AR Smart House: Using Customized Interactive 3d Models to Control IoT Devices

Huang Jingwen
44191527-4

Master of Engineering

Supervisor: Prof. Jiro Tanaka
Graduate School of Information, Production and Systems
Waseda University

September 2021

Abstract

In recent years, with the development of technology and the improvement of people's requirements for life. More and more families are turning their houses into smart houses. Smart speakers to manipulate smart homes has matured day by day, but there are still many scenes such as noisy outdoors, libraries that need to be kept quiet is difficult to use the smart speaker. Also, when people use a mobile app, it is difficult for everyone to have an intuitive understanding using simple numbers and text prompts. And these systems have only a few elements to make users interested. When users are away from home, it is difficult to remember that they need to operate in advance.

In this paper, we introduce AR technology to provide a new experience for smart house to provide more intuitive feedback and easy-to-understand operations. We designed and implemented an AR smart house system that allows user use customized AR 3d model to control IoT devices which are used in their house to provide a more intuitive feedback and a more interesting interaction. This system allows users to reconstruct a 3d model using photos from multiple angles of the object that they prefer. Multiple models can be created, and can be connected to the corresponding IoT devices. Then users can put these models into the AR environment via mobile phone, and control the IoT device by interacting with model, and the model can reflect the state of environment and the device through unique actions.

Keywords: smart house, augmented reality, 3d model, IoT

Acknowledgements

At the beginning, I am very happy to express my heartfelt thanks and gratitude to my supervisor, Pro. Jiro Tanaka, who is a highly respected and responsible mentor with rigorous academic attitude and innovative ideas. During the two years of studying in Waseda, he gave me a lot of academic help and guidance. He led me to study the field of augmented reality and human-computer interaction. During every stage of preparation of this thesis, he spared no effort to give me a lot of valuable suggestions.

Besides, it is grateful to all IPLAB members. They discussed a lot with me and always give me lots of inspiration. They not only provided me with a lot of help in research, but also gave me a lot of companionship and help in life in Japan. In the case of the COVID-19 epidemic, we had a difficult time. Thanks for your companionship and support.

Finally, I want to give my gratitude to my mother and all family. You give me financial support and warm mental support throughout my master study life.

Contents

List of figures

List of tables

Chapter 1

Introduction

1.1 Introduction

Augmented Reality(AR) is a technology which aims to create the illusion that virtualimages are seamlessly blended with the real world.[1] When things may include any objects, home appliances, devices, vehicles, etc. connect to the internet in specific infrastructure via standard protocols then the whole system is said to be Internet of Things (IoT). [2] AR smart house is a system that uses AR and IoT technology to help users control smart appliances in AR scene.

Now, people usually use two ways to control smart house devices, one is voice assistants like Google assistant.[3] The second is the buttons on the mobile app, like Philips Hue app.[4]

But they still have some limitations as following.

- The voice interaction using voice assistant requires a quiet environment, which is not convenient for operation in a crowd. And, because of the need to use dialogue interaction, it usually takes a long time.
- Both voice assistant and mobile app use numerical values to give feedback of the situation at user‘s house. But for some people such as children or people who are not sensitive to numbers, it is difficult to understand the actual temperature and so on from these values.

- The system does not consider the promotion of users, and has less element that can make users interested. When users are away from home, it is difficult to remember that they need to operate in advance.

Our system hopes to solve these limitations and provide a more relaxing experience with richer interactive in smart house control scenes. Therefore, our system allows users to interact with the smart house devices using customized interactive 3D models.

1.2 Organization of the thesis

The rest of the thesis is organized as follows: chapter 2 will introduce some background and basic knowledge related to this work. Chapter 3 will give the overall goal of this research and a brief introduction to the research method. Chapter 4 will introduce the design of our system in detail, including the overall design of the system and the detail design of each part. Chapter 5 introduces our system implementation, including the hardware and software used and the detail implementation methods of the functions. Chapter 6 introduces some related work of this research. Chapter 7 briefly summarizes this work and talks about some possible future work.

Chapter 2

Background

2.1 Augmented Reality

2.1.1 Augmented Reality

The definition of augmented reality proposed by Ronald T. Azuma‘s if a general definition. He defines AR that has the following 3 characteristics: [5]

1. Combines real and virtual
2. Is interactive in real time
3. Is registered in three dimensions

Another definition is proposed by Paul Milgram and Fumio Kishino. They regard the real environment and the virtual environment as the two ends of a continuous system, and the one between them is called "mixed reality".[6]

2.1.2 Using 3D model in AR

Literally, markerless augmented reality means the AR technology which does not need the marker. Markerless AR refers to a software application that doesn’t require prior knowledge of a user’s environment to overlay virtual 3D content into a scene and hold it to a fixed point in space.[7] Markerless AR recognizes feature points in the environment instead of specific patterns prepared in advance, which allows users to freely use AR anywhere.

Currently, AR devices and interaction methods usually have following types, using head-mounted AR devices and gesture interaction, mobile devices and gesture interaction, and mobile devices and touch screen interaction. Hürst et al. [8] concluded that touchscreen operations outperformed freehand movements in creating and editing 3D models in AR applications on mobile devices.

Using cute animals and plants, or other models in the AR environment is a common way to enhance the interest of the AR system. And research shows that using cute models in the system can have a positive effect on changing user behavior.[9] And, some investigations in research show that using Animated characters with the same characteristics as home appliances would be useful to represent home appliances.[10]

2.2 Smart House System

A Smart House is a house or living environment that contains the technology to allow home devices and systems to be controlled automatically. [11] Smart houses user can easily monitor and control all home devices/home appliances through internet.[2] In order to realize such a system, the indispensable technology is the Internet of Things(IoT). With the development of technology, smart houses have begun to be widely used in families

2.2.1 Smart Speaker

Typically, smart speakers include Wi-Fi and Bluetooth connectivity. Simpler products range from Wi-Fi-enabled boom boxes to adjustable RGB LED smart speakers that can be spread through a home for distributed or localized stereo sound.[12] The number of smart speakers used in homes is increasing, and integrated multiple functions including smart home control. Users only need to speak their needs to the smart speaker in natural language, such as "please turn on the air conditioner ", and then they can operate the appliances connected to the network in their house.

Smart speaker provides users with a natural interaction, but at the same time we noticed that it has many problems. First, because of the voice interaction, smart speaker requires the quiet environment with less noise. Also, many people will not use their own smart speaker when others are present. Secondly, smart speaker needs to use time to interact, and the feedback they get is not intuitive. If the smart speaker fails to understand the user's meaning, it may require more communication to achieve what the user wants to do. Moreover, if the smart speaker executes the wrong instruction, it will take a long time to correct it.

2.2.2 AR smart house system

Interactive visualization is considered as an important issue of the smart house.[13] witch usual smart house system lacks. So, the solution considered by some studies is to combine it with augmented reality. Technically, smart house needs to use IoT technology. It is a combination of physical objects with virtual representations and services. And Augmented reality provides an ideal interface to IoT applications by superimposing virtual information about smart objects and services on a user's view of the real world.[14] AR and IoT two seemingly unrelated concepts, they might have different objectives, but can be complementary to each other along with the potential advantages and expected synergies of integrating them.[15]

Chapter 3

Related Work

In this section, I will introduce the related works of this research. First, I will briefly introduce the current smart house system, and then I will introduce the AR smart house system in more detail. At last, I will introduce some research related to 3D models and non-human avatars in the AR environment.

3.1 Smart House

Nowadays, we have many kinds of appliances in our homes, some of which are very suitable for being incorporated into the smart house system. Some early research proposed by Laehyun Kim, Wanjoo Park, Hyunchul Cho, and Sehyung Park[16] introduced a new universal remote controller that gives easy-to-control interface for home devices such as TV, video/audio player, room lighting and temperature control. Their focus is on the design of interaction between humans and the remote control.

Dongyu Wang, Kazunori Sugiura, and Yui Murase[17] is developing a User-centered universal controlling system that will provide a new experience of appliances controlling. They found that by connecting with sensor technology, and other web services, home appliances controlling could become more user-friendly and enjoyable.

Abdulla AlHammadi et al. [18] overview how the Internet of Things (IoT) is integrated with Artificial Intelligence to abstract the idea of remotely connecting and monitoring physical objects within smart houses. They said that smart homes turned out to be the new developing wave of technical growth.

The current smart house system has a variety of man-machine interfaces, among which the more common ones include the voice control, the wall-mounted touchscreen, or the mobile application. Michal Luria, Guy Hoffman and Oren Zuckerman[19] presented a novel design for a home control interface in the form of a social robot and they experimentally compared the robot to these common smart-home interfaces. And explored the advantages and disadvantages of these methods.

