period is relatively short. Therefore, the user is required to have higher shooting accuracy, which helps to enhance the intensity of the game battle.

[0104] In addition, the increment progress of the quantity of available virtual bullets is displayed during the process of increasing virtual bullets, which helps the user to determine an increasing time of the quantity of available virtual bullets through the increment progress, thereby making the game more strategic.

[0105] In the embodiments of the present disclosure, the client may automatically throw a virtual missile in the game battle. Optionally, a throwing time or throwing location of the virtual missile may be determined by the server, and the virtual missile may be used for increasing competitiveness of a virtual object in the game battle. Optionally, after the first virtual object obtains the virtual missile, generation of an obtaining instruction for the virtual missile may be triggered, and the client may control the first virtual object to obtain a capability additive effect included in the virtual missile according to the obtaining instruction, where the capability additive effect is used for increasing a capability value of the first virtual object, for example, shortening the first time period or the second time period, increasing an attack speed or a movement speed, and the like. Certainly, the virtual missile may alternatively bring a specific gain effect for the virtual object, for example, a stealth skill, teleport skill, or a death-free skill. In other possible implementations, after successfully killing the second virtual object, the first virtual object may alternatively obtain the capability additive effect or the gain effect. Optionally, the capability additive effect or the gain effect may have a time limitation, or have no time limitation. A capability effect of the virtual object is increased through the virtual missile, to improve the intensity of the game battle.

[0106] FIG. 8 shows a flowchart of a virtual object control method according to another embodiment of the present disclosure. The method may be applied to the game running environment of the terminal 10 shown in FIG. 1, for example, an entity executing the method may be a client of the application installed on the terminal 10 (hereinafter referred to as the "client"). The method may include the following steps (801 to 807):

[0107] Step 801: Display a mode selection interface of a shooting game.

[0108] Step 802: Start, in response to an instruction of selecting a target game mode in the mode selection interface, a game battle of the target game mode, and display a battle interface of the game battle.

[0109] Step 803: Control a first virtual object to operate in a virtual environment provided for the game battle.

[0110] Step 804: Control, in response to a shooting instruction of shooting a virtual firearm used by the first virtual object, the virtual firearm to fire a virtual bullet.

[0111] Step 805: Determine, in a case that the virtual bullet hits a second virtual object in the game battle, that the second virtual object is switched from an alive state to a death state.

[0112] Steps 801 to 805 are the same as steps 301 to 305 in the embodiment of FIG. 3. For details, one may refer to description in the embodiment of FIG. 3, which are not described herein again.

[0113] Step 806: Obtain, in response to the end of the game battle, a kill quantity of the first virtual object in the game battle.

[0114] The kill quantity refers to a quantity of enemies killed by a virtual object in a game battle. In the embodiments of the present disclosure, the kill quantity refers to a quantity of second virtual objects killed by the first virtual object in the game battle. Optionally, in a case that a duration of the game battle reaches a preset threshold, the game battle ends. Further, the client obtains the kill quantity of the first virtual object in the game battle.

[0115] Step 807: Display a battle result of the first virtual object in the game battle according to the kill quantity.

[0116] Optionally, the client may sort kill quantities of all virtual objects in the game battle, and display the battle result of the first virtual object in the game battle. The battle result includes the result of sorting. Certainly, the battle result may alternatively include other data of the first virtual object, for example, a hit rate of the first virtual object in the current game battle, a rank of the hit rate of the first virtual object in the current game battle, a capability value of the first virtual object in the current game battle, and a rank of the capability value of the first virtual object in the current game battle.

[0117] In the embodiments of the present disclosure, the killed virtual object may be respawned after a preset time length, and the preset time length may be set by the server. Certainly, the killed virtual object may refuse to be respawned according to actual conditions, which is not limited in the embodiments of the present disclosure.

[0118] In conclusion, in the technical solution provided in the embodiments of the present disclosure, when a game battle ends, a battle result is displayed according to a kill quantity, which increases the intensity of the game battle.

[0119] Exemplarily, referring to FIG. 9, a technical solution of the present disclosure is described by using an example that a game mode is a "one-hit kill" mode.

[0120] Step 901: A client obtains a selection instruction of selecting a "one-hit kill" mode, and displays a corresponding game battle interface according to the selection instruction.

[0121] Step 902: The client detects whether a first virtual object hits a second virtual object. In a case that the first virtual object hits the second virtual object, step 903 is performed; and in a case that the first virtual object does not hit the second virtual object, step 904 is performed.

[0122] Step 903: The client controls the second virtual object to be killed, and controls a quantity of available virtual bullets of the first virtual object to be increased by 1.

[0123] Step 904: The client controls the quantity of available virtual bullets to be reduced by 1.

