

VIRTUAL OBJECT CONTROL METHOD AND APPARATUS, COMPUTER DEVICE, AND STORAGE MEDIUM

RELATED APPLICATION

This application claims priority to Chinese Patent Application No. 202010507467.0, entitled "VIRTUAL OBJECT CONTROL METHOD AND APPARATUS, COMPUTER DEVICE, AND STORAGE MEDIUM" filed with the China National Intellectual Property Administration on June 5, 2020, which is incorporated herein by reference in its entirety.

FIELD OF THE TECHNOLOGY

[0001] Embodiments of this application relate to the field of computer technologies, and in particular, to a virtual object control method and apparatus, a computer device, and a storage medium.

BACKGROUND OF THE DISCLOSURE

[0002] With the development of computer technologies and the widespread popularization of electronic games, a virtual object may be controlled to perform various operations in the electronic game, thereby greatly improving the game effect. A skill casting operation is a common operation. A user may control, according to an aiming direction, the virtual object to cast a skill, but needs to determine the aiming direction before casting the skill.

[0003] The user generally performs a touch operation in a touch region by using a finger, to determine the aiming direction according to a touch position of the touch operation. However, because the finger has a relatively large contact area with the touch region, and it is difficult to control the touch position, the actual touch position is very likely to be inconsistent with a touch position intended by the user, resulting in an inaccurate aiming direction.

SUMMARY

[0004] Embodiments of this application provide a virtual object control method and apparatus, a computer device and a storage medium, to improve the accuracy of an aiming direction.

[0005] The embodiments provide a virtual object control method, including:

[0006] determining, in response to a touch operation on a touch region, at least two touch positions on which the touch operations act, the at least two touch positions being selected from a preset quantity of touch positions on which the touch operations finally act;

[0007] combining the at least two touch positions according to a preset policy, and determining a target touch position of the touch operation;

[0008] determining a first aiming direction indicated by the target touch position; and

[0009] controlling, according to the first aiming direction, a first virtual object to perform a skill casting operation.

[0010] The embodiments further provide a virtual object control apparatus, including:

[0011] a touch position determining module, configured to determine, in response to a touch operation on a touch region, at least two touch positions on which the touch operations act, the at least two touch positions being selected from a preset quantity of touch positions on which the touch operations finally act;

[0012] a target position determining module, configured to combine the at least two touch positions according to a preset policy, and determine a target touch position of the touch operation;

[0013] a first direction determining module, configured to determine a first aiming direction indicated by the target touch position; and

[0014] a first control module, configured to control, according to the first aiming direction, a first virtual object to perform a skill casting operation.

[0015] The embodiments further provide a computer device, including a processor and a memory, the memory storing at least one instruction, the at least one instruction being loaded and executed by the processor to implement the operations performed in the virtual object control method.

[0016] The embodiments further provide a computer-readable storage medium, storing at least one instruction, the at least one instruction being loaded and executed by a processor to implement the operations performed in the virtual object control method.

[0017] In the method and apparatus, the computer device, and the storage medium provided in the embodiments of this application, an aiming direction is no longer determined according to only a last touch position of a touch operation in a touch region, but at least two touch positions on which the touch operations act are determined, and a target touch position

determined according to the at least two touch positions is comprehensively considered, thereby avoiding a case that the last touch position generated due to a misoperation of a user is inconsistent with a touch position intended by the user. If the obtained target touch position can reflect the touch position intended by the user, an aiming direction indicated by the target touch position better meets a requirement of the user, thereby improving the accuracy of the aiming direction. Subsequently, a first virtual object is controlled, according to the determined aiming direction, to perform a skill casting operation, so that more accurate control of the skill casting operation of the first virtual object is also achieved.

BRIEF DESCRIPTION OF THE DRAWINGS

[0018] To describe the technical solutions in the embodiments of this application more clearly, the following briefly introduces the accompanying drawings required for describing the embodiments. Apparently, the accompanying drawings in the following description show only some of the embodiments of this application, and a person of ordinary skill in the art may still derive other accompanying drawings from the accompanying drawings without creative efforts.

[0019] FIG. 1A is an architectural diagram of an applicable implementation environment according to an embodiment of this application.

[0020] FIG. 1B is a flowchart of a virtual object control method according to an embodiment of this application.

[0021] FIG. 2 is a flowchart of another virtual object control method according to an embodiment of this application.

[0022] FIG. 3 is a schematic diagram of a virtual scene interface according to an embodiment of this application.

[0023] FIG. 4 is a schematic diagram of a touch region according to an embodiment of this application.

[0024] FIG. 5 is a schematic diagram of another virtual scene interface according to an embodiment of this application.

[0025] FIG. 6 is a schematic diagram of another virtual scene interface according to an embodiment of this application.

[0026] FIG. 7 is a schematic diagram of another virtual scene interface according to an embodiment of this application.

[0027] FIG. 8 is a schematic diagram of another virtual scene interface according to an embodiment of this application.

[0028] FIG. 9 is a schematic diagram of another virtual scene interface according to an embodiment of this application.

[0029] FIG. 10 is a schematic diagram of another virtual scene interface according to an embodiment of this application.

[0030] FIG. 11 is a schematic diagram of another virtual scene interface according to an embodiment of this application.

[0031] FIG. 12 is a schematic diagram of another virtual scene interface according to an embodiment of this application.

[0032] FIG. 13 is a flowchart of controlling a virtual object to cast a skill according to an embodiment of this application.

[0033] FIG. 14 is a schematic diagram of another touch region according to an embodiment of this application.

[0034] FIG. 15 is a schematic diagram of another touch region according to an embodiment of this application.

[0035] FIG. 16 is a flowchart of determining a target touch position according to an embodiment of this application.

[0036] FIG. 17 is a schematic structural diagram of a virtual object control apparatus according to an embodiment of this application.

[0037] FIG. 18 is a schematic structural diagram of another virtual object control apparatus according to an embodiment of this application.

[0038] FIG. 19 is a schematic structural diagram of a terminal according to an embodiment of this application.

[0039] FIG. 20 is a schematic structural diagram of a server according to an embodiment of this application.

DESCRIPTION OF EMBODIMENTS

[0040] To make objectives, technical solutions, and advantages of the embodiments of this application clearer, the following further describes implementations of this application in detail with reference to the accompanying drawings.

[0041] It may be understood that, the terms "first", "second", and the like used in this application may be used for describing various concepts in this specification. However, the concepts are not limited by the terms unless otherwise specified. The terms are merely used for distinguishing one concept from another concept. For example, without departing from the scope of this application, a first virtual object may be referred to as a second virtual object, and the second virtual object may be referred to as the first virtual object.

[0042] For ease of understanding of the embodiments of this application, nouns related in the embodiments of this application are explained.

[0043] A multiplayer online battle arena (MOBA) is an arena in which at least two opposing camps occupy respective map regions in a virtual scene, and compete against each other using a specific victory condition as an objective. The victory conditions include, but not limited to, at least one of occupying forts or destroy forts of an opposing camp, killing virtual objects in the opposing camp, ensuring own survivals in a specified scenario and time, seizing a specific resource, and outscoring an opponent within a specified time. The battle arena may take place in rounds, and each round of the battle arena may have the same map or different maps. Each virtual team includes one or more virtual objects, for example, 1 virtual object, 2 virtual objects, 3 virtual objects, or 5 virtual objects.

