

Development of Unity3D Network Functionality on Android

Shangguan Dayan¹ and Huang Xinyuan^{1,2}

¹College of Art and Design, Beijing Forestry University, China

^{1,2}School of Animation and Digital Arts, Communication University of China,
China
shanggd@bjfu.edu.cn

Abstract

In the field of Mobile Terminal (MT), Unity3D game engine has displayed outstanding rendering power and highly aggregated script editing function while in practice the network function of Unity3D has not been fully developed yet. This paper, based on the comparison with traditional network technology, proposed to invoke Java to create a WIFI Local Area Network (WLAN), in which Unity network view component and remote procedure call (RPC) are employed to synchronize data and images on both ends of the network, thus developing a Unity online game. Finally the author of this paper has designed an online Unity3D chess game based on Android mobile platform. The results indicate that the invocation of Java has made up the deficiency of Unity network while RPC, as a reliable, free and economical method in the network transmission displays great adaptability. The combination of Java and RPC has proved to be a simple but an effective way to develop mobile online game.

Keywords: Unity3D, Android, Network connection

1. Introduction

Unity3D is a multi-platform game development tool as well as a fully integrated game engine, which provides functions such as rendering engine, physics engine, scripting engine, lightmap and scene management and supports these three programming languages, JavaScript, C# and Boo [1-4]. The networking protocol of Unity3D comprises network object management, State Synchronization and RPC. Currently all the network functionality of multiplayer game on Unity3D is based on this protocol. The on-line mode of the game also abides by Unity3D protocol. Firstly WLAN is built through Wi-Fi and on basis of Unity3D network functionality a Client/Server mode double player platform is established in this LAN. Any mobile device in this LAN can create a server and other devices only need to connect to the server to play online as clients. All mobile devices employ the same game code and they can choose to be server or client. In the mode of online game, all the relative data of the game will be transmitted synchronously, which is bidirectional and dependable. This paper, based on a self-development chess game on Android mobile terminal, illustrates implementation plan of online game functionality of a creative 3D chess game supported by Unity3D game engine.

2. Related Work

Despite of Master Server mode which requires connection to the Internet, network communication of Unity3D needs to be in the same LAN. For this reason a WLAN needs to be established. The common wireless technologies are Wi-Fi, Bluetooth and burgeoning NFC.

2.1. Comparison among Three Wireless Technologies

With the development of network and communication technology, wireless communication plays an important role in people's life and close-range wireless communication technology is becoming the focus of attention. The current close-range wireless communication technology includes Bluetooth, ZigBee, WiFi, UWB, IrDA, NFC, etc, which are based on the special requirements of transmission distance and transmission rate or emphasized on the need to the applications such as power consumption and function expansion. None technology can satisfy all the requirements up till the present moment. Figure 1 shows the different wireless communication technologies' transmission rate and transmission distance. As shown in Figure 1.

