ATLANTIS PRESS

Advances in Intelligent Systems Research (AISR), volume 145

2017 International Conference on Electronic Industry and Automation (EIA 2017)

A Research on Movement and Detection of Cursors in an Immersive VR Weather Application

Shouyao WANG*
School of Digital Media and Design Arts
Beijing University of Posts and Telecommunications
Beijing, China
E-mail: wangshouyao@bupt.edu.cn
+* Corresponding author

Shenjia HU
School of Computer Science
Beijing University of Posts and Telecommunications
Beijing, China
E-mail: pymess@126.com

***Abstract*—Our Immersive VR Weather Application offers a new method to access weather forecasts. Made by software like Unity and Maya, using VR technologies, the weather data are shown with three-dimensional scenes and human models. This expressive and direct way provides a highly immersive experience for the user. This paper discusses some crucial points in our VR Weather application — the movement detection of the cursor.**

Keywords-virtual reality; cursor movement; interaction detection

I. INTRODUCTION

Weather has a strong influence on human activities and is a hot topic throughout the history of mankind. Accessing weather forecasts has become more convenient nowadays with websites and smartphone applications. As smartphones and the internet prevail, weather service providers actively bring out new methods to satisfy peoples' needs and capture the market. However, traditional smartphone weather applications show everything on two-dimensional planes, mostly with words and diagrams. Some of them managed to provide exquisite computer graphics to visualize the weather, yet that is still less intuitive and visualized than our three-dimensional virtual reality solution.

As the virtual reality (VR) technologies and computer graphics (CG) technologies develop, building an expressive three dimensional weather forecast program becomes more practicable. That is exactly what our application try to achieve.

US weather service provider AccuWeather, together with Samsung Gear VR, has brought out Weather for Life, a VR weather application in which the user can receive real-time weather information updated every minute. Immersive 360 degrees videos are provided for the user to feel the weather [1]. Digital media studio Element X Creative has developed a similar weather application, the Weather Viewer. It is shown in VR mode, too [2]. Though, these two application provided only stereo scene, did not provide other objects – like a dressed model – for the user to watch.

Behold Typhoon VR exhibited in the first Chinese Meteorological Technological Week showed typhoon in all kinds of conditions in VR. In addition to the visual observation provided by the glasses, its special seats can vibrate accordingly, providing more realistic experience [3]. Different to our system, Behold Typhoon VR is specially designed for exhibition of typhoon scenes, not for real-time weather reporting.

Compared to the applications mentioned above, our VR system stands out by its humanoid model, speech bubble and vocal report. Our system provided weather and background sound effects that change according to the weather, enabling the user to feel the weather in an immersive way. Our 3D humanoid model changes its clothes and equipment accordingly, too. A speech bubble floats above the model, showing hints for the current weather. Clicking the bubble plays a vocal version of the hint.

This application builds stereoscopic scenes, human models and sound effects and enables users to "feel" the weather data in an immersive way.

This paper introduces the technology of the movement and detection of the cursor, the interaction method we put forward due to the inability to touch the screen of the phone locked in the Baofeng V VR glasses. The detection allows the user to notice the interactive parts of our application, thus offering a smarter experience.

The Baofeng VR provides a platform VR application, in which the cursor is a white point in the center of the view that morphs to a circle when gazing at interactive parts. In our system, the behavior is quite similar, but our cursor is a red big dot that turns blue on the detection at interactive buttons, much more noticeable.

II. THEORIES OF THE APPLICATION

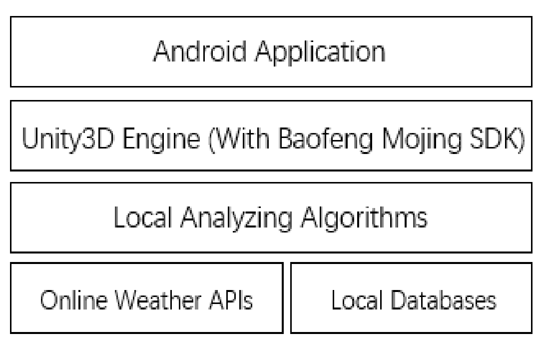

Figure 1. Structure of the application

Databases and online weather APIs (Application Programming Interfaces) serves as the foundation, providing data for the upper algorithms and applications. As we cannot predict the weather ourselves, third part online weather APIs are needed.

