ISSN: 2688-9323

Research on Virtual VR Device Management System based on Cloud

Yaxin Xing, Yanbin Long

Liaoning University of Science and Technology, China

Abstract

This design discloses a cloud based virtual VR device management system, in which the virtual VR device management method applied in the cloud VR application scenario specifically includes the following sub steps: the server receives the start VR request sent by the client; The server starts the virtual VR device corresponding to the VR application; The server sets the resolution and frame rate of the rendered image driven by virtual VR;The virtual VR device in the server sends the rendered image to the client; The server receives the position, posture and key data of the handle and title sent by the client, and sends the received position, posture and key data of the handle and title to the VR application; The server sends the VR application handle vibration feedback data to the client. On the basis of not affecting the VR application on the server, this design can virtual out the VR device for the application to identify, and when the client sends the corresponding data, the corresponding data can be used as the correct input data to achieve the purpose of cloud.

Keywords

Image Processing; Cloud; Virtual VR Device; Management System.

1. Preface

With the rapid development of computer technology, virtual reality technology has gradually become mature, and more and more electronic products related to virtual reality technology enter people's daily production and life.Virtual reality technology (English name: VirtualReality, abbreviated as VR), also known as spirit technology, is a new practical technology developed in the 20th centuryWith the emergence of this technology, the field of electronic games has gradually begun to use virtual reality technology to enhance the game experience, which can not only play an interesting exercise effect, but also enable users to experience a new game experience. Therefore, virtual VR is popular with more and more users. But there is also a problem in the application of virtual VR, that is, if there is no VR device, The VR application software you want to use cannot be opened, which has caused many users' problems.

Therefore, how to provide a virtual VR device management method that can open VR application software without VR devices is an urgent problem for those skilled in the art.

2. Design Content

This design provides a cloud based virtual VR device management system, which specifically includes the following sub steps: the server receives the start VR request sent by the client;The server starts the virtual VR device corresponding to the VR application according to the start VR request;In response to starting the virtual VR device, the server sets the resolution and frame rate of the rendered image driven by the virtual VR, and makes the resolution and frame rate of the rendered image effective; In response to the completion of the VR application, the virtual VR device in the server sends the rendered image to the client, and the client receives the VR application image; The server receives the position, posture and key data of the handle

ISSN: 2688-9323

and title sent by the client, and sends the received position, posture and key data of the handle and title to the VR application;The server sends the handle vibration feedback data of VR application to the client, and the client receives the handle vibration feedback data.

Starting VR request includes setting the resolution and frame rate required by VR application.

The client is one of VR devices, mobile phones and PC mobile terminals.

The virtual VR device corresponding to the server starting VR application includes sending the handle and head display data to the driver of the virtual VR device through the specified data interface, so that the virtual VR device can be equipped with a head display and a handle.

The specified interface is the API interface of OpenVR.

OpenVR's API is presented as a C++interface class and is a pure virtual function, including TrackedDevicePoseUpdated function, Present function, and GetWindowBounds function.

Create an image texture according to the resolution set in the start VR request, and copy the created image texture to the game texture of the head display screen, so as to achieve the consistency between the resolution of the image data output by the VR application and the resolution set by the client.

According to the frame rate set in the start VR request, the virtual VR drives the creation thread to calculate the rendering time, and complete the frame rate setting of the rendered picture according to this time.

The cloud based virtual VR device management system is characterized in that it includes the server and client, and the server and client conduct data interaction to complete any of the above methods.

The service end comprises a first receiving unit, a starting unit, a setting unit, a first transmitting unit, a second receiving unit and a second transmitting unit;The first receiving unit is used to receive the start VR request sent by the client;Start unit, which is used to start the virtual VR device corresponding to the VR application according to the start VR request;The setting unit is used to set the resolution and frame rate of the rendered image driven by virtual VR, and make the resolution and frame rate of the rendered image effective;A transmitting unit for sending the rendered picture to the client, and the client receives the VR application picture;The second receiving unit is used to receive the position, posture and key data of the handle and title sent by the client, and send the received position, posture and key data of the handle and title to the VR application;The second transmitting unit is used to send the handle vibration feedback data of VR application to the client, and the client receives the handle vibration feedback data.

3. Implementation Mode

This design provides a virtual VR device management method applied to cloud VR application scenarios. This method is an important part of cloud VR applications. By writing drivers, it is believed that there is a VR device on the server side in the VR application scenarios.

This example implements the device driver function based on the driver layer interface provided by OpenVR, and exists in the form of virtual drive, so that it can be considered that there is a VR device on the server in the VR application scenario.

It includes the following steps:

Step S110: The server receives the VR start request sent by the client.

The VR start request sent by the client includes the preset resolution and frame rate that the VR application needs to reach.

Among them, the clients are VR devices, PCs, mobile phones and other mobile terminals.

ISSN: 2688-9323

Step S120: The server starts the virtual VR device corresponding to the VR application according to the start VR request.

On the server side, it implements the Activate function of the ITrackedDeviceServerDriver class of the OpenVR driver layer to realize that the virtual VR device has a head display and a handle.