Many people have proposed automated smarthouse systems, but Intille, Stephen S[20] thought that technology should empower users by allowing them to make decisions for themselves, rather than having the home make decisions for them.

Erik Kučera, Oto Haffner and Štefan Kozák [21] describes an interactive 3D application that simulates virtual tour of the smart home and its exterior. And in its future research, they mentioned that it would be interesting to use this experience and develop the application for mixed/augmented reality that will communicate with real sensors and actuators.

3.2 AR Smart House

How to allow users to interact with the system in smart house is a topic, and AR is an excellent Human interface, which has led to many AR smart house related researches in recent years. Rafael Giménez and Marc Pous [22] noted that "AR has recently been touted as one of the ideal interfaces for IoT."

There is an exist application SmartARHome[23] that connects the world of smart home automations with the augmented reality. This is a marker-based system and integrates Samsung SmartThings and Philips Hue devices.

Rina Umeyama and Hidekazu Suzuki[24] proposes a smart appliance controller named "iHAC" allows users control smart appliances intuitively by using not only character information but also a video image displayed by augmented reality technology. he work absorbs the differences in communication protocols.

Dongsik Jo and Gerard Jounghyun Kim[25] did a survey about the current state of AR enabled IoT and discuss research on issues about realizing a future smart and interactive living environment. This article proposes that there are some issues including data management, display device and interfaces and interaction methods in AR and IoT. Also, this paper mentioned that AR-based object control and interface may be a key component of the prospective feature from combining AR and IoT.

Ching-Hu Lu[26] propose an interactive eco-feedback with three kinds of information visualization integrated with a 3D pet-raising game, which synchronously visualizes the information of the physical environment with the virtual environment by leveraging IoT enabled technologies in hopes of enhancing user experience and prolonging users' engagement in energy savings

In Maribeth Gandy and Blair MacIntyre[27] collected feedback from some new media designers who use augmented reality design tools, it shows that mixed reality has been incorporated to enrich the playfulness of a system, particularly for non-technologists.

Shinya Mihara, Kohei Kawai, Hideki Shimada, and Kenya Sato[28] proposed embodied Visualization with Augmented Reality for Networked System 3 (EVANS 3) to solve operation problems with obvious and intuitive controls by implementing augmented reality (AR) technology. They implemented our proposed LED Marker instead of the use of graphical images (image markers) as the AR markers to identify the network home appliance's location at any distant and altering light environments.

Sora Inomata, Kosuke Komiya, Koya Iwase, and Tatsuo Nakajima[10] proposed AR Smart Home, which uses the augmented reality technology and gesture recognition technology. They conducted an evaluation and found that operating home appliances with gestures and interacting with virtual 3D home appliances instead of the actual home appliances are acceptable in terms of usability. And also evaluated the availability of animated characters with the same characteristics as home appliances, and other objects with characteristics similar to those of home appliances, etc. when operating with other 3d virtual objects.

Atieh Mahroo et al. introduced the HoloHome[29] , an Augmented Reality framework which aims to provide new means of interaction with the Smart Home and its components.

The main purpose of the HoloHome is to provide a Mixed Reality environment implemented on the Microsoft HoloLens, to allow the user interaction with the Smart Home devices and appliances through the Augmented objects.

Smart city is also another development direction of the combination of AR and IoT technology, but it is concerned about the application in larger urban scenes.Keonhee Cho, Hyeonwoo Jang, Lee Won Park, SeungHwan Kim and Sehyun Park[30] proposed a new interface EMS that can be easily accessed and utilized by consumers. They designed an energy management system that can connect people and computers through IoT and AR technology.

3.2.1 AR 3D Model

Pokémon Go is a popular mobile game published by Niantic around the world [31]. They use a well-known IP in Japan to provide a game for catching pokemon in an AR environment. Among them, pokemon, which has a cute 3D model, is superimposed in the real environment by AR technology, and is loved by countless people in the world.

Isaac Wang, Jesse Smith and Jaime Ruiz[32] conducted a study to examine users' perceptions and behaviors when interacting with virtual agents in AR. And they showed how the presence of a non-human embodiment affected how users felt about it.

Nahal Norouzi[33] presented one prior experiment which was designed to provide a better understanding of the requirements of virtual animals in augmented reality as companions. She mentioned the capability of virtual animals such as a virtual dog as an interface aimed at conveying the digital information to the user and its influence on user's decision making.

Telmo Zarraonandia, Alvaro Montero, Paloma Diaz, and Ignacio Aedo[34] proposed a system working on the implementation of an AR game for encouraging people learning about vegetation in their local environment. They identify the photos of the plants they have taken to obtain "seeds" from the database. Afterwards, these seeds can be cultivated and the cultivated plants can be planted at home.

Lele Feng, Xubo Yang, and Shuangjiu Xiao[35] present MagicToon, an interactive modeling system with mobile AR that allows children to build 3D cartoon scenes creatively from their own 2D cartoon drawings on paper. In this system, the authors found that a simple AR system with cartoon can more stimulate children's creativity.

Chapter 4

Research Goal and Approach

4.1 Goal

The goal of our research is to provide a smart house system with a rich AR interactive experience that focus on using personalized 3d models to attract users' attention which provide intuitive feedback and control. The goal can be divided into the following points:

1. The system allows users to create their own models by uploading photos of real objects.
2. The system can report the current state of the room to the user through the displayed model and the state of the model.
3. The system allows the user to control the smart house device by controlling the AR model

4.2 Approach

To achieve the goal, we will introduce two main aspects of our approach: 3D model control and IoT device control.

First of all, we need to get the 3D model that can be used in the AR environment. As shown in the Figure 4.1, in our prototype system, the model needs to go through three processes: construction, rigging, and control.

In the creation part, we need to reconstruct the 3D model of actual object from multiple photos taken by mobile phone, and then bind the bones for them to make the model movable, and finally make them controllable by the users.

Fig. 4.1 3D model processing

And, in order to put the model into the AR environment, we import ARCore to recognize the environment.

Some of the appliances we use are traditional appliances that only support infrared remote control, so we first need to connect them to the Internet. Then connect them to the model created before.

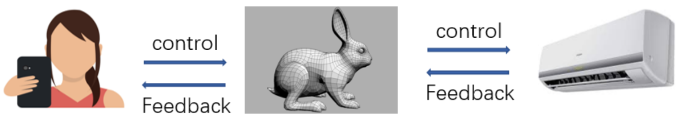


Fig. 4.2 The relationship between users, models and IoT devices

We will implement the IoT device control part according to the structure shown in the figure 4.2. The user can control the IoT device through the control model, and the IoT device can also give feedback to the user through the model.