Local analyzing algorithms extract the weather data in database and modify the scene accordingly. It tells the engine which scene to use, what clothes the model shall wear, how bright the sun shall be, what the UI board shall show, etc.

The Unity3D Engine integrates the models and scenes built in MAYA and the C# scripts as a project, thus building it to target platform – in this case, and installable Android APK pack.

The top level is the Android application. This enables smartphones run VR applications. Putting the phone into the VR glasses – in this case we use the Baofeng Mojing V – enables the user to watch a stereoscopic view.

The structure of the Android interface is shown in Fig. 2.

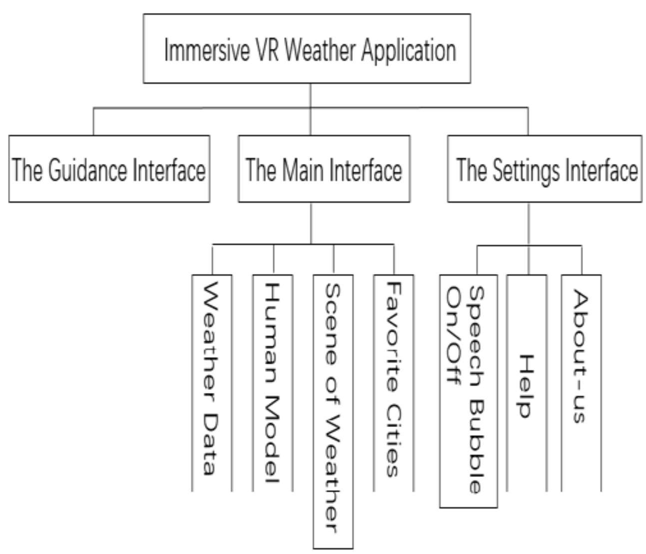

Figure 2. Structure of the android interface

When the application is running, the fundamental structure acquires the weather data from third party API, local analysis algorithm decides which models should be shown via the Unity-made app. Eventually the user can see all kinds of stereoscopic models – the scene, the humanoid, the clothes and the speech bubble, etc.

The Interface in the App, like shown in Fig 2, can be divided into three components – the guidance interface, the main interface, the settings interface.

The guidance interface – appears when the apps is started, serves as a loading screen technically, and serves as a brief hint saying the user should use this app with VR glasses.

The main interface – includes both the floating board and the whole scene. Here the user may inspect weather data in the visualized way and the traditional way. A board floats in the room, where displays the name of the current city, the real-time weather status, a 3-day weather forecast and today's sunrise/sunset time. Clicking the buttons on the board can switch to different favorite cities, and the favorite list can be edited on the board. Details can be seen if needed. A detail board shows five useful details – weather condition, wind power, dressing index, flu index, and sport index. The humanoid and the scene will change accordingly as the city changes or the weather changes.

The settings interface - this interface is a special board that will show when the settings button is clicked. Three buttons are on the board – speech bubble on/off, help and about-us.

The operations are on the board, which suits the habits of users, whereas the weather will also be shown in the 3D scene, providing a straightforward and immersive weather experience.

III. Crucial Technologies

VR technologies aim at offering a new HMI, different from traditional 3D interactive environments; it allows users to participate into the system rather than being an outsider monitoring a screen. This implies that not only complex CD technologies will be involved, but human senses should also a great component.

In our application, a cursor will be displayed at the center of view, much like the fingers for a touch screen or the mouse pointer for PC.

A. Movement of the Cursor

In our application, the user moves the cursor by rotating his or her head.

The VR gears have a head tracking module, which is realized by the MEMS (Micro-Electro-Mechanical System) in the Head Mount Display (HMD). Accelerometers, gyroscopes and magnetometers are coordinated to form a 6-axis or 9-axis MEMS head pose sensor. Expert VR devices reach 60Hz head pose out pose, and Sony PS VR can even reach 120Hz, which is more than enough to synchronize the virtual stereo environment with human poses [4].