[0044] MOBA game: a game in which a plurality of forts are provided in a virtual scene, and users in different camps control virtual objects to battle in the virtual scene, occupy forts or destroy forts of an opposing camp. For example, in the MOBA game, virtual objects of a plurality of users may be divided into two opposing camps. The virtual objects are scattered in the virtual scene to compete against each other, and the victory condition is to destroy or occupy all enemy forts. The MOBA game takes place in rounds, and a duration of one round of the MOBA game is from a moment at which the game starts to a movement at which a specific camp meets the victory condition. FIG. 1A is an architectural diagram of an applicable implementation environment according to an embodiment of this application. As shown in FIG. 1A, the implementation environment may include a server 10 and a plurality of terminals such as terminals 30, 40, and 50, that communicate with each other through a network 20. The server 10 may be a game server that provides online game services. The terminals 30, 40, and 50 may be computing devices capable of running online games, such as a smartphone, a person computer (PC), a tablet computer, and a game console. Users of a plurality of terminals may access, through the network 20, a same round of online game provided by the server 10 to fight an online battle.

[0045] Virtual object: a movable object in a virtual scene. The movable object may be in any form, for example, is a virtual character, a virtual animal, or a cartoon character. When the virtual scene is a three-dimensional (3D) virtual scene, the virtual objects may be 3D models. Each virtual object has a shape and a volume in the 3D virtual scene, and occupies a part of space in the 3D virtual scene. The virtual object is a 3D character constructed based on a 3D human skeleton technology. The virtual object wears different skins to implement different appearances. In some implementations, the virtual object may be alternatively implemented by using a 2.5-dimensional model or a two-dimensional (2D) model. This is not limited in this embodiment of this application.

[0046] Virtual scene: It is a virtual scene displayed (or provided) by an application program when run on a terminal. The virtual scene may be used for simulating a 3D virtual space, and the 3D virtual space may be an open space. The virtual scene may be a virtual scene that simulates a real environment in reality, or may be a semi-simulated semi-fictional virtual scene, or may be an entirely fictional virtual scene. The virtual scene may be any one of a 2D virtual scene, a 2.5-dimensional virtual scene, and a 3D virtual scene. For example, the virtual scene may include a river, underbrush, land, a building, and the like. The virtual scene is used for a battle between at least two virtual objects, and the virtual scene further includes virtual resources used for the at least two virtual objects. For example, the virtual resources are used for arming the virtual object or are props such as a weapon required to fight a battle with other virtual objects.

[0047] For example, the virtual scene may be a virtual scene in any electronic game. Using the electronic game is a MOBA as an example, the virtual scene is provided with a square map. The square map includes a lower left corner region and an upper right corner region that are symmetrical. Virtual objects belonging to two opposing camps occupy the regions respectively, and an objective of each side is to destroy a target building in an opponent's region to win a game victory.

[0048] The virtual object control method provided in the embodiments of this application is applicable to a plurality of scenarios, for example, is applicable to a battle scene in a game. In the battle scene, the terminal displays a first virtual object, the user controls the first virtual object to perform a skill casting operation, and by using the virtual object control method provided in the embodiments of this application, may determine an aiming direction when the first virtual object performs the skill casting operation.

[0049] FIG. 1B is a flowchart of a virtual object control method according to an embodiment of this application. This embodiment of this application is executed by a terminal, and the terminal may be a portable terminal, a pocket-sized terminal, a handheld terminal, or another type of terminal such as a mobile phone, a computer, or a tablet computer. Referring to FIG. 1B, the method includes the following steps:

[0050] 101. A terminal determines, in response to a touch operation on a touch region, at least two touch positions on which the touch operations act, the at least two touch positions being selected from a preset quantity of touch positions on which the touch operations finally act.

[0051] In this embodiment of this application, the terminal displays a virtual scene interface. The virtual scene interface includes a virtual scene. The virtual scene may include a first virtual object, or may include a river, underbrush, land, a building, a virtual resource used by a virtual object, and the like. In addition, the virtual scene interface may further include a touch button, a touch region, and the like, so that the user may control, by using the touch button or the touch region, the first virtual object to perform operations. For example, the virtual object may be controlled to perform operations such as adjusting a posture, crawling, walking, running, riding, flying, jumping, driving, and picking, or the virtual object may be controlled to perform a skill casting operation or other operations.

[0052] The first virtual object is a virtual object controlled by the user. The virtual scene may further include other virtual objects than the first virtual object, and the other virtual objects may be virtual objects controlled by other users, or may be virtual objects automatically controlled by the terminal, such as a monster, a soldier, and a neutral creature in the virtual scene.

[0053] When performing a skill casting operation, the first virtual object needs to cast a skill on another virtual object, or to cast a skill on a specific direction, or to cast a skill on a specific position. However, regardless of a case, an aiming direction needs to be first determined when a skill is cast.

[0054] In this embodiment of this application, the touch region is used for triggering the skill casting operation, and has a function of adjusting the aiming direction. A finger of the user touches the touch region, and performs the touch operation in the touch region, thereby generating touch positions. The touch positions indicate the aiming direction when the skill is cast. The user may select an intended aiming direction by moving the finger in the touch region. Once the finger of the user performs a lifting action, the terminal may determine an aiming

direction according to a touch position when the finger is lifted, and control, according to the aiming direction, the first virtual object to perform the skill casting operation.

[0055] In the related art, the finger of the user touches the touch region, the finger moves in the touch region, and the user then lifts the finger. In this process, the terminal determines at least two touch positions, and controls, according to an aiming direction indicated by a last touch position, the first virtual object to perform the skill casting operation. However, in an actual application, when the user lifts the finger, the finger may move slightly, resulting in a displacement of the touch position, and a new touch position is generated after the touch position intended by the user, resulting when the aiming direction indicated by the actual last touch position is not the aiming direction intended by the user. In this embodiment of this application, to improve the accuracy of the aiming direction and meet a requirement of the user, at least two touch positions on which the touch operations act may be determined, and the touch positions are comprehensively considered subsequently to determine a more accurate aiming direction.

[0056] 102. The terminal combines the at least two touch positions according to a preset policy, and determines a target touch position of the touch operation.

[0057] 103. The terminal determines a first aiming direction indicated by the target touch position.

[0058] In this embodiment of this application, the terminal determines a target touch position according to the at least two touch positions, and uses an aiming direction indicated by the target touch position as a first aiming direction. Because the at least two touch positions are very likely to include the touch position intended by the user, compared with the last touch position, the target touch position can better reflect the requirement of the user, thereby improving the accuracy of the aiming direction.

[0059] The target touch position is used for indicating the first aiming direction of the first virtual object. The first aiming direction may be any direction in the virtual scene. For example, using the first virtual object as an origin, the first aiming direction may be the left, upper right, lower right, or the like of the first virtual object, or may be represented in a more precise manner. The first aiming direction may be a 30-degree direction, a 90-degree direction, or the like of the first virtual object.

[0060] 104. The terminal controls, according to the first aiming direction, a first virtual object to perform a skill casting operation.

[0061] After determining the first aiming direction, the terminal controls the first virtual object to perform the skill casting operation on the first aiming direction.