Chapter 5

System Design

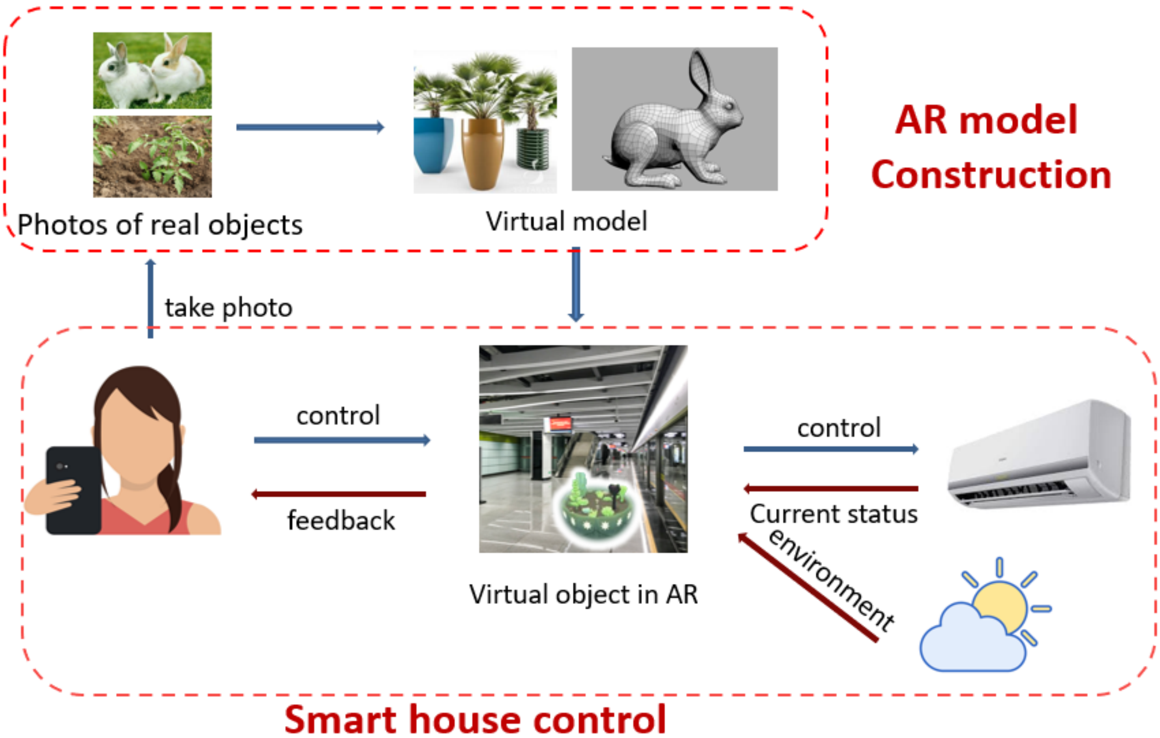


Fig. 5.1 System overview

Figure 5.1 shows the overall design of this system. This system uses 3D models created by the user which is based on real objects to control the IoT device. It can be roughly divided into two main parts.: AR model construction part and smart house control part.

First, users photograph objects in their daily lives. Based on these photos, the model will be constructed. Then these models will be respectively connected to the corresponding IoT devices and the users will place these models on their side through the AR device. The user can control the IoT device through the interaction with the model, and the model can reflect the state of environment and the device through unique actions.

5.1 3D Model Construction

In this system, 3D models can be built based on user photos of objects with multiple degrees. This function is provided by the existing software Reality Capture Photo. Taking photos from a more complete angle as much as possible and choosing objects with more feature points will help build a more realistic model.

The squirrel model shown in Figure 5.2 is based on 53 photos from different angles. The texture and color changes on this squirrel toy provide a wealth of feature points.

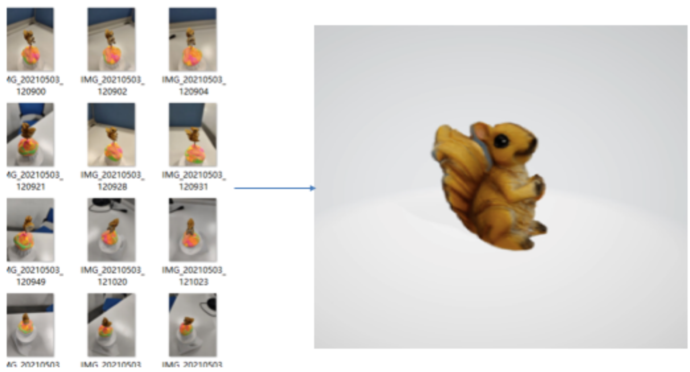
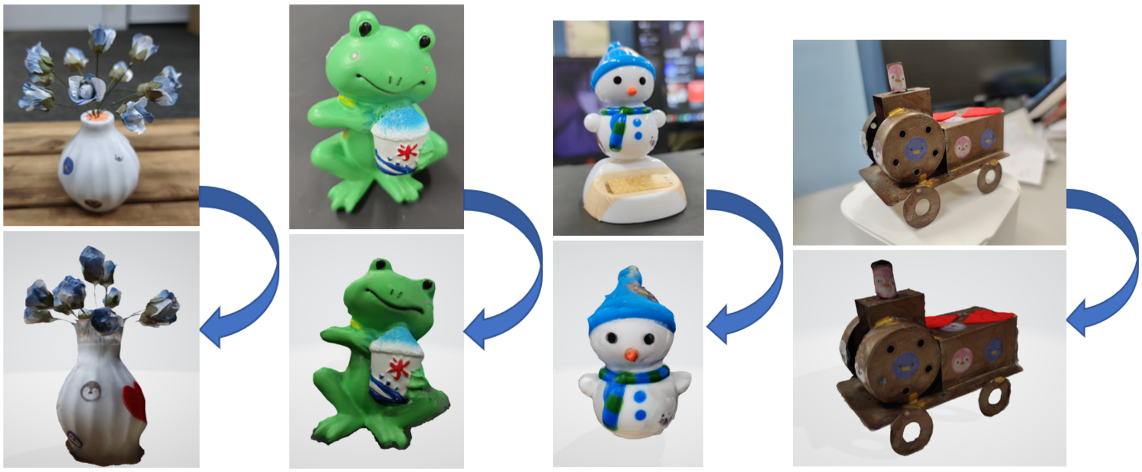


Fig. 5.2 a squirrel model construction based on photos

Using the same method, we created multiple 3D models as Figure 5.3 shows, such as vase, frog, snowman, train, etc. In the prototype system mentioned in this article, we use these models as examples of system design, providing reasonable and intuitive interactive operations. In actual use, users can upload their own models to use.



(a) Flower vase model (b) Frog model (c) Snowman model (d) Train model

Fig. 5.3 3D models used in this system

5.2 Smart House Control

In this system, we not only want to use devices that can be connected to the Internet, but also want to control some traditional home appliances. Therefore, we use Nature Remo to connect traditional electrical appliances to the Internet and turn them into IoT devices.

Users can control the appliances connected to the system in their homes through the AR model. In this study, we take the use of traditional infrared control appliances air conditioners and lights, and the sweeping robot which controlled by mobile phones as examples.

In addition, the system will combine the current time, temperature, brightness and other conditions to provide users with control suggestions.

This section will introduce the functions that can be controlled by this system in detail.

Air conditioning is a common electrical appliance that is used throughout the year. In this system, we convert traditional air conditioners operated by infrared remote controllers into IoT devices. The air conditioner can realize the functions of switch, mode conversion, and temperature adjustment.

1. Switch: The air conditioner can be turned on or off.

2. Mode conversion: Mode can be adjusted to cooling or heating mode.

3. Temperature: Temperature can be adjusted in the range of 18 to 30 in units of 1 degree Celsius.

Lights are another essential appliance in daily life. Lights are another essential household appliance in daily life. In this system, we convert traditional lights operated by infrared remote control into IoT devices. The lights we use can be switched on and off to adjust the brightness, and it also has a night light mode.

1. Switch: The light can be turned on or off.

2. Brightness: Can be adjusted between 10 different brightness.

3. Night mode: Some people turn on the night mode when they sleep, and this is also the lowest brightness mode.

Sweeping robots are becoming more and more common in our lives. Today's sweeping robots usually use mobile apps to control them via the Internet. We also integrated it into this system. The user can control the sweeping robot to start, stop, or return to charging.

1. Start: Use automatic mode to start sweeping.

2. Stop: Stop immediately.

3. Return to charging : Return to charging point and stop

5.3 Connection of Smart Devices and 3D model

The system links the 3D model with the corresponding appliances, and allowing users to realize the function of controlling smart appliances through the model.

In our prototype system, we have implemented several common simple scenarios, and we use the model and appliances shown in the Figure 5.4 to complete out prototype system. Among them, we use three different models of squirrel, frog and snowman to control and express the state of the air conditioner. The squirrel represents the on or off of the air conditioner, and when the user operates, it can give feedback to the user. The frog who eats ice and snowman represent the current temperature of the room. Eating shaved ice is reminiscent of summer, so the frog represents the temperature of the room. The snowman represents winter so the room temperature is low, which is in line with people's intuition. And we associate the vase with the lamp, the flowers and vases are independent, and there are also models of the sun and the moon. Finally, we use a locomotive model to connect with the sweeping robot, and let the running locomotive represent it.