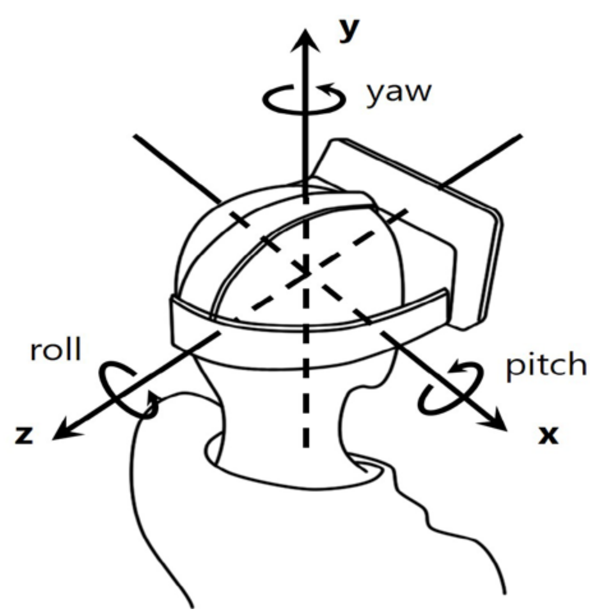

Figure 3. Head pose tracking

The cursor in our application is actually a quad, that is, a flat 2D plane rendered in 3D space. It is designed as always in front of the camera, with a fixed distance, always facing the camera, as if it is fasted to a stick fixed to the camera. With this design, the cursor will look like it is a sticker on the screen. Its movement is realized by the LateUpdate () call in the FocusSensor.cs.

LateUpdate() is called at the end of each frame, after every scripts' Update() call has been finished. Writing this here instead of the Update() call can prevent it from shaking by the different execution order of scripts each frame. The content of LateUpdate sets the position and rotation of the cursor object according to the current human view position.

B. Cursor Detection

In our application, the cursor moves in the virtual space, remaining relative stationary with the head. It is normally red. When the cursor hovers above a button in the view, the cursor will turn blue by changing the material of its renderer, indicating the user may interact by clicking here. Also, the button hovered will turn blue. This function makes the interaction process more interactive.

Unity UI components, like buttons, are driven by Event Systems. With the Baofeng Mojing SDK installed, the camera in the scene will let out an invisible ray. The head rotation is simulated by the camera rotation in the program, and the camera, in turn, moves the ray.

What the ray hits are returned to the event system each frame. Also, UI components like buttons will trigger their OnHover call when hit by this ray, much like what happens when the mouse pointer hovers on buttons. The focus sensor checks the event system each frame. If the hovered objects include any button components – which are the only interactive UI components we used here – the cursor is turned blue, otherwise it will be red.

There are actually two cursor objects, one for each eye. It is because the two eyes has a horizontal distance in between, if they look at the same 3D object, what's after it will be different. It is the same when aiming with a rifle or pistol, we should close one eye. Here we created two cursors, one for each eye, and sets their render layer so each eye can only see the cursor designed for them.

Each cursor object has a Pointer.cs attached to them. To change their color, simply call the Focus and Lost function.

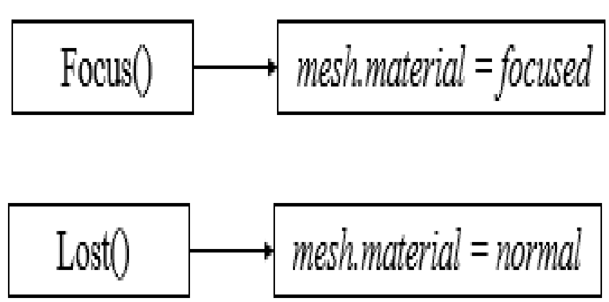

Figure 4. Flowchart about focus function and lost function

FocusSensor.cs has a function isOnButton to decide whether any intractable objects are being hovered. The method is mentioned above. After the isOnButton returns its value, SetFocus or SetNormal will be called.