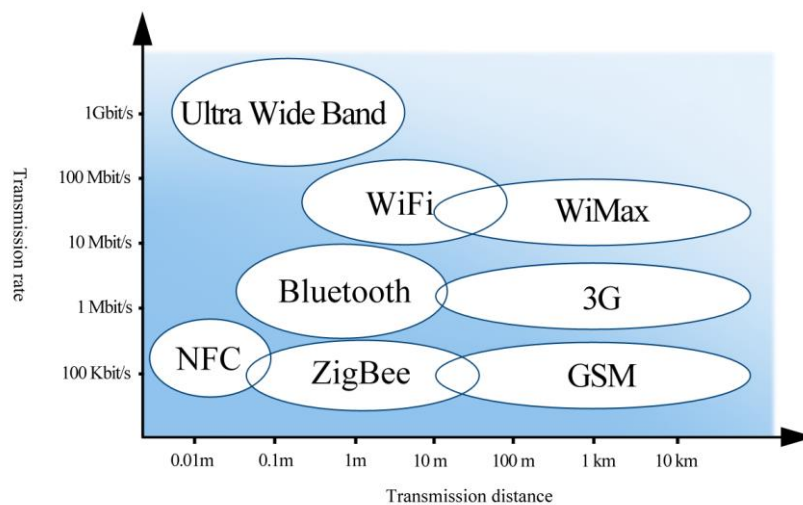


Figure 1. Transmission Rate and Distance of Different Wireless Technologies

Among numerous technologies, Bluetooth turns into the most popular options for close-range wireless connection with high-performance and low-cost by virtue of its characteristics of high-speed frequency hopping and other advanced technologies. So far most of the popular online games on multi mobile devices adopt Bluetooth technology. Bluetooth, by virtue of its advanced technologies like high speed frequency hopping (FH) and advantages of high performance and low cost, has become the most common plan for the close-range wireless connections. Bluetooth takes IEEE802.15 standard, 2.4GHz ISM frequency band and 3Mb/S bandwidth speed. Nowadays, it has been set in the majority of mobile devices. The FH speed rate is 1600 per second and will jump to 3200 per second as multi-slot packets build links [5]. Due to this high speed frequency hopping, Bluetooth has strong ability of anti-interference and multiple accesses. Bluetooth technology can be used on world-wild scale with no need of license and it is known for strong ability for anti signal attenuation, low cost and power consumption.

Near Field Communication (NFC) is a high-frequency wireless communication technology over extremely short distance, typically requiring a distance of 10 cm or less, for electronic devices to exchange data point-to-point without touch [6]. At present, NFC technology has 2 kinds of communication model including passive and active, and 5 basic application patterns-“Touch and Go”, “Touch and Confirm/Pay”, “Touch and Connect”, “Touch and Explore”, “Load and Touch”, *etc.* Heretofore, NFC has been incorporated into the standardization system by international standard organizations. NFC, which builds upon RFID and has downward compatibility, is mainly applied in handheld mobile devices, like providing M2M communications for mobile phones. The most distinguished

point of NFC is its natural security and functionality of quick connection establishment and short-range information exchange. As a kind of short-range wireless technology, NFC now has highly concerned by handset manufacturers, operators around the world and credit card companies rapidly because of its convenient use and consumer appeal.

WIFI is a brand new plan to use wireless access for LAN. As a kind of WLAN, it is commonly referred to IEEE802.11b product, which is equipped with 2.4GHz operating frequency and ISM free band. Compared to Bluetooth and NFC, WIFI is distinguished for its high transmission speed, long effective distance and compatibility with all 802.11DSSS devices [7]. WIFI has the advantages of integration with the existing wired Ethernet networks, lower cost. Compared with traditional LAN cabling cost, WiFi network's construction cost of is lower. As long as a set number of access points installed, WiFi network will meet the signal coverage of the designated area. At present, the demand of WiFi technology in mobile devices, smart home, intelligent home appliances and other fields is increasing day by day.

With larger coverage area and higher transmission speed, built-in WIFI hotspot has become a common function of modern mobile devices. As long as two players are in the same LAN, despite of their rooms or floors, anytime and anywhere they can establish network connections. As shown in Table 1.

Table 1. Comparison Between Three Technologies

	Bluetooth	Wi-Fi	NFC
Communication mode	Active-Active	Active-Active	Active-Active/Positive
Speed	2.1 Mbps	11Mbps	106, 212, 424, 868, 721, 115 kbps, etc.
Distance	0~10m	0~100m	0~20cm
Setting time	6s	3-10s	< 0.1s
Security	Software	Software	Hardware
Network type	Point-to-multipoint	Point-to-multipoint	Point-to-point
Cost	Medium	High	Low
Advantages	Low cost and power consumption, high transmission speed	Wider bandwidth, stronger radio-frequency signal, improved security	Low cost, more convenience, more intuitive
Main application	Mobile phones, game consoles, earphones, stereo audio streaming, automobiles, PC, etc.	Data application such as online games, long distance calls, browsing the Web, downloading audio/video files, etc.	File sharing, contact payment, contact connection, contact browsing, contact downloading, etc.

In conclusion, NFC, due to its short distance and low speed of data transmission, cannot fulfill the requirements of online games. Compared with Bluetooth, WIFI is more adaptable to online game which requires longer distance, more freedom and higher speed. Meanwhile, combined with the needs and characteristics of game developing engine Unity3D, in order to implement online game function, a steady network connection should be established in the WLAN, which could be well fulfilled by WIFI technology.