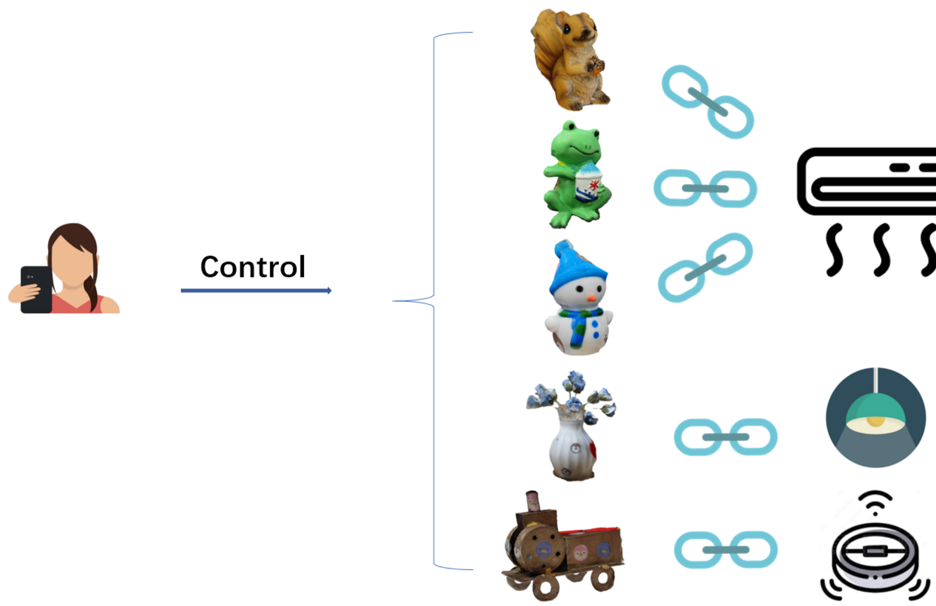


Fig. 5.4 Relationship between model and appliance in the prototype system

5.3.1 Air Conditioning Control

In the prototype system, three models were used to control and give feedback of the air conditioner and current temperature. The frog and snowman (Figure 5.3b, Figure 5.3c) respectively indicate that the current temperature is high or low. And the squirrel model (Figure 5.2) is used to indicate whether the air conditioner is on or off. We use squirrel's sleeping or waking to represent different states of air conditioner. The squirrel contains three main actions as Figure 5.5 shows. They are awake state, eating state and sleeping state. Where eating is a transitional one-time action that give user the feedback of changing temperature of the air conditioner, awake and sleeping are a continuous state action that indicate the air conditioner is on or off . In addition, we also have a transitional action, which is a little jump. It will be used in the transition of sleeping and waking state to make the squirrel's movements more natural.

(a) Normal(Awake) (b) Eating (c) Sleeping

Fig. 5.5 3 status of squirrel

The relationship between the 3 states and the air conditioner is shown in the following flowchart 5.6. First, the system judges the current temperature. If the temperature is high, a frog model eating shaved ice is displayed, and if the temperature is low, a snowman model is displayed. After that, the switch of the air conditioner is judged. If the air conditioner is on, a squirrel model that is awake is displayed, and if the air conditioner is off, a model of a squirrel that is asleep is displayed. The user can switch the air conditioner switch by touching the squirrel. After that, if the air conditioner is turned on, the user can control the temperature of the air conditioner by feeding different foods.

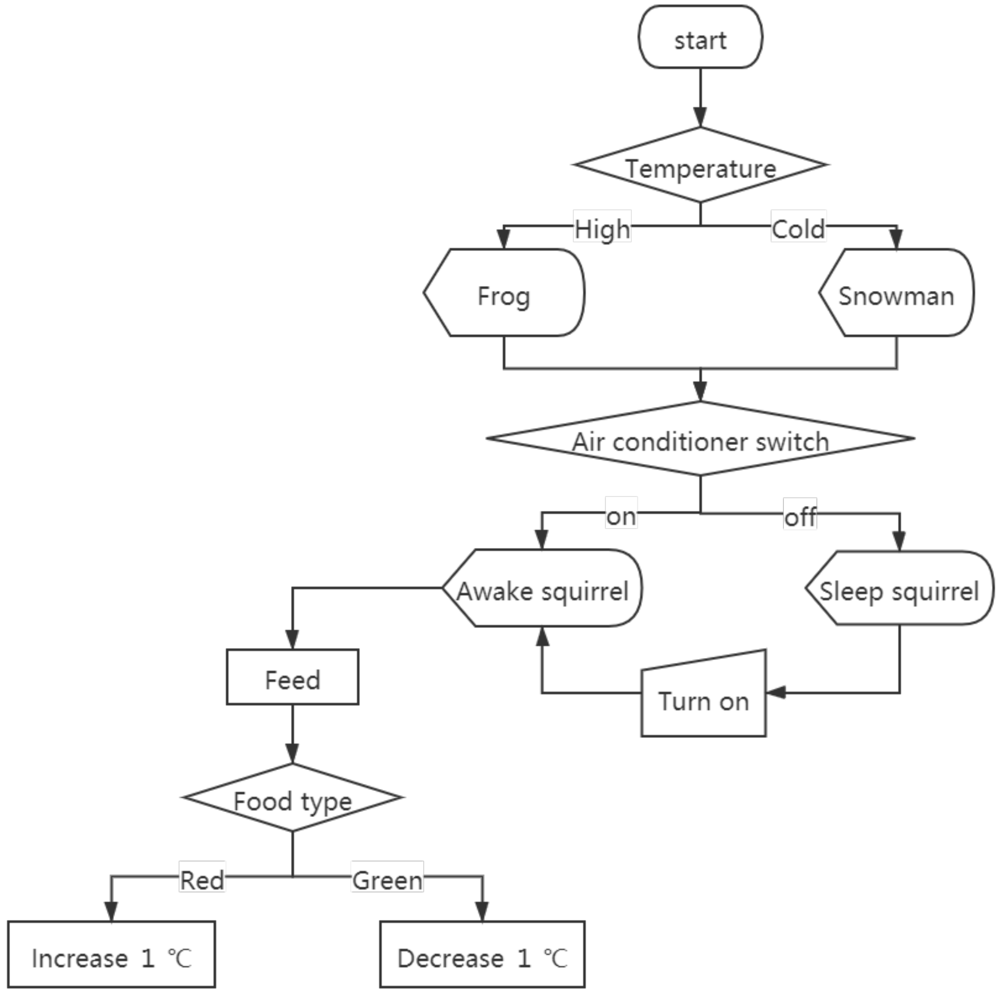


Fig. 5.6 Air Conditioning Control flow chart

5.3.2 Light Control

In this system, in order to control the light, we use a model of a vase (Figure 5.3a). As Figure 5.7 shows, the number of flowers represents the intensity of the light, and at the same time, the lighting of the vase in the AR environment will also change with the actual illumination situation in the user's home. And, depending on the lamp mode, the sun or the moon will be displayed next to the vase.

The user can control the height of the sun by dragging the sun model to control the brightness of the light. When the user drags the sun down, the number of flowers will be reduced shows the brightness of the light is reduced, and the light of the vase will also be weaker because the lighting from the lamp is reduced. On the contrary, when the user drags the sun upwards, the brightness of the light will increase, the flowers will increase, and the vase light will become stronger.

Fig. 5.7 Vases models in different states

5.3.3 Sweeping Robot Control

In our prototype, we associate the locomotive (Figure 5.3d) with the sweeping robot. The locomotive model has three main states as shown in Figure 5.8, which are inactive, activated and flashing states. They respectively represent that the sweeping robot is off, on and charging.

When the sweeping robot is off, the model is gray. When the sweeping robot starts to sweep the floor, the locomotive will be activated, which appears to be running around and emitting steam. When the sweeping robot is charging, the model will shine.

When the user drags an inactive train back and forth, he will be activated and turn on the sweeping robot. After that, if the user wants to stop it, he can stop the running train model.

(a) Inactive (b) Active (c) Charging

Fig. 5.8 3 status of locomotive

5.4 User Control

5.4.1 Detect Plane

Whether the user wants to view or control any appliance, the first thing that system needs to do is to detect the plane.

As shown in the Figure 5.9, the system first detects the plane and marks the position of the recognized plane with a grid. Then, the user clicks on the plane to instantiate the corresponding model at any position on the recognized plane.

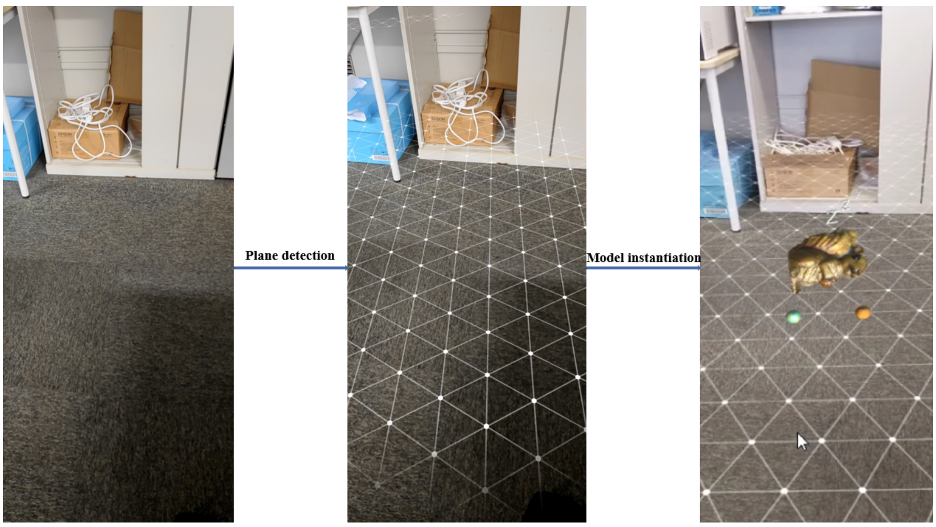


Fig. 5.9 Plane detection and model instantiation

5.4.2 Gesture Control

The system allows the user to control the model through the mobile phone. The operations supported by our prototype system include zooming in and out, moving objects in the detected plane, and moving objects in the vertical direction.