If isOnButton returns true, then currently the cursor is on a clickable object. Pointer.cs will change the rendering material to the blue material, focused.

If isOnButton returns false, then currently the cursor is on no clickable object. Pointer.cs will change the rendering material to the blue material, normal.

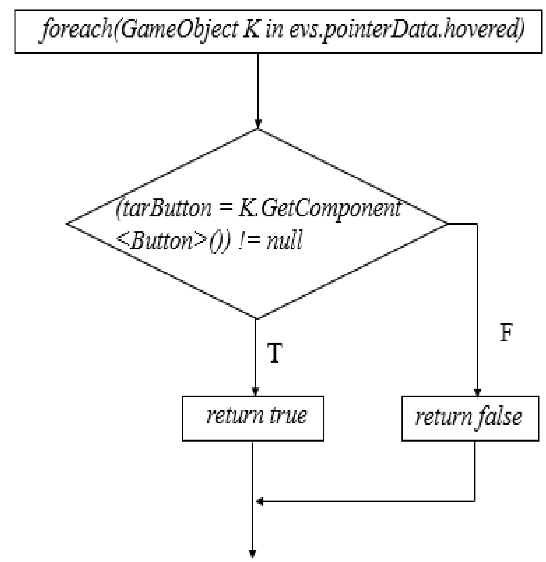

Figure 5. Flowchart about isOnButton. function

Like mentioned above, Update is called once each frame. Writing codes like this enables the intuitive cursor-color-change, or cursor detection.

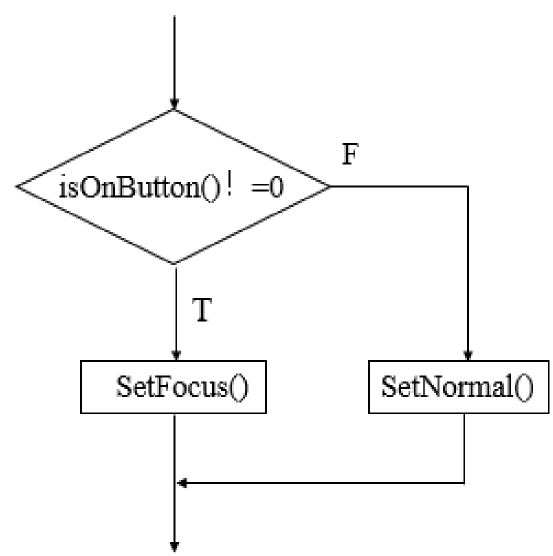

Figure 6. Flowchart about update function

Note that the cursor itself (or themselves) cannot detect anything. It is the event system and the augmentation of the event system by the Baofeng Mojing SDK what detects the button. The cursors just listen and wait for commands.

IV. Conclusion

The paper introduces the cursor movement and detection techniques in our Immersive VR Weather Application. The head mount display detects the pose of the user's head, the event system detects whether there are buttons hovered and sets the color accordingly. We believe that immersive VR weather applications, ours and those of others, will bring better immersive experiences to users, bringing VR farther.

Acknowledgements

The present work is the result of all joint efforts. I am heartily thankful to Beijing University of Posts and Telecommunications Innovation Program "Weather Assistant App Based on VR".

References

[1] AccuWeather, "AccuWeather Launches Samsung Gear Virtual Reality Application, Powered by Oculus," Mar. 2017, http://www.accuweather.com/en/press/64729468.

[2] Jamie Feltham, "Gear VR Gets Weather Viewing App," Oct. 2015, https://www.vrfocus.com/2015/10/gear-vr-gets-weather-viewing-app/.

[3] Guangzhou Daily, "VR brings you experience Typhoon and weather forecast has been accurate to kilometers," May 2017, http://news.dayoo.com/guangzhou/201705/22/150079_51267660.htm.

[4] Cai Bo and Li Yang Mei, 3D audio acquisition and reproduction system using in virtual reality technology. INFORMATION & COMMUNICATIONS,2017 (03): 121-122.