[0062] The first virtual object may have different types of skills, for example, may include a direction-type skill, an object-type skill, and a position-type skill. For different types of skills, when the first virtual object is controlled to perform the skill casting operation, for different objects, for example, in the object-type skill, the first virtual object is controlled to perform the skill casting operation on a virtual object in the aiming direction in the virtual scene; in the position-type skill, the first virtual object is controlled to perform the skill casting operation on a specific position in the aiming direction in the virtual scene; and in the direction-type skill, the first virtual object is controlled to perform the skill casting operation on the aiming direction in the virtual scene.

[0063] In the method provided in the embodiments of this application, an aiming direction is no longer determined according to only a last touch position of a touch operation in a touch region, but at least two touch positions on which the touch operations act are determined, and a target touch position determined according to the at least two touch positions is comprehensively considered, thereby avoiding a case that the last touch position generated due to a misoperation of a user is inconsistent with a touch position intended by the user. If the obtained target touch position can reflect the touch position intended by the user, an aiming direction indicated by the target touch position better meets a requirement of the user, thereby improving the accuracy of the aiming direction. Subsequently, a first virtual object is controlled, according to the determined aiming direction, to perform a skill casting operation, so that more accurate control of the skill casting operation of the first virtual object is also achieved.

[0064] FIG. 2 is a flowchart of another virtual object control method according to an embodiment of this application. Referring to FIG. 2, this embodiment is executed by a terminal. Referring to FIG. 2, the method includes the following steps:

[0065] 201. A terminal displays a skill casting button of a first virtual object through a virtual scene interface corresponding to the first virtual object.

[0066] In this embodiment of this application, the virtual scene interface is used for displaying a virtual scene within a field of view of a first virtual object. The virtual scene interface may include a skill casting button of the first virtual object, or may include the first virtual object and other virtual objects, or may include a river, underbrush, land, a building, a virtual resource used by a virtual object, and the like.

[0067] The virtual objects may be divided into a plurality of types of virtual objects. For example, the virtual objects may be divided into a plurality of types according to shapes of the virtual objects or skills of the virtual objects or according to other division standards. For example, if the virtual objects are divided into the plurality of types according to the skills of the virtual objects, the virtual objects may include a fighter-type virtual object, a mage-type virtual object, a support-type virtual object, an archer-type virtual object, and an assassin-type virtual object. In this embodiment of this application, the first virtual object may be any type of virtual object.

[0068] The first virtual object may have one or more skill casting buttons, and different skill casting buttons correspond to different skills. Optionally, the skill casting button includes a text or an image, and the text or the image is used for describing a skill corresponding to the skill casting button. The embodiment of this application is described only by using any skill casting button of the first virtual object as an example.

[0069] For example, referring to a virtual scene interface 300 shown in FIG. 3, the virtual scene interface 300 includes a first virtual object 301 and a second virtual object 302, the first virtual object 301 and the second virtual object 302 belonging to different camps. The virtual scene interface 300 further includes a plurality of skill casting buttons 303, the plurality of skill casting buttons being located at a lower right corner of the virtual scene interface. In addition, an upper left corner of the virtual scene interface displays a complete virtual scene map.

[0070] 202. The terminal displays a touch region through the virtual scene interface in response to a trigger operation on the skill casting button.

[0071] The terminal displays a skill casting button of the first virtual object through the virtual scene interface, the user performs a trigger operation on the skill casting button, and the terminal detects the trigger operation performed by the user on the skill casting button, and displays a touch region corresponding to the skill casting button through the virtual scene interface. The trigger operation may be a click operation, a sliding operation, or another operation.

[0072] If the first virtual object has the plurality of skill casting buttons, and the user performs a trigger operation any skill casting button, the terminal displays a touch region corresponding to the skill casting button through the virtual scene interface.

[0073] The touch region may be in a circle, square, or another shape. The touch region may be located at any position of the virtual scene, for example, at a lower right corner or a lower left corner of the virtual scene.

[0074] In a possible implementation, the touch region includes a first touch subregion and a second touch subregion, the second touch subregion being outside the first touch subregion. The finger of the user touches the touch region. If the user lifts the finger, and a terminal touch position falls within the first touch subregion, the terminal controls the first virtual object to quickly cast a skill; and if the user lifts the finger, and the terminal touch position falls within the second touch subregion, the terminal controls the first virtual object to actively perform aiming to obtain an aiming direction.

[0075] For example, referring to a touch region 400 shown in FIG. 4, the touch region 400 is a circular region, a shaded part is a first touch subregion, a blank part is a second touch subregion, and a dot represents a touch position when the finger presses.

[0076] 203. The terminal determines, in response to a touch operation on the touch region, at least two touch positions on which the touch operations act.

[0077] The terminal displays the touch region, the finger of the user touches the touch region, the terminal detects a touch point corresponding to the finger of the user, the finger of the user moves in the touch region, and a position of the touch point also changes accordingly until the finger of the user lifts. The touch operation on the touch region is completed, and the terminal may determine, by detecting the touch operation, at least two touch positions on which the touch operations act.

[0078] In a possible implementation, when the finger of the user lifts, if a terminal touch position of the touch point falls within the second touch subregion, the terminal determines a target touch position of the touch operation according to the at least two touch positions. That is, if the terminal touch position falls within the second touch subregion, it is considered that the user intends to control, according to the operation of the user, the first virtual object to determine an aiming direction.

[0079] Optionally, when the terminal touch position falls within the second touch subregion, the target touch position determined by the terminal may fall within the second touch subregion, or may fall within the first touch subregion. If the target touch position falls within the second touch subregion, the following step 204 is performed; and if the target touch position

falls within the first touch subregion, the terminal controls the first virtual object to quickly cast the skill.

[0080] In a possible implementation, the touch operations include a press operation of the touch point, a sliding operation of the touch point, and a lifting operation of the touch point. When the finger of the user is in contact with the touch region, the finger of the user is recognized as the touch point. The terminal determines, in response to the press operation corresponding to the touch point in the touch region, an initial touch position corresponding to the press operation; determines at least one intermediate touch position during sliding of the touch point in the touch region; determines, in response to the lifting operation corresponding to the touch point in the touch region, a terminal touch position corresponding to the lifting operation; and determines the preset quantity of touch positions from the at least one intermediate touch position and the terminal touch position. When the terminal detects the press operation of the touch point in the touch region, it indicates that it is ready to perform the skill casting operation. The process of sliding the touch point is an adjustment of the aiming direction. When the terminal detects the lifting operation of the touch point, it indicates that the adjustment of the aiming direction has been completed.

[0081] Optionally, the terminal determines, in response to the press operation of the touch point in the touch region, whether the touch point falls within the touch region. If yes, the terminal determines the initial touch position corresponding to the press operation; and if not, the terminal does not perform the skill casting operation.

[0082] In a possible implementation, to avoid conflicts between touch operations of a plurality of fingers of the user, the terminal assigns a touch identifier to the touch point in response to the press operation corresponding to the touch point in the touch region. Subsequently, a position of the touch point in the touch region may change. If the terminal may detect a touch position and a touch identifier corresponding to the touch position, the terminal determines at least two touch positions that are detected in the touch region and match the touch identifier, to ensure that the determined at least two touch positions belong to the same finger of the user, that is, belong to the same touch operation. The touch point is a position point generated because the finger of the user is in contact with a display screen. The touch identifier may be fingerID or another type of identifier.