Among them, in the air-conditioning scene, the user can drag food or squirrels to feed the squirrels or let the squirrels search for food. As Figure 5.10, After feeding the squirrel, the squirrel will make the action of eating.

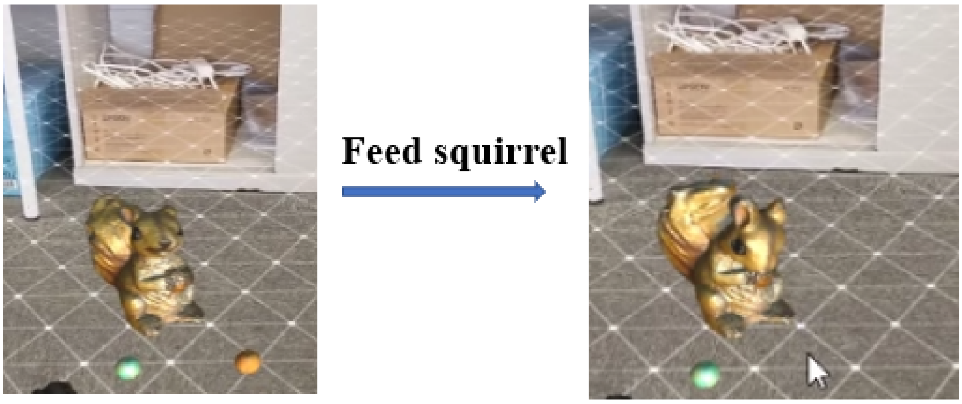


Fig. 5.10 Feed squirrel

In the light scene, the user can drag the sun in the vertical direction to change the height of the sun like Figure 5.11 shows. The height of the sun has a certain limit, and the user can move the sun up and down freely within this limit. The height of the sun directly affects the model's illumination and the number of flowers.

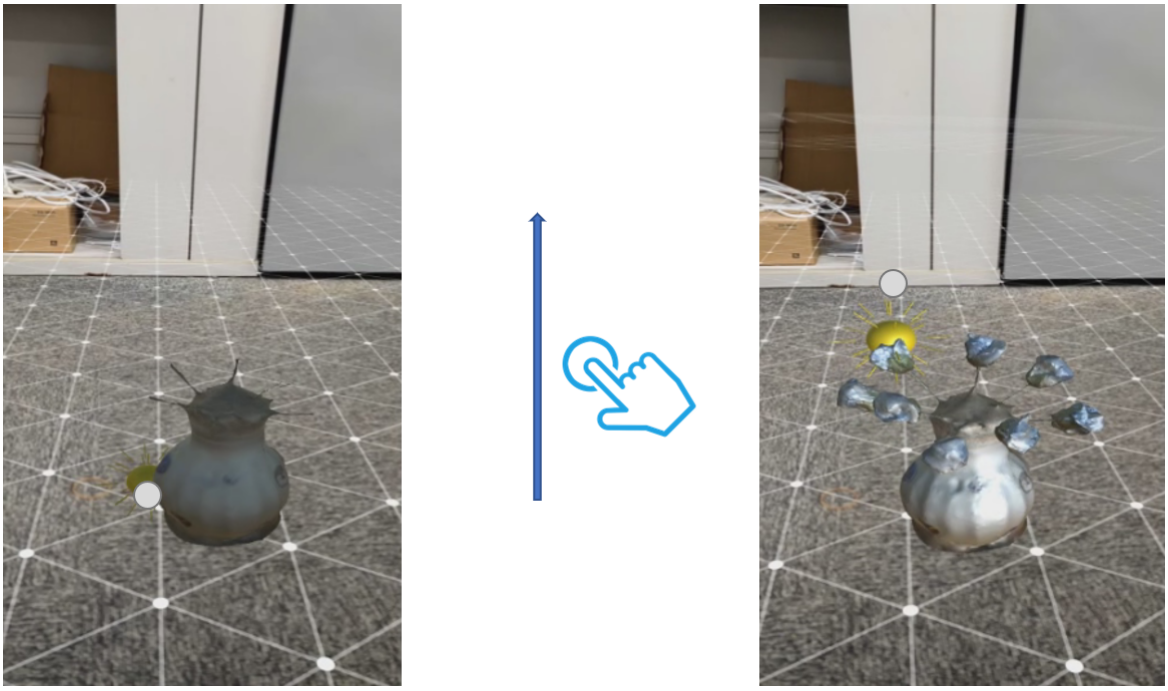


Fig. 5.11 Move the sun up and down

In the scene of the sweeping robot, user can drag the locomotive to move around, and the moving range reaches a certain level, the system will recognize it, and it will be recognized as a sign to turn on the sweeping robot. As Figure 5.12 shows, after the locomotive is activated, the color becomes richer, and it emits steam to make a sound.

Fig. 5.12 Drag the locomotive

5.5 Use Cases

3 related scenarios are included in our prototype system. The first is related to air conditioning and temperature, the second is related to lighting and lights, and the third is related to cleaning and sweeping robots. At the beginning of the system, the user can select the system he wants to view and control. The entrance scene is shown in Figure 5.13.

There are 3 buttons in the upper right corner of this interface, which prompt 3 different scenarios. The text in the middle of the screen prompts the user to select one of the scenes. The style of the button will also give user some information. The temperature button in the picture shows that the temperature is out of the comfortable range. The illumination button is dark gray, and the color of the button directly indicates the current brightness. The ordinary white button means that no operation is required.

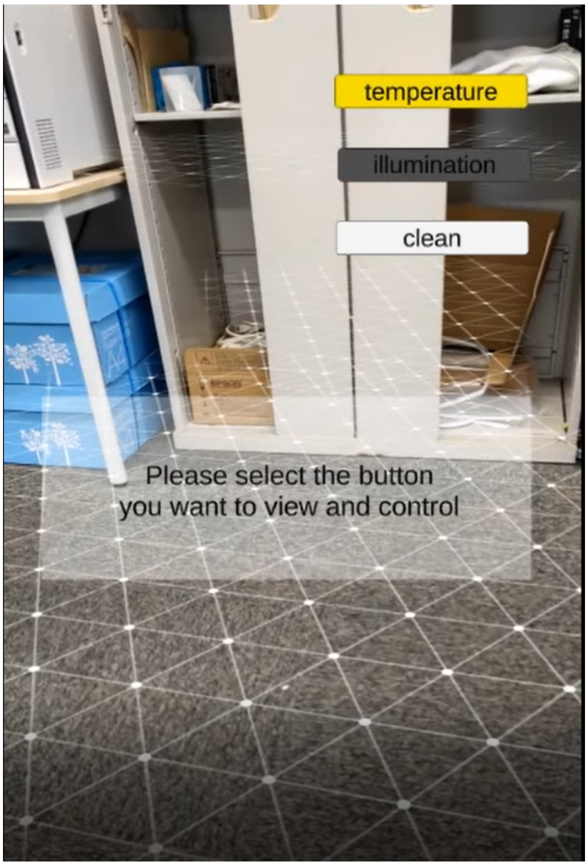


Fig. 5.13 Entrance scene

5.5.1 Use Case 1

A user was outside and preparing to go home. It was very cold. Before leaving in the morning, user opened the windows for ventilation. User thinks that it must be cold in the house now. He got on the tram and decided to check the temperature at home.

As Figure 5.14 shows, after user selects the temperature button, a new prompt appears, ask user to wait for plane detection and click on the desired position on the detected plane to instantiate the model. It is winter, and the system detects that the home is only 10 degrees Celsius. So, the system shows a snowman in the AR environment, and ask user through the text whether user want to check the condition of the air conditioner.