OpenVR provides a unified data interface for VR games, so that VR games do not have to deal directly with manufacturers' SDKs.OpenVR APIs (Application Program Interface) are presented in the form of C++interface classes, and are pure virtual functions, such as TrackedDevicePoseUpdated function, Present function, GetWindowBounds function, etc.

Call the TrackedDevicePoseUpdated function to send the handle and head display data to the driver of the virtual VR device.

Step S130: In response to starting the virtual VR device, the server sets the resolution and frame rate of the rendered image driven by the virtual VR, and makes the resolution and frame rate of the rendered image effective.

The start VR request sent by the specific client contains the resolution required by the VR application. The image texture is created according to the resolution set, and the game texture rendered to the head display screen is copied according to the created image texture, so that the resolution of the image data output by the VR application is consistent with the resolution set by the client,The GetWindowBounds function of the IVRDisplayComponent class returns the set resolution.

The start VR request sent by the client includes the frame rate that the VR application needs to reach. According to the set frame rate, a thread is created when the virtual VR is driven to calculate the rendering time. For example, if the frame rate set by the client is 60 frames, the server needs 1/60 of about 16ms to render a frame, and notifies the VR application to render the picture by calling the VsyncEvent function.

Step S140: In response to the completion of the VR application, the virtual VR device in the server sends the rendered image to the client, and the client receives the VR application image.

Step S150: The server receives the position, posture and key data of the handle and title sent by the client, and sends the received position, posture and key data of the handle and title to the VR application.

Step S160: The server sends the handle vibration feedback data of VR application to the client, and the client receives the handle vibration feedback data.

The server obtains events through the PollNextEvent interface. The event type is VREvent_Input_HapticVibration refers to the feedback data of handle vibration, and then sends the corresponding feedback data of handle vibration to the client.

Step S160: VR application sends handle vibration feedback according to the preset program,

Among them, steps S110-S140, S150 and S160 are executed simultaneously in parallel. Through the above steps, this embodiment can open VR application software without VR equipment, which is more convenient.

4. Example 2

This embodiment provides a virtual VR device management system applied in the cloud VR application scenario, including a server 210 and a client 220. The server 210 and the client 220 conduct data interaction to complete the above virtual VR device management method applied in the cloud VR application scenario.

The server 210 includes a first receiving unit 310, a starting unit 320, a setting unit 330, a first transmitting unit 340, a second receiving unit 350, and a second transmitting unit 360.

The first receiving unit 310 is used to receive the start VR request sent by the client.

ISSN: 2688-9323

The starting unit 320 is connected with the first receiving unit 310 to start the virtual VR device corresponding to the VR application according to the request for starting VR.

The setting unit 330 is connected with the starting unit 320 to set the resolution and frame rate of the rendered picture driven by the virtual VR, and make the resolution and frame rate of the rendered picture effective.

The transmitting unit 340 may be a virtual VR device, connected with the setting unit 330, for sending the rendered picture to the client, which receives the VR application picture.

The second receiving unit 350 is used to receive the position, posture and key data of the handle and title sent by the client, and send the received position, posture and key data of the handle and title to the VR application.

The second transmitting unit is used to send the handle vibration feedback data of VR application to the client, and the client receives the handle vibration feedback data.

5. Conclusion

Through the data interaction between the client and the server, this design can create VR devices for applications to identify without affecting the VR applications on the server. When the client sends the corresponding data, the corresponding data can be used as the correct input data to achieve the purpose of cloud.

References

[1] Design of library management system based on virtual reality technology [J]Ai Wenhui, Automation Technology and Application, 2020 (08).

[2] Design of real estate exhibition management system based on virtual reality technology [J]Xu Yong;Yan Yubao, Computer Knowledge and Technology, 2009 (32).

[3] Application progress of virtual reality technology in health management [J]Hong Wenjing;Bu Xiaofan;Gold color;Li Shuangshuang;Ailing Liu;Liu Xiangyu. Chinese Journal of Health Management, 2021 (05).

[4] Research progress on application of virtual reality technology in nursing education [J]Chen Saijun. International Journal of Nursing, 2018 (18).

[5] Application of virtual reality technology in the diagnosis and treatment of obsessive-compulsive disorder [J]Yan Zijun;Wang Pengchong;Li Zhanjiang. Chinese Journal of Behavioral Medicine and Brain Science, 2022 (02).

[6] Application of Internet+Virtual Reality Technology Education Mode in Clinical Teaching of Urology [J] Chen Jianfan; Sun Luping; Zhuo Yumin. International Medical and Health Herald, 2020 (21).

[7] Application progress of virtual reality technology in relieving anxiety of children [J]Chen Yingxiang; Sun Jing; Yang Ying; Li Yang. Chinese Journal of Modern Nursing, 2022 (01).

[8] Practical Research on the Application of Virtual Reality Technology in Libraries [J]Li Peng, Electronic Technology and Software Engineering, 2022 (21).

[9] Analysis on the application effect of computer technology to improve virtual reality technology [J]Liu Yang. Information recording materials, 2022 (10).

[10] Power grid emergency rescue safety training system based on virtual reality technology [J] Fengbihong; Hu Lihua; Pan Jin. Industrial Control Computer, 2022 (12).