2.2. WIFI connection in Unity3D

On Android platform, to establish network connection between two mobile terminals through WIFI, we need the support of Java, so as to create a WLAN. Nevertheless, Java cannot be directly used in Unity3D engine. Hence invocation is needed here. In order to invoke Java in Unity3D, activity module will be used to provide Unity interface in Java project. Detailed invocation method is summarized as follows:

- XiangQiActivity, inheritance to UnityPlayerActivity class, is marked as the main Activity, in which the onCreate(Bundle savedInstanceState), StartActivity0 (String name), StartActivity1(String name) need to be rewritten. StartActivity0(String name) and StartActivity1(String name) are invoked by C# script. Invoke background program to open a new Activity;
- Create two classes. WifiActivity1.java, WifiActivity2.java, which inherit to Activity class and transmit string parameters to new Activity. Obtain parameters through this. getIntent(). getStringExtra("name") differentiate two Activity;
- Write a layout file, main.xml and a configuration file AndroidManifest.xml;
- Integrate Java files and lay them in the Plugins->Android folder in Unity project. In the Unity project, create C# script and bind to camera. In this way, the code of Java could be invoked.

To create a LAN for the game, one player needs to establish AP as a server while the other scans the AP information and connects as a client, thus forming a network virtual link between two Android terminals based on TCP. Finally, through WIFI connection, Android client is in the same LAN with Android server.

3. Implementation of Online Game in Unity3D

3.1. Network Functionality of Unity3D

Instead of being self-existent, the network protocol of Unity3D, along with game objects of the engine, provides network function support for games. This means that rather than simply sending messages to clients or servers, all network communication of Unity3D is related to a certain game object. In a scene, all the virtual objects are controlled by scene editor of Unity3D which provides a C# container class related to geometry, transformation nodes, textures, physical properties and C# class to control virtual performance and behaviors of virtual scene object. This C# container class is called Game Object [8]. All the objects in the game are game objects, such as "pawn", "knight" and "gun" and other game roles controlled by players. As shown in Figure 2.



Figure 2. The "Knight" of Game Object

Game object is the container of all the other components. In order to adopt the network functionality of Unity3D, Network View component must be attached to game objects. Once the attachment is established, the game objects on each of the hosts will build up

communication connection automatically. The game object on any host can communicate with “itself” on other hosts and this communication is independent.

3.1.1. Component: Components are used to attach to game objects. For example, we attach a Camera component to Main Camera object and turn this game object to camera object, thus showing the game world to the clients. To every game object available to be controlled by players (red and blue chess pieces in the game), Animation component, Character Controller component and Script component have also been attached to manage the animation play, movement in the crashing condition and clients related operation processing. It's worth noting that in Unity3D, all the scripts are programmed and saved as a special kind of component and therefore they can be attached to game objects.

In this game, Network View, as the component closely related to the network function of online game, needs to be attached to each game object controlled by players (32 red and blue chess pieces) and to QiPlayer game object which controls game rules and the whole chessboard so as to realize the picture synchronization of client and server.

3.1.2. Network View: Network View is a component to transmit data in the network. In the same project, every Network View component is through Network View ID as the only mark, which can be automatically allocated.

In Unity3D network transmission, every data packet will be sent to a certain object linked by Network View which is further specified by Network View ID. Unity3D can trace the Network View ID, locate the game object observed by corresponding Network View and unpack the data packet to apply to the game object.

Through the Network View, we can start two types of network communication between game objects: state synchronization and RPC. Before starting network communication, network connection of Unity3D channel needs to be established first.

3.2. Establishment of Network Connection

Unity3D can be applied to various types of network connection, among which the most common ones are Master Server mode connection and Client/Server direct connection, etc. In this game C/S direct connection is adopted for a WLAN has already been built through the invocation of Java. The Script components to establish network connection are all attached to the corresponding game objects in game scene Two. According to clients' operation, we can load the game scene JZX or return to the game Menu scene. The network connection of Unity3D is controlled by build-in Network and can be obtained by Network.peerType. Entering into scene Two, everyone's network is disconnected. Clients can choose the color of the chess pieces that they want to play and the client who plays red ones is the server while the one who plays blue ones is the client.

Click the red headshot and run NetWork Server script.

```
STATUS OnClick()
CALL Network.InitializeServer(maxcnt, port, false)
/*initialization*/
RETURN
STATUS Update()
IF(Network.peerType==NetworkPeerType.Server) /*If the server has been
started*/
lable.text<—Network.player.ipAddress /*Show server's IP
address*/
IF(Network.connections.Length==allowed) /*If client connects to this
server*/
CALL Application.LoadLevel(JZX) /*Loading game scene
JZX*/
```

```
ENDIF  
ENDIF  
RETURN
```

In the `OnClick` of script, `Network.InitializeServer` is used to initialize the server. It has three parameters: the first one represents the maximal connections (exclusive of itself), the second one is port number while the third one is whether using NAT or not. At this time, we begin to monitor the certain Port and network state turns to Server.