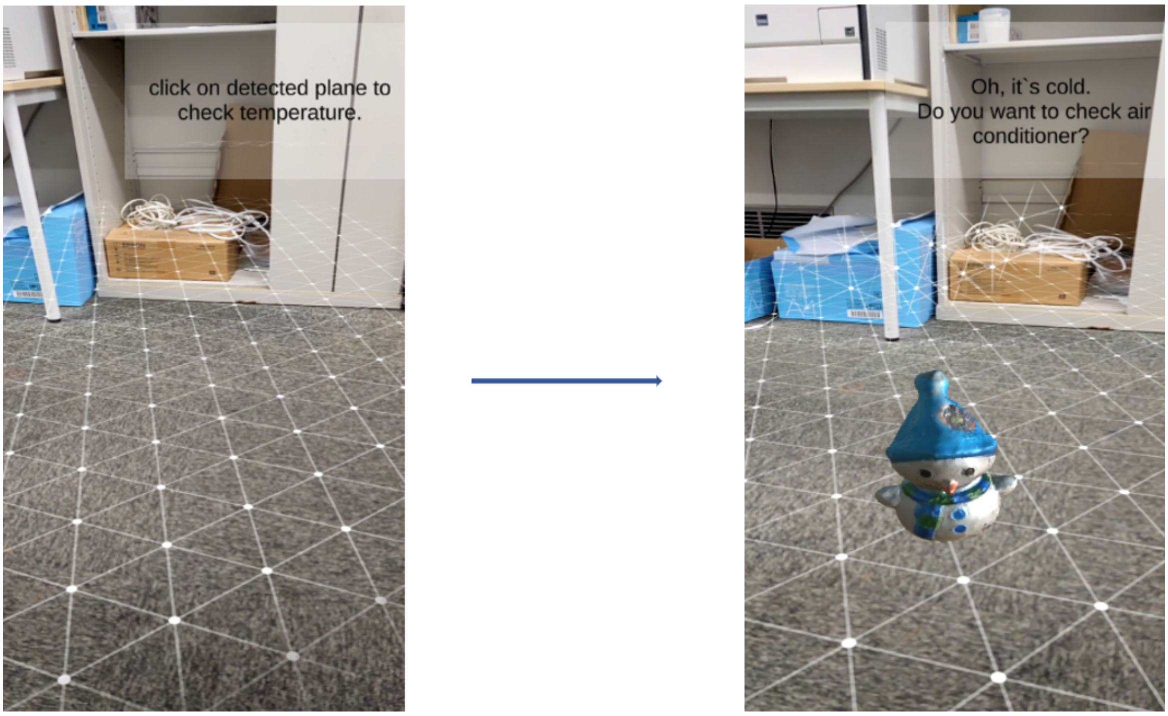


Fig. 5.14 The model representing the temperature

The user decided to turn on the air conditioner, so that he or she clicked on the detected plane again to call a squirrel model representing the air conditioner. At this time, the squirrel was sleeping, indicating that the air conditioner was turned off. So, the user touched the squirrel and turned on the air conditioner. The system recommends that the user increase the air conditioner, so red apples are displayed as Figure 5.15 shows. Then, the user feeds the apple to the squirrel and raises the temperature of the air conditioner.

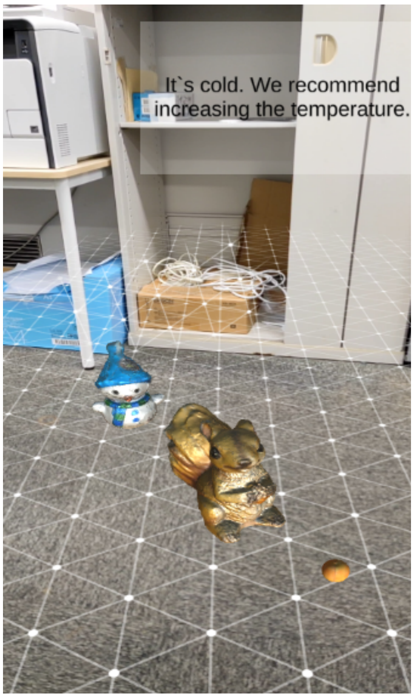


Fig. 5.15 The models indicating the air conditioner status

The structure of this scene is shown in the Figure 5.16. The user learns the temperature of the room from the model and controls the air conditioner in the room through another model.

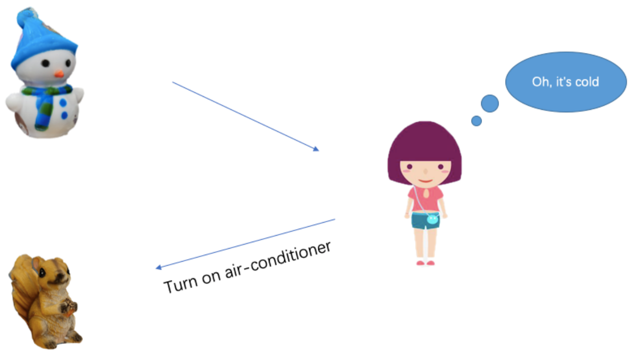


Fig. 5.16 air conditioning and temperature scene

5.5.2 Use Case 2

A user encountered unexpected overtime this day. He or she lives alone, and his worried about encountering a thief because of his late return. So, he decided to turn on the headlights at home. After user selects the lighting in the entrance scene, similar to the first scene, a new prompt appears, ask user to wait for plane detection and click on the desired position on the detected plane to instantiate the model.

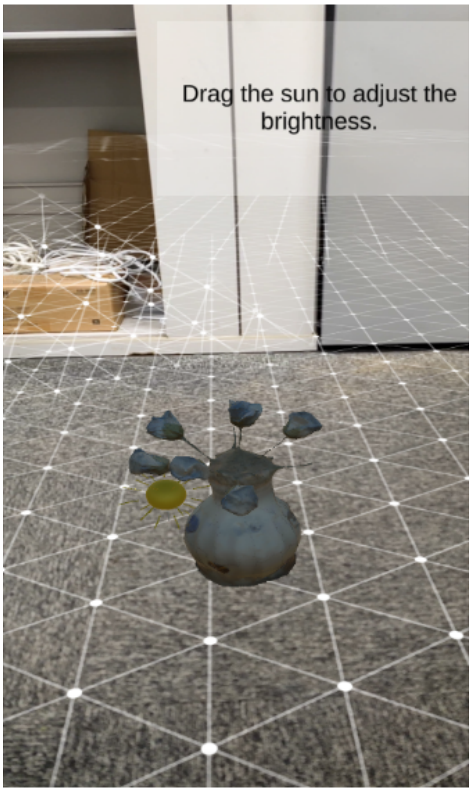


Fig. 5.17 The model that represent light and illumination

Use called out the vase model connected to the light. The light in the house was not turned on at this time, so there was only one flower in the vase, and the vase was dark because there is no light. He turned on the light, a model of the sun rose next to the vase, and the vase was filled with flowers. He thought that there was no need for bright lights when there was no one in the house, so he pulled the sun down and dimmed the lights. The vase becomes darker and there are fewer flowers in the vase as Figure 5.17 shows.

5.5.3 Use Case 3

A user used a sweeping robot to clean the house three days ago. When he clicks the "clean" button, the system will prompt him first to click the detected plane to check clean situation. He followed the prompt and clicked on the ground to instantiate the locomotive model. Then, the system showed a grey locomotive and asked him if he wanted to start sweeping. As Figure 5.18 shows, the user thinks that it should be cleaned, so he drags the locomotive in the detected plane, and the locomotive starts to work with steam. After a while, the sweeping was over and the locomotive stood still.