[0083] During the touch operation, the position of the touch point when the finger of the user presses is the initial touch position. During sliding of the touch point in the touch region, the touch identifier of the touch point does not change, but the touch position may change, and

further more touch positions may be generated. In addition, the touch position generated by the finger of the user matches the touch identifier of the touch point, a position of the touch point when the finger of the user lifts is the terminal touch position, and the terminal touch position is a last touch position during the touch operation.

[0084] In a possible implementation, for all the touch positions on which the touch operations act, because several initial touch positions have a little impact on the aiming direction when the aiming direction is determined, several last touch positions have a great impact on the aiming direction. Therefore, the determined at least two touch positions on which the touch operations act may include the terminal touch position of the touch operation and at least one touch position before the terminal touch position.

[0085] In a possible implementation, the process in which the user performs the touch operation on the touch region is a dynamic process. When the user performs the touch operation, the terminal acquires the virtual scene interface, to obtain at least two virtual scene interfaces arranged according to an arrangement order. If each virtual scene interface includes a touch position of a touch point, the at least two touch positions are also arranged according to an arrangement order. For example, if the at least two touch positions are arranged according to an arrangement order from the initial touch position to the terminal touch position, the terminal may select, from the at least two touch positions, a preset quantity of touch positions arranged at the end. For example, if two touch positions are selected, the terminal touch position and a previous touch position of the terminal touch position are selected. For example, if the at least two touch positions are arranged according to an arrangement order from the terminal touch position to the initial touch position, the terminal may select, from the at least two touch positions, a preset quantity of touch positions arranged at the top.

[0086] Optionally, the virtual scene interface may be acquired according to a fixed time interval. Correspondingly, a time interval between any two adjacent touch positions in the determined at least two touch positions is the same.

[0087] In a possible implementation, the touch position may be represented by coordinates, and each touch position has corresponding coordinates. By processing coordinates of the at least two touch positions, coordinates of the target touch position may be determined, thereby determining the target touch position. Optionally, a coordinate system is established by using a center of the touch region as a coordinate origin, and each position in the touch region has corresponding coordinates.

[0088] In a possible implementation, the virtual scene interface may further include another touch region of the first virtual object. The another touch region is used for controlling actions of the first virtual object such as moving forward, reversing, and moving, and operations may be performed on both the touch region corresponding to the skill casting button and the another touch region, so that the first virtual object casts the skill while taking an action. Certainly, the virtual scene interface may further include other touch regions. This is not limited in this embodiment of this application.

[0089] Another implementation of step 203 is similar to that of step 101, and details are not described herein again.

[0090] 204. The terminal determines a target touch position of the touch operation according to the at least two touch positions.

[0091] In a possible implementation, the touch position is represented by using the coordinates. The terminal determines weights of the at least two touch positions according to an arrangement order of the at least two touch positions; and performs weighted combination on coordinates of the at least two touch positions according to the weights of the at least two touch positions, to obtain coordinates of the target touch position. The weights of the at least two touch positions may be preset, and a sum of the weights of the at least two touch positions is 1. A quantity of at least two touch positions may alternatively be preset.

[0092] Optionally, the coordinates of the touch position may include a horizontal coordinate and a vertical coordinate. The terminal performs weighted summation on horizontal coordinates of the at least two touch positions according to the weights of the at least two touch positions, to obtain a horizontal coordinate of the target touch position; and performs weighted summation on vertical coordinates of the at least two touch positions according to the weights of the at least two touch positions, to obtain a vertical coordinate of the target touch position.

[0093] For example, if the terminal needs to determine the target touch position according to three touch positions arranged at the end, the terminal presets weights q_1 , q_2 , and q_3 of the three touch positions, where q_1 is a weight of a terminal touch position, q_2 is a weight of a penultimate touch position, and q_3 is a weight of an antepenultimate touch position. If coordinates of the terminal touch position are (x_1, y_1) , coordinates of the penultimate touch position are (x_2, y_2) , and coordinates of the antepenultimate touch position are (x_3, y_3) , a horizontal coordinate of the target touch position is: $x = x_1 * q_1 + x_2 * q_2 + x_3 * q_3$, and a vertical coordinate of the target touch position is: $y = y_1 * q_1 + y_2 * q_2 + y_3 * q_3$, to obtain coordinates of the target touch position (x, y) .

[0094] Optionally, if the at least two touch positions include the terminal touch position and the previous touch position of the terminal touch position, a weight of the terminal touch position may be set to 0, and a weight of the previous touch position of the terminal touch position may be set to 1. The finger of the user presses the touch region. Because the finger of the user has a relatively large contact area with the touch region, when the user lifts the finger, the touch position of the touch point generates a displacement, and a new touch position is generated after the touch position intended by the user. That is, the terminal touch position is very likely to be a touch position generated due to a misoperation of the user, and the previous touch position of the terminal touch position is very likely to be a touch position used by the user to determine the aiming direction. Therefore, the previous touch position of the terminal touch position is determined as the target touch position.

[0095] Another implementation of step 204 is similar to that of step 102, and details are not described herein again.

[0096] 205. The terminal determines a first aiming direction indicated by the target touch position.

[0097] In a possible implementation, a center point of the touch region is used as an origin, a direction in which the target touch position is located at the center point is determined according to the coordinates of the target touch position, and the direction is used as a first aiming direction. Correspondingly, for the first virtual object in the virtual scene interface, the first aiming direction is an aiming direction of the first virtual object. Subsequently, when performing the skill casting operation, the first virtual object casts the skill on the first aiming direction of the first virtual object by using the first virtual object as an origin.

[0098] Another implementation of step 205 is similar to that of step 103, and details are not described herein again.

[0100] 206. The terminal controls, according to the first aiming direction, a first virtual object to perform a skill casting operation.

[0101] After determining the first aiming direction, the terminal controls the first virtual object to perform the skill casting operation on the first aiming direction.

[0102] In a possible implementation, the terminal determines an aiming position in the first aiming direction according to the first aiming direction and a first preset distance; and controls the first virtual object to perform the skill casting operation on the aiming position. A

distance between the aiming position and the first virtual object is the first preset distance, the first preset distance being a skill casting distance set by using the first virtual object as the origin.

[0103] In another possible implementation, the terminal determines a second virtual object that has a shortest distance from the first virtual object in the first aiming direction; and controls the first virtual object to perform the skill casting operation on the second virtual object. After determining the first aiming direction, the terminal may automatically obtain the virtual object in the first aiming direction, and select, according to a position of the obtained virtual object, a virtual object that has a shortest distance from the first virtual object as the second virtual object.

[0104] Optionally, when the second virtual object is a virtual object controlled by another user, according to different skill types, the second virtual object may belong to the same camp with the first virtual object, or may belong to an opposing camp with the first virtual object. For example, if a skill of the first virtual object is an attack-type skill, the second virtual object and the first virtual object may belong to the opposing camps, and the first virtual object performs an attack operation on the second virtual object; if the skill of the first virtual object is a treatment-type skill, the second virtual object may belong to the same camp with the first virtual object, and the first virtual object performs a treatment operation on the second virtual object.