When creating script under Assets catalogue in Unity3D, there will be an `Update` method which will be invoked in very frame [9]. If the server has been started, it will show server's IP address and meanwhile waits for client's connection. Once a client connects to this server, the network connection is accomplished and game scene `JZX` will begin loading.

If we choose log in as clients, we need to type in the server's IP address in the textbox below the blue chess pieces. Click the "blue" button to run `NetWork Client` script.

```
STATUS OnClick()  
ip<—CALL input.text()  
CALL Network.Connect(ip, port) /*Connect to the  
server */  
RETURN  
  
STATUS Update()  
IF(Network.peerType==NetworkPeerType.Client) /*Successfully  
connected*/  
CALL Application.LoadLevel(JZX) /*Loading game scene  
JZX*/  
ENDIF  
RETURN
```

Among the above, `input` is an example of user-defined public class `UIInput` and the text method is used to obtain the content in the textbox. `Network.Connect` is taken to connect to the server of appointed IP and Port. If connection is successful, the network state will turn to `Client`, which represents that both sides have connected to the network. `Client` begins to start loading game scene `JZX`.

The network connection state of will keep being connected until the `Network Disconnect` in the `ButtonBreak` script is invoked to disconnect the connection. `ButtonBreak` script is attached to the `Return Button` game objects in scene `Two` and scene `JZX`.

3.3. Network Data Transmission

After the connection between server and client is established, game scene synchronization and the transmission of core data can be started. Network data transmission mainly occurs in the `JZX` game scene. In the online mode of the game, server and client have their own set of objects supporting the game which means that the game objects of the whole chessboard and chess pieces are generated by server and client respectively. In order to enable the two players to see the same game, all these game objects need to be synchronized by data sharing. In this game, the two players are logically equal and bilateral synchronization must be carried out. Hence the `State Synchronization` of `Network View` component is set to be off, which means that here we do not use state synchronization (client executes events and send the results to server, which merely transmits). Instead another communication mode: `Remote Procedure Calls (PRC)` is adopted which is dependable, well-organized, more economical and with no limitations on transmission direction.

2.3.1. Remote Procedure Calls: Online chess games, different from racing games, instead of continuous synchronization of involved game objects' state, the locations where the chess pieces start and stop, the time for two players to take turn and the game result are all the information we need to transmit. Therefore after game objects in one host make move, the other hosts in the network should be informed to execute RPC function to repeat this one-time game event.

RPC is a function claiming to be [RPC] (JS is @RPC) in a script attached to a game object which includes Network View component.

```
[RPC]
STATUS accessSelector(String name,INTi,INTj,VECTOR3 post,INTtag)
IF(tag!=cancel)
obj<— CALL GameObject.Find(name)
CALL FingerTap()
ELSE
CALLFingerLongPress()
ENDIF
RETURN
```

This RPC function can be invoked by any function of any other Script components in the game objects. RPC function can have any number of parameters, but the number of parameter will influence on the game performance. Through the profile analyzer of Unity3D, we analyze the condition of CPU in the system. It turns out that when players move chess pieces, script class CPU is most frequently used and the usage rate of the RPC function is over50%. As shown in Figure 3.

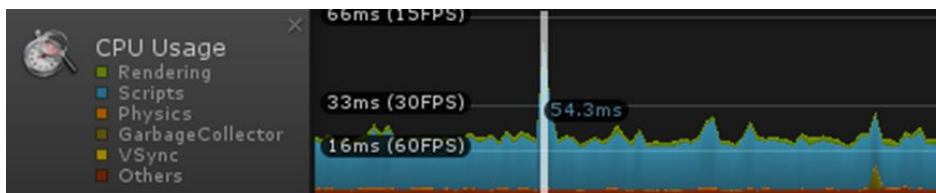


Figure 3. CPU Usage Rate in Chess Pieces Moving Process

According to test results, under the same operation, every additional pass parameter of string class RPC function adds, the usage rate of its CPU will increase 2.6% in average. As shown in Figure 4 and Figure 5.