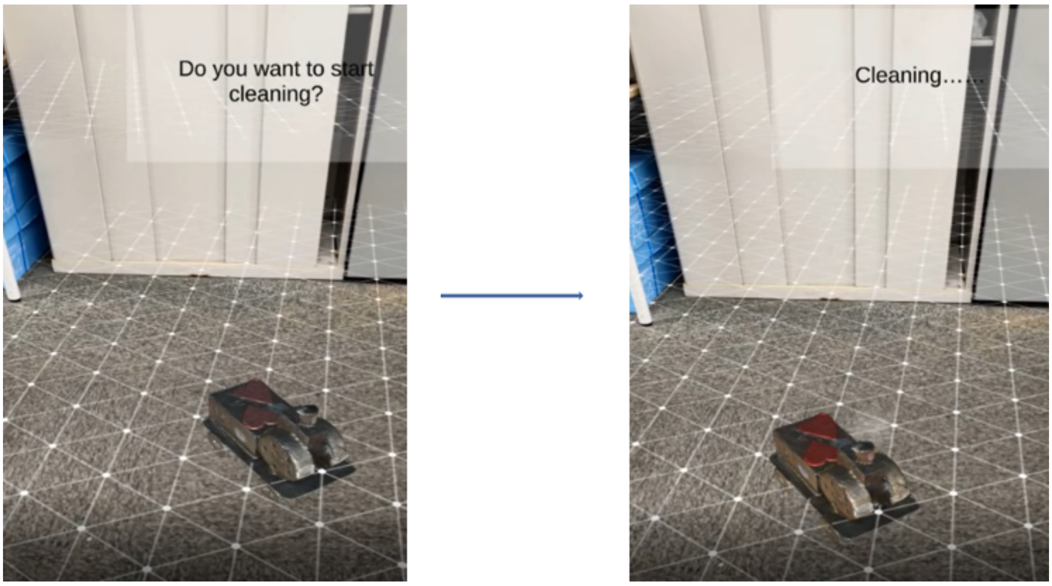


Fig. 5.18 The model that represent sweeping robot

Chapter 6

System Implementation

6.1 System Hardware

We use the Android smart phone XIAOMi 11 as the platform on which the program runs. Use a pc for program development. And, in order to connect traditional appliances to the network, we use the smart remote control Nature Remo. And, in our prototype system, 3 types of appliances were used for development. The specific information of these devices and the appliances used in our prototype system are as Table 6.1 shows, and Nature Remo 3 [36] is shown as Figure 6.1.

Table 6.1 The information of Hardware

Device	**Information**
Android smart phone	XIAOMI 11
Smart remote control	Nature Remo 3
Air conditioning	Daikin arc478a15
Light	Daiko tdtnb907-ch1
Sweeping robot	Eufy roboVacB201

Fig. 6.1 Nature Remo 3

6.2 Development Environment

This system uses the following interfaces and software for development:

1. ReCap photo, a cloud-based modeling tool, provided by Autodesk company helps users create high-resolution textured 3D models from photos using the power of cloud computing. This software is used by users to create 3D models from objects in daily life.

2. The 3ds Max is a professional 3D computer graphics program for making 3D animations, models, games and images. It is provided by Autodesk company and is used to process the model used in the prototype system.

3. ARcore[37] also known as Google Play Services for AR, is a software development kit developed by Google that allows for augmented reality applications to be built. We use it to provide AR contents for Android system.

4. The interface Nature Remo API provides the reading from Nature Remo sensors, and the sending of infrared signals from Remo. It is used to connect to traditional infrared

remote control appliances by Nature Remo. The python API robovac is used to connect the sweeping robot.

Unity is used to develop AR applications, and Microsoft Visual Studio 2019 is used to write C# scripts. The versions of some of the software and services used are shown in Table 6.2.

Table 6.2 Software and Services

Device	Information
Unity	2019.4.22f1
3DsMax	3Ds Max 2021
Microsoft Visual Studio	Microsoft Visual Studio Community 2019 V16.8.5
ARcore	arcore-unity-sdk-1.22.0

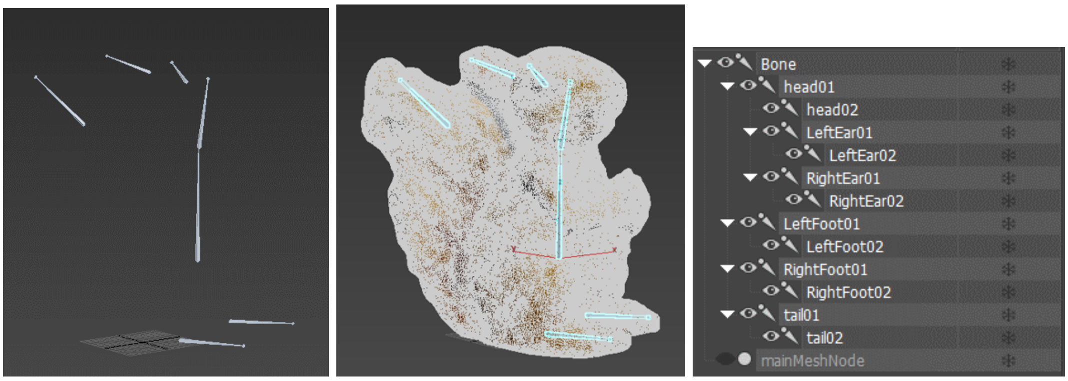
6.3 3D Model Preprocessing

In this system, users can upload photos taken by themselves through ReCap photo and build 3d models. Modeling requires users to take photos from multiple angles of objects with rich feature points. But such a model may be flawed and cannot make any movements. So, in our prototype system, we need to do some preprocessing on these models.

6.3.1 3D Rigging

3D rigging refers to setting up an invisible skeleton for the model and binding this skeleton to the 3d model. After rigging, we can make the model do corresponding actions by controlling the bones to make movements. We use 3dsMax to help us rigging.

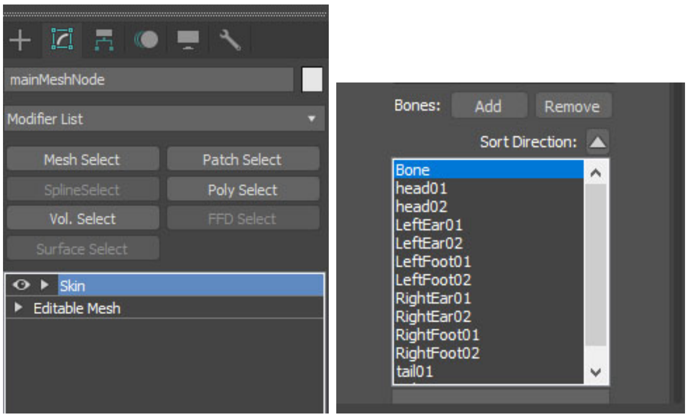
Figure 6.2 shows the bones we created for the squirrel using 3dmax. Figure 6.2a shows only the bones. Figure 6.2b shows the mesh and bones at the same time that you can see the correspondence between them. Figure 6.2c is the bone structure in 3dmax, you can see that we have performed rigging for the body and head, two ears, two feet, and tail.



(a) Bones (b) Bones with mesh (c) Bone structure

Fig. 6.2 3D Rigging for squirrel model

After creating the bones, we need to bind the bones to the corresponding mesh. As shown in the Figure 6.3, we use the skin modifier in 3dsMax. It lets use create character animation by deforming a skin mesh with bones.



(a) Add modifier skin (b) Add bones

Fig. 6.3 Add bones for mesh

After rigging is successful, we can freely create animation for the model. We define key points on the timeline in 3ds Max. As shown in the Figure 6.4, the green cuboid is the corresponding different key. 3ds Max will automatically supplement the transition key between the two key points. Then, we can finally output the animated model.

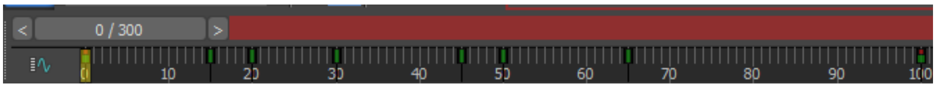
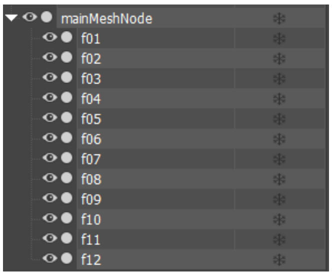


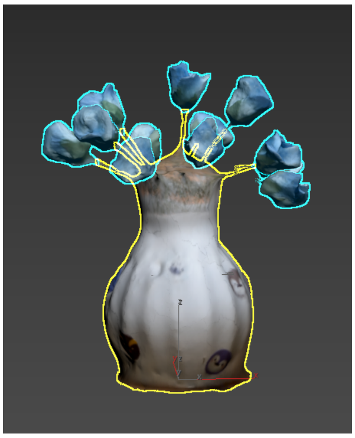
Fig. 6.4 Timeline with action key

6.3.2 Detach Model

The Detach command detaches the selected object from its group. In this prototype system, we divided the vase and the flower into 13 separate parts as shown in the Figure 6.5. It includes a vase named mainMeshNode and 12 flowers named from f01 to f12.



(a) Mesh structure



(b) Detached model

Fig. 6.5 Detached flower vase model

6.4 Connect home appliances

6.4.1 Framework

Nature Remo is used in this system to connect traditional appliances to the Internet. This system obtains the environmental data and house appliances data from Nature Remo by accessing the Nature Remo API.

Nature Remo API has two access methods, cloud and local. When the HTTP client is within the same local network with Remo, the client can discover and resolve IP of Remo using Bonjour, and then send a HTTP request to the IP.[2] Otherwise, when the client which is our system and Remo are not in the same subnet, the cloud API needs to be used.