[0105] The embodiment of this application is applicable to a battle scene. The virtual scene of the virtual scene interface includes a first virtual object and a second virtual object. If other virtual objects belong to the same camp with the first virtual object, the user may control the first virtual object to perform a treatment operation to treat the other virtual objects; and if the other virtual objects belong to the opposing camps with the first virtual object, the user may control the first virtual object to perform an attack operation to attack the other virtual objects.

[0106] For example, when the terminal controls the first virtual object to perform the skill casting operation to attack the second virtual object, the virtual scene interface changes as shown in FIG. 5 to FIG. 12.

[0107] Referring to a virtual scene interface 500 shown in FIG. 5, after the user performs a trigger operation on a skill casting button, a circular touch region is displayed in the virtual scene interface 500, and a dot displayed in the touch region represents a current touch position 1. In this case, an aiming direction 1 of a first virtual object 501 is displayed in a virtual scene, and the user may preview the current aiming direction 1 through the virtual scene interface 500.

[0108] A virtual scene interface 600 shown in FIG. 6 is a virtual scene interface displayed after the finger of the user moves in a direction indicated by an arrow in a touch region and is based on the virtual scene interface 500. After the finger moves, a touch position in the touch region changes from the touch position 1 to a touch position 2, and the aiming direction of the first virtual object 501 also changes accordingly from the aiming direction 1 to an aiming direction 2. A virtual scene interface 700 shown in FIG. 7 is a virtual scene interface displayed after the finger of the user moves again in the touch region and is based on the virtual scene interface 600. A touch position in the touch region of the virtual scene interface 700 in FIG. 7 changes again from the touch position 2 to a touch position 3, and the aiming direction of the first virtual object 501 also changes again from the aiming direction 2 to an aiming direction 3. A virtual scene interface 800 shown in FIG. 8 is a virtual scene interface displayed after the finger of the user moves again in the touch region and is based on the virtual scene interface 700. A touch position in the touch region of the virtual scene interface 800 in FIG. 8 changes again from the touch position 3 to a touch position 4, and the aiming direction of the first virtual object 501 also changes again from the aiming direction 3 to an aiming direction 4.

[0109] A virtual scene interface 900 shown in FIG. 9 is a virtual scene interface in which the finger of the user lifts and leaves the touch region and is based on the virtual scene interface 800. In this case, the touch region is no longer displayed in the virtual scene interface 900, and the terminal determines a final aiming direction 5 of the first virtual object 501 according to a determined target touch position.

[0110] A virtual scene interface 1000 shown in FIG. 10 is a virtual scene interface in which the first virtual object 501 starts to cast a skill according to the aiming direction 5 and is based on the virtual scene interface 900. In this case, a preset duration is displayed in a skill casting button, and the preset duration represents a skill cooling duration. Within the preset duration, the user cannot trigger the skill casting button again until the duration is reduced to 0. A virtual scene interface 1100 shown in FIG. 11 is a virtual scene interface in which the first virtual object 501 has cast a corresponding skill and is based on the virtual scene interface 1000. In this case, the duration displayed in the skill casting button is reduced. A virtual scene interface 1200 shown in FIG. 12 is a virtual scene interface in which the skill cast by the first virtual object 501 has attacked the second virtual object and is based on the virtual scene interface 1100. In this case, the duration displayed on an upper layer of the skill casting button is reduced again.

[0111] In addition, all the foregoing embodiments are described by using an example in which the target touch position falls within the second touch region. In a possible

implementation, if the touch point still falls within the first touch subregion from a moment at which the user presses the finger to a moment at which the user lifts the finger, that is, at least two touch positions each fall within the first touch subregion, the terminal determines a second aiming direction according to a preset rule; and controls, according to the second aiming direction, the first virtual object to perform a skill casting operation. The preset rule may be a rule that is preset, and the user may learn about the preset rule according to a related description of the skill, to determine whether to control, according to the preset rule, the first virtual object to cast the skill or to control, by manually adjusting the aiming direction, the first virtual object to cast the skill.

[0112] Optionally, if the preset rule is that the skill is cast on a virtual object that has a distance less than a second preset distance, the terminal determines a position of a third virtual object that has a distance from the first virtual object less than the second preset distance; and determines a second aiming direction according to a position of the first virtual object and the position of the third virtual object, and performs the skill casting operation according to the second aiming direction, to cast the skill on the third virtual object. The first preset distance may be a longest distance that the skill corresponding to the skill casting button may cast.

[0113] Optionally, if there are a plurality of virtual objects other than the first virtual object within a range in which the distance from the first virtual object is less than the first preset distance, any one virtual object may be selected from the plurality of virtual objects as the third virtual object, or a virtual object with minimum health points may be selected from the plurality of virtual objects as the third virtual object.

[0114] FIG. 13 is a flowchart of casting a skill according to an embodiment of this application. Referring to FIG. 13, the procedure of casting the skill includes the following steps:

[0115] 1301. A user triggers a skill casting button, and a terminal displays a touch region, and detects a press operation on a touch point.

[0116] 1302. The terminal is ready to cast a skill.

[0117] 1303. The terminal detects whether the user performs a lifting operation on the touch point. If yes, step 1304 is performed; and if not, step 1305 is performed.

[0118] 1304. Determine a target touch position according to determined at least two touch positions, determine a first aiming direction indicated by the target touch position, and perform a skill casting operation.

[0119] 1305. The terminal determines whether the user performs a sliding operation on the touch point. If yes, step 1303 is performed; and if not, step 1302 is performed.

[0120] The embodiment of this application is described only by using a touch region corresponding to a skill casting button of a first virtual object as an example. In another embodiment, for touch regions of other skill casting buttons of the first virtual object, a manner similar to that of the foregoing embodiments may be used to control the first virtual object to perform the skill casting operation.

[0121] The embodiment of this application is described only by using an example in which an execution body is a terminal. In another embodiment, step 204 and step 205 may be performed by a server connected to the terminal. That is, the terminal determines, in response to the touch operation on the touch region, at least two touch positions on which the touch operations act, and sends the at least two touch positions to the server. The server determines a target touch position of the touch operation according to the at least two touch positions, determines, according to the target touch position, a first aiming direction indicated by the target touch position, and sends the first aiming direction to the terminal. The terminal controls, according to the first aiming direction, the first virtual object to perform a skill casting operation.

[0122] In the related art, referring to a schematic diagram of a touch region shown in FIG. 14, a solid dot in the touch region 1400 represents a terminal touch position 1401 intended by the user, and a hollow dot represents an actual terminal touch position 1402. It can be learned from FIG. 14 that the terminal touch position 1401 intended by the user is greatly different from the actual terminal touch position 1402. If an aiming direction is determined directly according to the actual terminal touch position 1402, the determined aiming direction is greatly different from the aiming direction intended by the user, resulting in failure to achieve an effect intended by the user.

[0123] Referring to a schematic diagram of a touch region shown in FIG. 15, a trajectory shown in the touch region 1501 is a trajectory generated by the touch operation, a solid dot in the touch region 1502 represents a terminal touch position 1511 intended by the user, and a hollow dot represents an actual terminal touch position 1521. By using the method provided in the embodiments of this application, a target touch position 1512 is determined. It can be learned from FIG. 15 that although the terminal touch position 1511 intended by the user is greatly different from the actual terminal touch position 1521, the determined target touch position 1512 is slightly different from the terminal touch position 1511 intended by the user, and an aiming

direction determined according to the target touch position is slightly different from an aiming direction intended by the user, thereby achieving an effect intended by the user.