Overview	Total	Self	Calls	GC Alloc	Time ms	Self ms
Network.Update	48.7%	0.3%	1	12.2 KB	26.82	0.19
QIPan.qipanrpc()	48.4%	45.1%	2	12.2 KB	26.63	24.82
Stack TraceUtility.PostprocessStacktrace()	1.5%	1.5%	3	3.8 KB	0.84	0.84
Stack TraceUtility.ExtractStack Trace()	1.2%	1.2%	3	6.0 KB	0.70	0.70
Instantiate	0.2%	0.2%	1	0 B	0.14	0.14
Static Collider.Create (Expensive delayed cost)	0.1%	0.1%	1	0 B	0.10	0.10
Animation.RebuildInternalState	0.0%	0.0%	1	0 B	0.01	0.01

Figure 4. RPC Function is Passed 5 String Type Parameters of CPU Usage

Overview	Total
Network.Update	51.4%
QIPan.qipanrpc()	51.0%
Stack TraceUtility.PostprocessStacktrace()	1.6%
Stack TraceUtility.ExtractStack Trace()	1.2%

Figure 5. RPC Function is Passed 6 String Type Parameters of CPU Usage

Hence it can be seen that the number of parameters of RPC function should be optimized to minimum so as to decrease network consumption and improve network performance.

2.3.2. Logic Control of Chess Play: To the Qoplayer object in the game attached the Qipan script component in which there is a RPC function qipanrpc and two entry functions operated by clients on touch screen. The first entry function is for users to choose chess pieces, move and attack point.

```

STATUS FingerTap(INT fingerIndex, VECTOR2 fingerPos, INT tapCount)
...
IF(IsConnect.single OR
    (IsConnect.doubleANDNetwork.isServer AND QPlayer=="RPlayer") OR
    (IsConnect.doubleANDNetwork.isClient AND QPlayer=="BPlayer") )
    /* If the game is a console one or online game mode in which the server plays
    the red chess pieces while the client plays the blue ones*/
    ..... /*Users are allowed to operate and process movement
logic*/
ENDIF
IF(IsConnect.double) /* If the game is in online game
mode */
    CALL networkView.RPC("accessSelector",
    RPCMode.Others,name,I,J,obj.transform.position, camera_tag) /*Invoke the
RPC */
ENDIF
...
RETURN
RETURN
    
```

IsConnect is a user-defined public class used to determine whether the game is an online or console based video game. If the game is a console based video game or online game and the server plays the red chess pieces and the client plays the blue chess pieces, users will be allowed to operate and process mobile logic to refresh the location of the chess pieces and process the animation. In addition, if it is in the online game mode, in the end it has to invoke the RPC function. Another entry function is use to cancel the operation by the user.

```

STATUS FingerLongPress(INT fingerIndex, VECTOR2 fingerPos)
IF(camera_tag== OBJECT) /*If the user long presses the
screen*/
    camera_tag<—OBJECT2
    QiMove<—QiPan.player.GetComponent<QiMove>
    CALL qiMove.Cancel() /*Deselect the
chess*/
    player<—null
ENDIF
IF(IsConnect.double)
    CALL networkView.RPC("accessSelector",
    RPCMode.Others,name,I,J,obj.transform.position, camera_tag) /*Invoke the
RPC */
ENDIF
RETURN
    
```

Camera tag is to determine the users' choice: chess piece objects (0), move or attack point (1) or no operation. If the user long presses the screen to deselect the chosen chess, the parent class scripts QiMove components of all the chess pieces' mobile script will be obtained by the user, meanwhile, the Cancel function will be invoked to deselect the objects of chess pieces and eventually the RPC function will be invoked to inform the another user of this operation.

NetworkView.RPC function is the key function to RPC, which at least requires two parameters: RPC function qipanrpc which will be invoked and already stated; receiving

object of RPC invocation. Besides, we can add any number of RPC parameters needed to be invoked, yet the parameters need to be one-to-one correspondence with parameter type and number of qipanrpc function.

After RPC receiving objects receive the parameters transmitted, we need to invoke the qipanrpc function and run the code like FigureGestures_OnFingerTap or FingerGestures_OnFingerLongPressn function. Through the RPC invocation, the data transmission and logic control have been realized, between mobile terminals in the process of chess play.