[0124] Step 905: The client detects whether the quantity of available virtual bullets is zero. In a case that the quantity of available virtual bullets is zero, step 906 is performed; and in a case that the quantity of available virtual bullets is not zero, step 902 is performed.

[0125] Step 906: The client controls the first virtual object so that the first virtual object cannot use a virtual bullet.

[0126] Step 907: The client detects whether increment progress of the quantity of available virtual bullets ends. In a case that the increment progress of the quantity of available virtual bullets ends, step 908 is performed; and in a case that the increment progress of the quantity of available virtual bullets does not end, step 906 is performed.

[0127] Step 908: The client controls the quantity of available virtual bullets to be increased.

[0128] Step 909: The client detects whether the quantity of available virtual bullets reaches an upper limit value. In a case that the quantity of available virtual bullets reaches the upper limit value, step 910 is performed; and in a case that the quantity of available virtual bullets does not reach the upper limit value, step 908 is performed.

[0129] Step 910: The client controls the quantity of available virtual bullets to stop being increased.

[0130] Step 911: The client detects whether a duration of a game battle reaches a preset threshold. In a case that the duration of the game battle reaches the preset threshold, the game ends; and in a case that the duration of the game battle reaches the preset threshold, the game continues.

[0131] The following is an apparatus embodiment of the present disclosure, which can be used to perform the method embodiments of the present disclosure. For details not disclosed in the apparatus embodiments of the present disclosure, one may refer to the method embodiments of the present disclosure.

[0132] FIG. 10 shows a block diagram of a virtual object control apparatus according to an embodiment of the present disclosure. The apparatus has a function of realizing the virtual object control method, and the function may be realized by hardware or by hardware executing corresponding software. The apparatus may be a terminal or may be disposed in a terminal. An apparatus 1000 may include: an interface display module 1101, a battle start module 1002, an object control module 1003, a bullet control module 1004, and a state switching module 1005.

[0133] The interface display module 1001 is configured to display a mode selection interface of a shooting game, the mode selection interface including at least one game mode.

[0134] The battle start module 1002 is configured to start, in response to an instruction of selecting a target game mode in the mode selection interface, a game battle of the target game mode, and display a battle interface of the game battle.

[0135] The object control module 1003 is configured to control a first virtual object to operate in a virtual environment provided for the game battle.

[0136] The bullet control module 1004 is configured to control, in response to a shooting instruction of shooting a virtual firearm used by the first virtual object, the virtual firearm to fire a virtual bullet, a damage value of the virtual bullet being greater than or equal to a hit point value of any virtual object in the game battle.

[0137] The state switching module 1005 is configured to determine, in a case that the virtual bullet hits a second virtual object in the game battle, that the second virtual object is switched from an alive state to a death state.

[0138] In an exemplary embodiment, the bullet control module 1004 is configured to control, in a case that the virtual bullet hits a second virtual object in the game battle, a quantity of available virtual bullets equipped on the virtual firearm to be increased by a first value after a first time period; and control, in a case that the virtual bullet does not hit the second virtual object in the game battle, the quantity of available virtual bullets equipped on the virtual firearm to be increased by a second value after a second time period, where the first time period is less than the second time period.

[0139] In an exemplary embodiment, as shown in FIG. 11, the apparatus 1000 further includes: an icon display module 1006.

[0140] The icon display module 1006 is configured to display a progress icon in the battle interface during a process of increasing the quantity of available virtual bullets, where the progress icon is used for indicating increment progress of the quantity of available virtual bullets.

[0141] In an exemplary embodiment, as shown in FIG. 11, the apparatus 1000 further includes: a quantity detection module 1007 and an instruction response module 1008.

[0142] The quantity detection module 1007 is configured to detect whether the quantity of available virtual bullets is zero after the shooting instruction is received.

5 **[0143]** The instruction response module 1008 is configured to respond to the shooting instruction in a case that the quantity of available virtual bullets is not zero.

[0144] In an exemplary embodiment, as shown in FIG. 11, the apparatus 1000 further includes: a time length obtaining module 1009 and a step execution module 1010.

10 **[0145]** The time length obtaining module 1009 is configured to obtain a time length between a moment at which the shooting instruction is received and a moment at which a previous virtual bullet is fired; and

[0146] The step execution module 1010 is configured to perform the step of the responding to the shooting instruction in a case that the time length is greater than a preset time length.

15 **[0147]** In an exemplary embodiment, as shown in FIG. 11, the apparatus 1000 further includes: an information receiving module 1011 and a firearm determining module 1012.

[0148] The information receiving module 1011 is configured to receive firearm configuration information transmitted by a server, where the firearm configuration information is used for configuring the virtual firearm used by the first virtual object in the game battle.

20 **[0149]** The firearm determining module 1012 is configured to determine the virtual firearm used by the first virtual object according to the firearm configuration information.