[0124] In the method provided in the embodiments of this application, an aiming direction is no longer determined according to only a last touch position of a touch operation in a touch region, but at least two touch positions on which the touch operations act are determined, and a target touch position determined according to the at least two touch positions is comprehensively considered, thereby avoiding a case that the last touch position generated due to a misoperation of a user is inconsistent with a touch position intended by the user. If the obtained target touch position can reflect the touch position intended by the user, an aiming direction indicated by the target touch position better meets a requirement of the user, thereby improving the accuracy of the aiming direction. Subsequently, a first virtual object is controlled, according to the determined aiming direction, to perform a skill casting operation, so that more accurate control of the skill casting operation of the first virtual object is also achieved.

[0125] In addition, in this embodiment of this application, the touch region is divided into a first touch subregion and a second touch subregion, so that the user can not only quickly perform the skill casting operation, but also can manually determine an aiming direction and then perform the skill casting operation, and the user may flexibly make a choice according to the skill cast by the first virtual object in the virtual scene, thereby improving the flexibility.

[0126] In addition, in this embodiment of this application, a touch identifier is assigned to the touch point, and the at least two touch positions that match the touch identifier are determined, to ensure that the determined at least two touch positions are touch positions of the same touch operation, thereby avoiding the interference of touch operations of other fingers in the virtual scene interface, and improving the operation accuracy.

[0127] In a possible implementation, FIG. 16 is a flowchart of determining a target touch position according to an embodiment of this application. The procedure includes the following steps:

[0128] 1601. A terminal detects a press operation on a touch point, to obtain an initial touch position, and assigns a touch identifier to the touch point.

[0129] 1602. The terminal determines whether the touch point falls within a touch region.

[0130] 1603. If the touch point falls within the touch region, the terminal assigns coordinates of the initial touch position to (x_n, y_n) , (x_{n-1}, y_{n-1}) ... until (x_1, y_1) . n represents a

quantity of touch positions that needs to be determined, n being a positive integer. If the touch point does not fall within the touch region, this procedure ends.

[0131] 1604. The terminal detects a movement operation on the touch point, to obtain a touch position after movement.

[0132] 1605. The terminal determines whether a touch identifier of the touch position after the movement matches the touch identifier of the foregoing touch point.

[0133] 1606. If yes, the terminal assigns coordinates of the touch position after the movement to (x_n, y_n) . If not, this procedure ends.

[0134] When the user moves the touch point again, step 1604 is repeatedly performed, (x_2, y_2) is assigned to (x_1, y_1) , (x_3, y_3) is assigned to (x_2, y_2) , by analogy, (x_{n-1}, y_{n-1}) is assigned to (x_{n-2}, y_{n-2}) , (x_n, y_n) is assigned to (x_{n-1}, y_{n-1}) , and coordinates of a touch position after the movement again are assigned to (x_n, y_n) .

[0135] 1607. The terminal detects a lifting operation on the touch point, to obtain a terminal touch position.

[0136] 1608. The terminal determines whether a touch identifier of the terminal touch position matches the touch identifier of the foregoing touch point.

[0137] 1609. If yes, the terminal assigns (x_2, y_2) to (x_1, y_1) , assigns (x_3, y_3) to (x_2, y_2) , by analogy, assigns (x_{n-1}, y_{n-1}) to (x_{n-2}, y_{n-2}) , assigns (x_n, y_n) to (x_{n-1}, y_{n-1}) , and assigns coordinates of the terminal touch position to (x_n, y_n) .

[0138] 1610. The terminal calculates coordinates of a target touch position according to coordinates of n touch positions and corresponding weights, to obtain the target touch position. n is a positive integer greater than 1.

[0139] For example, n is 3. That is, three touch positions need to be determined, to obtain coordinates (x_1, y_1) , (x_2, y_2) and (x_3, y_3) of the three touch positions. In the process of determining coordinates of the touch position, assuming that a total of 10 touch positions are obtained, after an initial touch position is obtained, coordinates of the initial touch position are assigned (x_1, y_1) , (x_2, y_2) and (x_3, y_3) ; after a second touch position is obtained, coordinates of the second touch position are assigned to (x_3, y_3) , and after a third touch position is obtained, (x_3, y_3) (the coordinates of the second touch position) is assigned to (x_2, y_2) , and coordinates of the third touch position are assigned to (x_3, y_3) ; and after coordinates of a fourth touch position are obtained, (x_2, y_2) (the coordinates of the second touch position) is assigned to (x_1, y_1) , (x_3, y_3) (the coordinates of the third touch position) is assigned to (x_2, y_2) , and the coordinates of the

fourth touch position are assigned to (x_3, y_3) until a terminal touch position is obtained. (x_2, y_2) (coordinates of an eighth touch position) is assigned to (x_1, y_1) , (x_3, y_3) (coordinates of a ninth touch position) is assigned to (x_2, y_2) , and coordinates of the terminal touch position are assigned to (x_3, y_3) , to obtain coordinates of last three touch positions.

[0140] FIG. 17 is a schematic structural diagram of a virtual object control apparatus according to an embodiment of this application. Referring to FIG. 17, the apparatus includes:

[0141] a touch position determining module 1701, configured to determine, in response to a touch operation on a touch region, at least two touch positions on which the touch operations act, the at least two touch positions being selected from a preset quantity of touch positions on which the touch operations finally act;

[0142] a target position determining module 1702, configured to combine the at least two touch positions according to a preset policy, and determine a target touch position of the touch operation;

[0143] a first direction determining module 1703, configured to determine a first aiming direction indicated by the target touch position; and

[0144] a first control module 1704, configured to control, according to the first aiming direction, a first virtual object to perform a skill casting operation.

[0145] In the apparatus provided in the embodiments of this application, an aiming direction is no longer determined according to only a last touch position of a touch operation in a touch region, but at least two touch positions on which the touch operations act are determined, and a target touch position determined according to the at least two touch positions is comprehensively considered, thereby avoiding a case that the last touch position generated due to a misoperation of a user is inconsistent with a touch position intended by the user. If the obtained target touch position can reflect the touch position intended by the user, an aiming direction indicated by the target touch position better meets a requirement of the user, thereby improving the accuracy of the aiming direction. Subsequently, a first virtual object is controlled, according to the determined aiming direction, to perform a skill casting operation, so that more accurate control of the skill casting operation of the first virtual object is also achieved.

[0146] In a possible implementation, referring to FIG. 18, the target position determining module 1702 includes:

[0147] a weight determining unit 1712, configured to determine weights of the at least two touch positions according to an arrangement order of the at least two touch positions; and

[0148] a coordinate determining unit 1722, configured to perform weighted combination on coordinates of the at least two touch positions according to the weights of the at least two touch positions, to obtain coordinates of the target touch position.

[0149] In another possible implementation, referring to FIG. 18, the coordinates of the target touch position include a horizontal coordinate and a vertical coordinate. The coordinate determining unit 1722 is configured to:

[0150] perform weighted summation on horizontal coordinates of the at least two touch positions according to the weights of the at least two touch positions, to obtain a horizontal coordinate of the target touch position; and

[0151] perform weighted summation on vertical coordinates of the at least two touch positions according to the weights of the at least two touch positions, to obtain a vertical coordinate of the target touch position.