4. Examples and Result Analysis

As a touch-screen 3D game with exquisite and colorful images, if each frame of animation in the network is synchronized, a large of number of bandwidth will be consumed, which will influence game performance and user experience. Therefore only on the log-in interface, we invoke the Java code for WIFI connection and create WLAN for Unity3D network function. Once entering the game interface, Unity3D is taken to control the data transmission through RPC. After many times of attempts and performance tests, it is proved that in order to simulate the same operation at RPC receiving end, game sending end only needs to transmit 5 parameters which are the name of the game object chose by the user at the game sending end, the location of the chess piece being moved and the type of game object by the user at the game sending end. The time delay between the user's touch-screen operation and RPC invocation is short that user can barely notice. Hence in this way the information about the color of the chess pieces and the change in their location can be instantly transmitted to client or server, thus realizing the synchronization from the user's aspect.

Performance tests and actual operation, the game is proved to display steady network connection, good reaction speed and image synchronization, which testifies that the combination of Java's WIFI module and Unity3D's RPC invocation is an effective way to develop 3D game in LAN.

5. Conclusions

This paper discusses the implementation of online play of a 3D chess game on mobile phone and combined with the characteristics of Unity3D game engine and its shortcomings in network function, the author proposes to integrate Java's Activity class, invoke Java in Unity3D to combine Java's WIFI module and Unity3D's network module, thus enabling two mobile terminals to play multiplayer online game in LAN with the aid of RPC invocation, which successfully leads to the development of Unity3D online 3D chess game based on Android platform. Henceforth, more plans on Unity3D network connection and online games based on Unity3D will be explored. The design sketch of online play effect is shown in Figure 6.



Figure 6. Connection Established, Game Starts

Acknowledgement

This research was supported by The Fundamental Research Funds for the Central Universities (RW2014-16). We also thank anonymous referees and the editor-in-chief.

References

- [1] R. Guo-Dong, C. Lin-Hua, T. Xue-Feng and F. Xian-Xing, "Virtual Museum Information Visualization System Based on Unity3D", *Computer Systems & Applications*, vol. 22, no. 59, (2013), pp. 86-90.
- [2] C. Jiunn-Ching, H. Cheng-Ting, C. Ming-Hsien and J. Yu-Jie, "Unity3D game-based learning system", *Proceedings of the 2nd International Conference on Innovation, Communication and Engineering*, (2014), pp. 235-238.
- [3] C. Ning and G. SuBin, "The development of real navigation view based on unity3D", *Advanced Materials Research*, vol. 846-847, (2014), pp. 1813-1817.
- [4] X. Jing-Ming, "Research on key technologies base Unity3D game engine", *ICCSE 2012 - Proceedings of 2012 7th International Conference on Computer Science and Education*, (2012), pp. 695-699.
- [5] Q. Gang, X. Zongze and Z. Lijuan, "Hop Selection Algorithm in Bluetooth and Its Performance Analysis", *Journal of Nanjing University of Aeronautics & Astronautics*, vol. 33, no. 6, (2001), pp. 574-578.
- [6] S. Xu-Dong, "Study and Implementation of NFC technology based on Android mobile OS, " software, vol. 34, no. 1, (2013), pp. 64-68.
- [7] P. Wen-Lian and Z. Lin, "Application of WiFi Technology in Staff Positioning System on Android Platform", *Jisuanji Yu Xiandaihua*, vol. 210, no. 2, (2013), pp. 159-162.
- [8] A. Mossel, C. Schönauer, G. Gerstweiler and H. Kaufmann, "ARTiFiCe - Augmented Reality Framework for Distributed Collaboration", *The International Journal of Virtual Reality*, vol. 11, no. 11, (2012), pp. 1-7.
- [9] Z. Hui-Juan, "Virtual Roaming System Based on Unity3D", *Computer Systems & Applications*, vol. 21, no. 1, (2012), pp. 36-40.

Authors

Shangguan Dayan, He received M.A. degree from Peking University, Beijing, China, in 2009. He is currently a Ph.D. student in Beijing Forestry University. He is the lecturer in college of Arts and Design at Beijing Forestry University. He current research interests include the mobile Internet and digital art.

Huang Xinyuan, He received B.S. degree and a Ph.D. from Beijing Institute of Technology, Beijing, China, in 1986 and 1991. He was the dean, a professor and doctoral supervisor in school of information at Beijing Forestry University. He is currently the dean, a professor and doctoral supervisor in school of Animation and Design Arts at Communication University of China, Beijing, China. He is the vice president and organization minister of the Association of Fundamental Computing Education in Chinese Universities, and a member of animation and digital entertainment committee of image and graphics computer institute of China. His research interests are computer animation and digital forestry.