[0150] In an exemplary embodiment, each virtual object in the game battle is configured to use the same virtual firearm; or, at least two virtual objects in the game battle are configured to use different virtual firearms.

25 **[0151]** In an exemplary embodiment, as shown in FIG. 11, the apparatus 1000 further includes: a quantity obtaining module 1013 and a result display module 1014.

[0152] The quantity obtaining module 1013 is configured to obtain, in response to the end of the game battle, a kill quantity of the first virtual object in the game battle.

[0153] The result display module 1014 is configured to display a battle result of the first virtual object in the game battle according to the kill quantity.

[0154] In an exemplary embodiment, as shown in FIG. 11, the apparatus 1000 further includes: an effect obtaining module 1015.

[0155] The effect obtaining module 1015 is configured to control, in response to an instruction that the first virtual object obtains a virtual missile, the first virtual object to obtain a capability additive effect included in the virtual missile, where the capability additive effect is used for increasing a capability value of the first virtual object.

[0156] In an exemplary embodiment, the bullet control module 1004 is further configured to control, after the virtual bullet hits the second virtual object, the virtual bullet to penetrate the second virtual object, and continuously move in a designated direction; and control the virtual bullet to stop moving in a case that a moving state of the virtual bullet meets a condition.

[0157] In conclusion, in the technical solution provided in the embodiments of the present disclosure, after entering a game battle of a target game mode, in a case that a virtual bullet fired by a virtual firearm used by a first virtual object hits a second virtual object, it is determined that the second virtual object is switched from an alive state to a death state. In this way, a product function of killing by one shot is achieved, which helps to improve kill efficiency in the game battle, speed up game process, and reduce a game battle duration, thereby reducing a processing overhead of a server.

[0158] When the apparatus provided in the foregoing embodiments implements functions of the apparatus, the division of the foregoing functional modules is merely an example for description. In the practical application, the functions may be assigned to and completed by different functional modules according to the requirements, that is, the internal structure of the device is divided into different functional modules, to implement all or some of the functions described above. In addition, the apparatus and method embodiments provided in the foregoing embodiments belong to the same concept. For the specific implementation process, reference may be made to the method embodiments, and details are not described herein again.

[0159] FIG. 12 shows a structural block diagram of a terminal 1200 according to an embodiment of the present disclosure. The terminal 1200 may be an electronic device, for example, a mobile phone, a tablet computer, a game console, an e-book reader, a multimedia player, a wearable device, a PC, or the like. 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.

[0160] Generally, the terminal 1200 includes a processor 1201 and a memory 1202.

[0161] The processor 1201 may include one or more processing cores. For example, the processor 1201 may be a 4-core processor or an 8-core processor. The processor 1201 may be implemented by using at least one hardware form of a digital signal processor (DSP), a field programmable gate array (FPGA), and a programmable logic array (PLA). The processor 1201 may alternatively include a main processor and a coprocessor. The main processor is configured to process data in an active state, also referred to as a central processing unit (CPU). The coprocessor is a low-power processor configured to process data in a standby state. In some embodiments, 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. In some embodiments, the processor 1201 may further include an artificial intelligence (AI) processor. The AI processor is configured to process computing operations related to ML.

[0162] The memory 1202 may include one or more computer-readable storage media that may be non-transitory. 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.

[0163] 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 by using a bus or a signal cable. Each peripheral may be connected to the peripheral interface 1203 by using a bus, a signal cable, or a circuit board. Specifically, the peripheral includes: at least one of a radio frequency (RF) circuit 1204, a display screen 1205, a camera component 1207, an audio circuit 1207, a positioning component 1208, and a power supply 1209.

[0164] A person skilled in the art may understand that the structure shown in FIG. 12 does not constitute a limitation to 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.

[0165] 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, when executed by a processor, implementing the virtual object control method.

[0166] Optionally, the computer-readable storage medium may include: a read-only memory (ROM), a random access memory (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).

[0167] In an exemplary embodiment, a computer program product is further provided, the computer program product, when executed by a processor, being used for implementing the virtual object control method.

[0168] "Plurality of" mentioned in this specification means two or more. "And/or" describes three relationships between associated objects. 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 "/" in this specification generally indicates an "or" relationship between the associated objects. In addition, the step numbers described in this specification merely exemplarily show a possible execution sequence of the steps. In some other embodiments, the steps may not be performed according to the number sequence. For example, two steps with different numbers may be performed simultaneously, or two steps with different numbers may be performed according to a sequence contrary to the sequence shown in the figure. This is not limited in the embodiments of the present disclosure.