[0152] In another possible implementation, referring to FIG. 18, the touch position determining module 1701 includes:

[0153] a first position determining unit 1711, configured to determine, in response to a press operation corresponding to a touch point in the touch region, an initial touch position corresponding to the press operation;

[0154] a second position determining unit 1721, configured to determine at least one touch position during sliding of the touch point in the touch region; and

[0155] a third position determining unit 1731, configured to determine, in response to a lifting operation corresponding to the touch point in the touch region, a terminal touch position corresponding to the lifting operation.

[0156] In another possible implementation, referring to FIG. 18, the apparatus further includes:

[0157] an identifier assignment module 1705, configured to assign a touch identifier to the touch point in response to the press operation corresponding to the touch point in the touch region; and

[0158] a second position determining unit 1721, configured to determine at least two touch positions that are detected in the touch region and match the touch identifier.

[0159] In another possible implementation, referring to FIG. 18, the first control module 1704 includes:

[0160] an object determining unit 1714, configured to determine a second virtual object that has a shortest distance from the first virtual object in the first aiming direction; and

[0161] a first control unit 1724, configured to control the first virtual object to perform the skill casting operation on the second virtual object.

[0162] In another possible implementation, referring to FIG. 18, the first control module 1704 includes:

[0163] an aiming position determining unit 1734, configured to determine an aiming position in the first aiming direction according to the first aiming direction and a first preset distance, a distance between the aiming position and the first virtual object being the first preset distance; and

[0164] a second control unit 1744, configured to control the first virtual object to perform the skill casting operation on the aiming position.

[0165] In another possible implementation, referring to FIG. 18, the touch region includes a first touch subregion and a second touch subregion, the second touch subregion being outside the first touch subregion.

[0166] The target position determining module 1702 is configured to determine the target touch position of the touch operation according to the at least two touch positions when the terminal touch position falls within the second touch subregion.

[0167] In another possible implementation, referring to FIG. 18, the apparatus further includes:

[0168] a second direction determining module 1706, configured to determine a second aiming direction according to a preset rule when the at least two touch positions all fall within the first touch subregion; and

[0169] a second control module 1707, configured to control, according to the second aiming direction, a first virtual object to perform a skill casting operation.

[0170] In another possible implementation, referring to FIG. 18, the second direction determining module 1706 includes:

[0171] an object position determining unit 1716, configured to determine a position of a third virtual object that has a distance from the first virtual object less than a second preset distance; and

[0172] a second direction determining unit 1726, configured to determine the second aiming direction according to a position of the first virtual object and the position of the third virtual object.

[0173] In another possible implementation, referring to FIG. 18, the apparatus further includes:

[0174] a button display module 1708, configured to display a skill casting button of the first virtual object through a virtual scene interface corresponding to the first virtual object; and

[0175] a touch region display module 1709, configured to display the touch region through the virtual scene interface in response to a trigger operation on the skill casting button.

[0176] All of the above-mentioned optional technical solutions may be combined randomly to form optional embodiments of this application, and details are not described herein again.

[0177] When the virtual object control apparatus provided in the foregoing embodiments controls the virtual object, only division of the foregoing functional modules is used as an example for description. In actual application, the functions may be allocated to and completed by different functional modules according to requirements. That is, an internal structure of the terminal is divided into different functional modules, to complete all or some of the functions described above. In addition, the embodiments of the virtual object control apparatus and the virtual object control method that are provided in the foregoing embodiments belong to the same concept. For a specific implementation process, refer to the method embodiment, and details are not described herein again.

[0178] FIG. 19 is a schematic structural diagram of a terminal 1900 according to an exemplary embodiment of this application. The terminal 1900 may be a portable mobile terminal, for example, a smartphone, a tablet computer, a Moving Picture Experts Group Audio Layer III (MP3) player, a Moving Picture Experts Group Audio Layer IV (MP4) player, a notebook computer, or a desktop computer. The terminal 1900 may also be referred to other names such as user equipment, a portable terminal, a laptop terminal, or a desktop terminal.

[0179] Generally, the terminal 1900 includes a processor 1901 and a memory 1902.

[0180] The processor 1901 may include one or more processing cores, for example, a 4-core processor or an 8-core processor. The processor 1901 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 1901 may alternatively include a

main processor and a coprocessor. The main processor is a processor configured to process data in an awake state, and is 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 1901 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 1901 may further include an artificial intelligence (AI) processor. The AI processor is configured to process computing operations related to machine learning.

[0181] The memory 1902 may include one or more computer-readable storage media. The computer-readable storage media may be non-transient. The memory 1902 may further include a high-speed random access memory (RAM) and a non-volatile memory, for example, one or more disk storage devices or flash memory devices. In some embodiments, a non-transitory computer-readable storage medium in the memory 1902 is configured to store at least one instruction, the at least one instruction being configured to be executed by the processor 1901 to implement the virtual object control method provided in the method embodiments of this application.

[0182] In some embodiments, the terminal 1900 may optionally include a peripheral device interface 1903 and at least one peripheral device. The processor 1901, the memory 1902, and the peripheral device interface 1903 may be connected through a bus or a signal line. Each peripheral device may be connected to the peripheral device interface 1903 by using a bus, a signal cable, or a circuit board. Specifically, the peripheral device includes: at least one of a radio frequency (RF) circuit 1904, a display screen 1905, a camera component 1906, an audio circuit 1907, a positioning component 1908, and a power supply 1909.

[0183] The peripheral device interface 1903 may be configured to connect at least one peripheral device related to input/output (I/O) to the processor 1901 and the memory 1902. In some embodiments, the processor 1901, the memory 1902, and the peripheral device interface 1903 are integrated on the same chip or circuit board. In some other embodiments, any one or two of the processor 1901, the memory 1902, and the peripheral device interface 1903 may be implemented on a separate chip or circuit board. This is not limited in this embodiment.

[0184] In some embodiments, the terminal 1900 may further include one or more sensors 1910. The one or more sensors 1910 include, but not limited to, an acceleration sensor 1911, a gyroscope sensor 1912, a pressure sensor 1919, a fingerprint sensor 1914, an optical sensor 1915, and a proximity sensor 1916.

[0185] A person skilled in the art may understand that the structure shown in FIG. 19 does not constitute a limitation on the terminal 1900, and the terminal may include more components or fewer components than those shown in the figure, or some components may be combined, or a different component deployment may be used.

[0186] FIG. 20 is a schematic structural diagram of a server according to an embodiment of this application. The server 2000 may vary greatly due to different configurations or performance, and may include one or more CPUs 2001 and one or more memories 2002. The memory 2002 stores at least one instruction, the at least one instruction being loaded and executed by the processor 2001 to implement the methods provided in the foregoing method embodiments. Certainly, the server may further include components such as a wired or wireless network interface, a keyboard, and an I/O interface, to facilitate I/O. The server may further include another component configured to implement functions of a device, and details are not described herein again.

[0187] The server 2000 may be configured to perform the steps performed by the server in the foregoing virtual object control method.

[0188] The embodiments of this application further provide a computer device, including a processor and a memory, the memory storing at least one instruction, the at least one instruction being loaded and executed by the processor to implement the operations performed in the virtual object control method according to the foregoing embodiments.

[0189] The embodiments of this application further provide a computer-readable storage medium. The computer-readable storage medium stores at least one instruction, the at least one instruction being loaded and executed by a processor to perform operations performed in the virtual object control method according to the foregoing embodiments.

[0190] The embodiments of this application further provide a computer program, storing at least one instruction, the at least one instruction being loaded and executed by a processor to perform operations performed in the virtual object control method according to the foregoing embodiments.

[0191] A person of ordinary skill in the art may understand that all or some of the steps of the foregoing embodiments may be implemented by hardware, or may be implemented a program instructing related hardware. The program may be stored in a computer-readable storage medium. The storage medium may be: a ROM, a magnetic disk, or an optical disc.

[0192] The foregoing descriptions are merely optional embodiments of the embodiments of this application, but are not intended to limit the embodiments of this application. Any modification, equivalent replacement, or improvement made within the spirit and principle of the embodiments of this application falls within the protection scope of this application.

CLAIMS

What is claimed is:

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

determining, in response to a touch operation on a touch region, at least two touch positions on which the touch operations act, the at least two touch positions being selected from a preset quantity of touch positions on which the touch operations finally act;

combining the at least two touch positions according to a preset policy, and determining a target touch position of the touch operation;

determining a first aiming direction indicated by the target touch position; and

controlling, according to the first aiming direction, a first virtual object to perform a skill casting operation;

wherein the touch region comprises a first touch subregion and a second touch subregion, the second touch subregion being outside the first touch subregion; and the determining a target touch position of the touch operation according to the at least two touch positions comprises:

determining the target touch position of the touch operation according to the at least two touch positions when the terminal touch position falls within the second touch subregion.

2. The method according to claim 1, wherein the determining a target touch position of the touch operation comprises:

determining weights of the at least two touch positions according to an arrangement order of the at least two touch positions; and

performing weighted combination on coordinates of the at least two touch positions according to the weights of the at least two touch positions, to obtain coordinates of the target touch position.

3. The method according to claim 2, wherein the obtaining coordinates of the target touch position comprises:

performing weighted summation on horizontal coordinates of the at least two touch positions according to the weights of the at least two touch positions, to obtain a horizontal

coordinate of the target touch position; and

performing weighted summation on vertical coordinates of the at least two touch positions according to the weights of the at least two touch positions, to obtain a vertical coordinate of the target touch position.

4. The method according to claim 1, wherein the determining at least two touch positions on which the touch operations act comprises:

determining, in response to a press operation corresponding to a touch point in the touch region, at least one intermediate touch position during sliding of the touch point in the touch region;

determining, in response to a lifting operation corresponding to the touch point in the touch region, an terminal touch position corresponding to the lifting operation; and

determining the preset quantity of touch positions from the at least one intermediate touch position and the terminal touch position.

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

assigning a touch identifier to the touch point in response to the press operation corresponding to the touch point in the touch region; and

the determining at least one intermediate touch position during sliding of the touch point in the touch region comprises:

determining at least two touch positions that are detected in the touch region and match the touch identifier.

6. The method according to claim 1, wherein the controlling, according to the first aiming direction, a first virtual object to perform a skill casting operation comprises:

determining a second virtual object that has a shortest distance from the first virtual object in the first aiming direction; and

controlling the first virtual object to perform the skill casting operation on the second virtual object.

7. The method according to claim 1, wherein the controlling, according to the first aiming direction, a first virtual object to perform a skill casting operation comprises:

determining an aiming position in the first aiming direction according to the first aiming direction and a first preset distance, a distance between the aiming position and the first

virtual object being the first preset distance; and

controlling the first virtual object to perform the skill casting operation on the aiming position.

8. The method according to claim 1, wherein after the determining, in response to a touch operation on a touch region, at least two touch positions on which the touch operations act, the method further comprises:

determining a second aiming direction according to a preset rule when the at least two touch positions all fall within the first touch subregion; and

controlling, according to the second aiming direction, the first virtual object to perform the skill casting operation.

9. The method according to claim 8, wherein the determining a second aiming direction according to a preset rule comprises:

determining a position of a third virtual object that has a distance from the first virtual object less than a second preset distance; and

determining the second aiming direction according to a position of the first virtual object and the position of the third virtual object.

10. The method according to claim 1, wherein before the determining, in response to a touch operation on a touch region, at least two touch positions on which the touch operations act, the method further comprises:

displaying a skill casting button of the first virtual object through a virtual scene interface corresponding to the first virtual object; and

displaying the touch region through the virtual scene interface in response to a trigger operation on the skill casting button.

11. A virtual object control apparatus, comprising:

a touch position determining module, configured to determine, in response to a touch operation on a touch region, at least two touch positions on which the touch operations act, the at least two touch positions being selected from a preset quantity of touch positions on which the touch operations finally act;

a target position determining module, configured to combine the at least two touch positions according to a preset policy, and determine a target touch position of the touch

operation;

a first direction determining module, configured to determine a first aiming direction indicated by the target touch position; and

a first control module, configured to control, according to the first aiming direction, a first virtual object to perform a skill casting operation;

wherein the touch region comprises a first touch subregion and a second touch subregion, the second touch subregion being outside the first touch subregion; and

a target position determining module is configured to determine the target touch position of the touch operation according to the at least two touch positions when the terminal touch position falls within the second touch subregion.

12. The apparatus according to claim 11, wherein the target touch position determining module comprises:

a weight determining unit, configured to determine weights of the at least two touch positions according to an arrangement order of the at least two touch positions; and

a coordinate determining unit, configured to perform weighted combination on coordinates of the at least two touch positions according to the weights of the at least two touch positions, to obtain coordinates of the target touch position.

13. The apparatus according to claim 12, wherein the target position determining module comprises:

a weight determining unit, configured to determine weights of the at least two touch positions according to an arrangement order of the at least two touch positions; and

a coordinate determining unit, configured to perform weighted combination on coordinates of the at least two touch positions according to the weights of the at least two touch positions, to obtain coordinates of the target touch position.

14. The apparatus according to claim 11, wherein the touch position determining module comprises:

a first position determining unit, configured to determine, in response to a press operation corresponding to a touch point in the touch region, an initial touch position corresponding to the press operation;

a second position determining unit, configured to determine at least one touch position

during sliding of the touch point in the touch region; and

a third position determining unit, configured to determine, in response to a lifting operation corresponding to the touch point in the touch region, a terminal touch position corresponding to the lifting operation.

15. The apparatus according to claim 11, wherein the first control module comprises:

an object determining unit, configured to determine a second virtual object that has a shortest distance from the first virtual object in the first aiming direction; and

a first control unit, configured to control the first virtual object to perform the skill casting operation on the second virtual object.

16. The apparatus according to claim 11, wherein the first control module comprises:

an aiming position determining unit, configured to determine an aiming position in the first aiming direction according to the first aiming direction and a first preset distance, a distance between the aiming position and the first virtual object being the first preset distance; and

a second control unit, configured to control the first virtual object to perform the skill casting operation on the aiming position.

17. A computer device, comprising a processor and a memory, the memory storing at least one instruction, the at least one instruction being loaded and executed by the processor to implement the operations performed in the virtual object control method according to any one of claims 1 to 10.

18. A computer-readable storage medium, storing at least one instruction, the at least one instruction being loaded and executed by a processor to implement the operations performed in the virtual object control method according to any one of claims 1 to 10.