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

CLAIMS

What is claimed is:

1. A virtual object control method, applicable to a terminal, the method comprising:

displaying a mode selection interface of a shooting game, the mode selection interface

5 comprising at least one game mode;

starting, in response to an instruction of selecting a target game mode in the mode selection interface, a game battle of the target game mode, and displaying a battle interface of the game battle;

controlling a first virtual object to operate in a virtual environment provided for the game battle;

controlling, in response to a shooting instruction of shooting a virtual firearm used by the first virtual object, the virtual firearm to fire a virtual bullet, a damage value of the virtual bullet being greater than or equal to a hit point value of any virtual object in the game battle; and

determining, in a case that the virtual bullet hits a second virtual object in the game battle, that the second virtual object is switched from an alive state to a death state.

2. The method according to claim 1, wherein after the controlling, in response to a shooting instruction of shooting a virtual firearm used by the first virtual object, the virtual firearm to fire a virtual bullet, the method further comprises:

controlling, in a case that the virtual bullet hits the second virtual object in the game battle, a quantity of available virtual bullets equipped on the virtual firearm to be increased by a first value after a first time period; and

controlling, in a case that the virtual bullet does not hit the second virtual object in the game battle, the quantity of available virtual bullets equipped on the virtual firearm to be increased by a second value after a second time period, wherein

the first time period is less than the second time period.

3. The method according to claim 2, further comprising:

displaying a progress icon in the battle interface during a process of increasing the quantity of available virtual bullets, wherein

the progress icon is used for indicating increment progress of the quantity of available virtual bullets.

4. The method according to claim 1, further comprising:

detecting whether the quantity of available virtual bullets is zero after the shooting instruction is received; and

5 responding to the shooting instruction in a case that the quantity of available virtual bullets is not zero.

5. The method according to claim 4, wherein before the responding to the shooting instruction, the method further comprises:

obtaining a time length between a moment at which the shooting instruction is received and a moment at which a previous virtual bullet is fired; and

10 performing the operation of responding to the shooting instruction in a case that the time length is greater than a preset time length.

6. The method according to claim 1, wherein after the starting of a game battle of the target game mode, the method further comprises:

15 receiving firearm configuration information transmitted by a server, wherein the firearm configuration information is used for configuring the virtual firearm used by the first virtual object in the game battle; and

determining the virtual firearm used by the first virtual object according to the firearm configuration information.

7. The method according to claim 6, wherein

20 each virtual object in the game battle is configured to use the same virtual firearm;

or,

at least two virtual objects in the game battle are configured to use different virtual firearms.

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

25 obtaining, in response to the end of the game battle, a kill quantity of the first virtual object in the game battle; and

displaying a battle result of the first virtual object in the game battle according to the kill quantity.

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

controlling, in response to an instruction that the first virtual object obtains a virtual missile,

the first virtual object to obtain a capability additive effect comprised in the virtual missile, wherein the capability additive effect is used for increasing a capability value of the first virtual object.

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

controlling, after the virtual bullet hits the second virtual object, the virtual bullet to penetrate
5 the second virtual object, and continuously move in a designated direction; and

controlling the virtual bullet to stop moving in a case that a moving state of the virtual bullet meets a condition.

11. A virtual object control apparatus, comprising:

an interface display module, configured to display a mode selection interface of a shooting
10 game, the mode selection interface comprising at least one game mode;

a battle start module, configured to start, in response to an instruction of selecting a target game mode in the mode selection interface, a game battle of the target game mode, and display a battle interface of the game battle;

an object control module, configured to control a first virtual object to operate in a virtual
15 environment provided for the game battle;

a bullet control module, configured to control, in response to a shooting instruction of shooting a virtual firearm used by the first virtual object, the virtual firearm to fire a virtual bullet, a damage value of the virtual bullet being greater than or equal to a hit point value of any virtual object in the game battle; and

20 a state switching module, configured to determine, in a case that the virtual bullet hits a second virtual object in the game battle, that the second virtual object is switched from an alive state to a death state.

12. The apparatus according to claim 11, further comprising: a bullet increasing module, configured to:

25 control, in a case that the virtual bullet hits the second virtual object in the game battle, a quantity of available virtual bullets equipped on the virtual firearm to be increased by a first value after a first time period; and

control, in a case that the virtual bullet does not hit the second virtual object in the game battle, the quantity of available virtual bullets equipped on the virtual firearm to be increased by a second
30 value after a second time period, wherein

the first time period is less than the second time period.

13. The apparatus according to claim 12, further comprising:

an icon display module, configured to display a progress icon in the battle interface during a process of increasing the quantity of available virtual bullets, wherein

5 the progress icon is used for indicating increment progress of the quantity of available virtual bullets.

14. 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, wherein the at least one instruction, the at least one program, the code set, or the instruction set is used to, when being
10 loaded and executed by the processor, execute the virtual object control method according to any one of claims 1 to 10.

15. 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, wherein the at least one program, the code set, or the instruction set is used to, when being loaded and executed by a processor, execute
15 the virtual object control method according to any one of claims 1 to 10.