When we use cloud API, the structure of the system is shown in the Figure 6.6. The client accesses the Nature Remo API to send an HTTP request, which is transmitted to the Nature Remo device via the Internet. Then Nature Remo sends out corresponding infrared signals to control the electrical appliances.

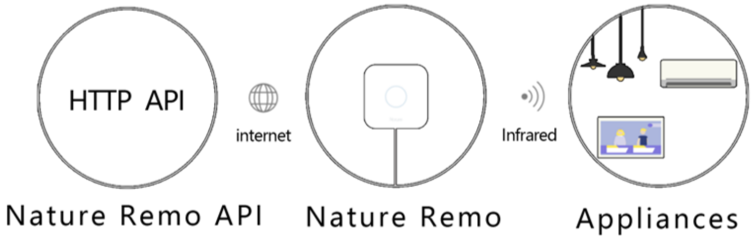


Fig. 6.6 Structure of Nature Remo

6.4.2 Sending HTTP Requests and Receiving HTTP Response

To use the service of Nature Remo API, system needs to send HTTPS requests to Nature Remo servers. And then, the receiving HTTP responses are formatted in JSON. By accessing the API, the system can obtain the current device status via GET and control the device via POST.

Our project is developed using unity and C#, in order to use C# to send HTTP requests and receive HTTP response, we use HttpClient in .Net Framework. It provides a class for sending HTTP requests and receiving HTTP responses from a resource identified by a URI.[3]

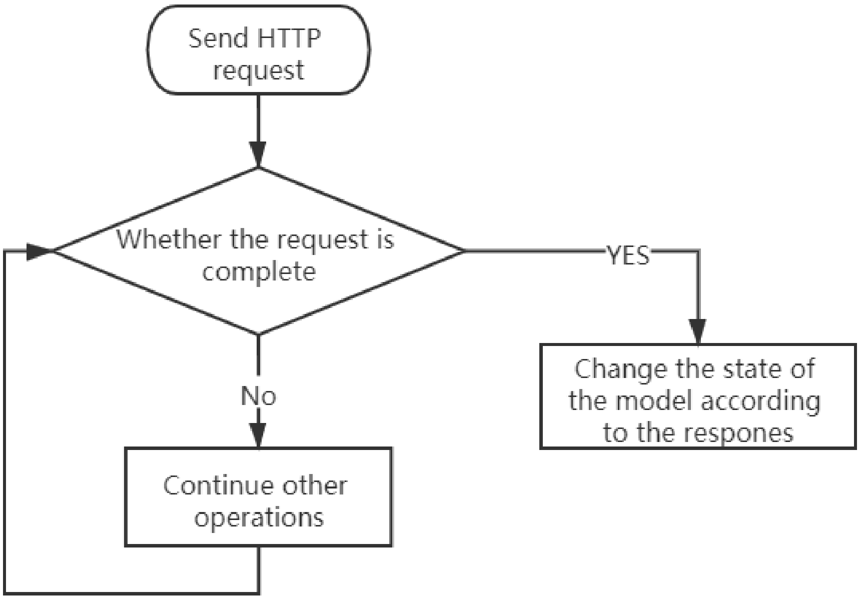


Fig. 6.7 asynchronous process

After we send the HTTP request, we need to wait for it to get response. This takes time, but Unity cannot stop the change of each frame of the screen, otherwise it will make user feel stuck. So, we use “async await” to use asynchronous programming, and the system can still perform other operations while waiting for the HTTP return. The asynchronous process is shown in Figure 6.7.

After sending the HTTP request and receiving the response, we need to analyze the received JSON data. C# cannot directly read JSON data, so in this system, we use the package Newtonsoft.Json to help process JSON data. Figure 6.8 shows the conversion of JSON data into C# classes. In this way, we can read the parameters in any desired key-value pair.

Fig. 6.8 Conversion of JSON data into C# classes

Left part is the data obtained through the request method GET to access the interface “/1/devices”. The value of “il” in section “newest_event” represents illumination, and the value of “te” represents temperature. These two values are what we mainly need. If we only want to read the temperature, then we only need to access ".newest_events.te.val" to get the temperature data.

6.4.3 Python Script

In order to connect the sweeping robot, we use its official interface robovac. But this interface only has the python version, so we need to use Unity to start the python script.

As Figure 6.9 shows, the essence of Unity calling the python script is to directly execute the python script, then get the return value from it.

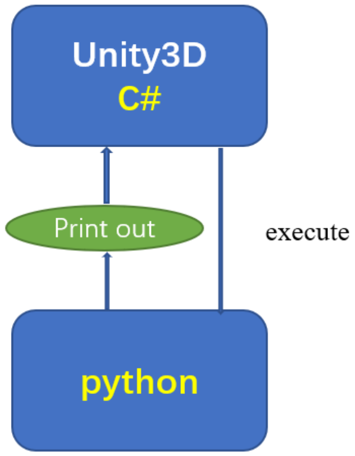


Fig. 6.9 Unity calling python script

In order to use C# to call python, we need to use the class process in .NET. This class provides access to local and remote processes and enables you to start and stop local system processes.

6.5 Environment Recognition

In this system, ARcore is used to help the environment recognitionin the AR system., which uses three key technologies to integrate virtual content with the real world as seen through your phone's camera. These allow the phone to understand and track its position, to detect the plane and lighting estimation.

For plane detection, as Figure 6.10 shows, we use the script DetectedPlaneGenerator provided by ARcore to detect the plane and instantiate a prefab on the it to display the position of the detected plane.

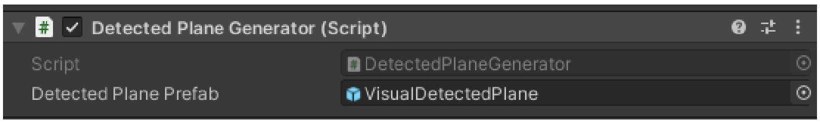


Fig. 6.10 Detected Plane Generator

After detecting a suitable plane, we need to place the AR model on the plane. In order to use the 2d mobile phone screen to specify three-dimensional position in the real world, ARcore uses Raycast.

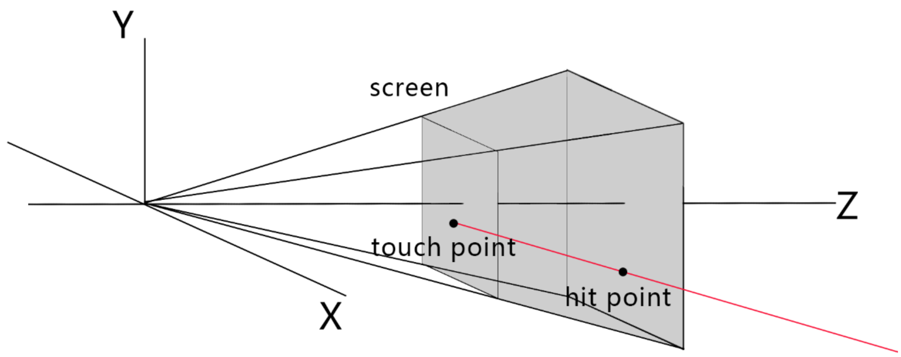


Fig. 6.11 Schematic diagram of Raycast

The basic idea of Raycast as Figure 6.11 shows is to emit an infinitely long ray line in one direction from a point in the three-dimensional world. I0 the direction of the ray, once it collides with something, collision detection will be generated. To the object. We can use the position of the camera to construct a ray from the point the user clicks on the screen, and perform collision detection with the plane in the scene. If a collision occurs, return to the collision position, so that we can place the 3d model on the detected plane.

6.6 Model control

6.6.1 Animation Control

In the prototype system, we used the squirrel model to control the air conditioner. And use squirrel sleeping or waking to represent different states of air conditioner. As mentioned earlier, we use 3ds Max to make these animations.

The animation obtained in this way is a connection with multiple together, at first, we need to split them.

As shown in the Figure 6.12 shows, there are four slices obtained after segmentation, where step is a transition animation,

Clips	Start	End
normal	0.0	81.0
step	100.0	127.0
sleep	130.0	195.0
eat	200.0	279.0

Fig. 6.12 animation clips

Then, connect these animations using Unity's Animator Controller. An Animator Controller allows us to arrange and maintain a set of Animation Clips and associated Animation Transitions for a character or object. The Animator Controller has references to the Animation clips used within it, and manages the various Animation Clips and the Transitions between them using a State Machine.

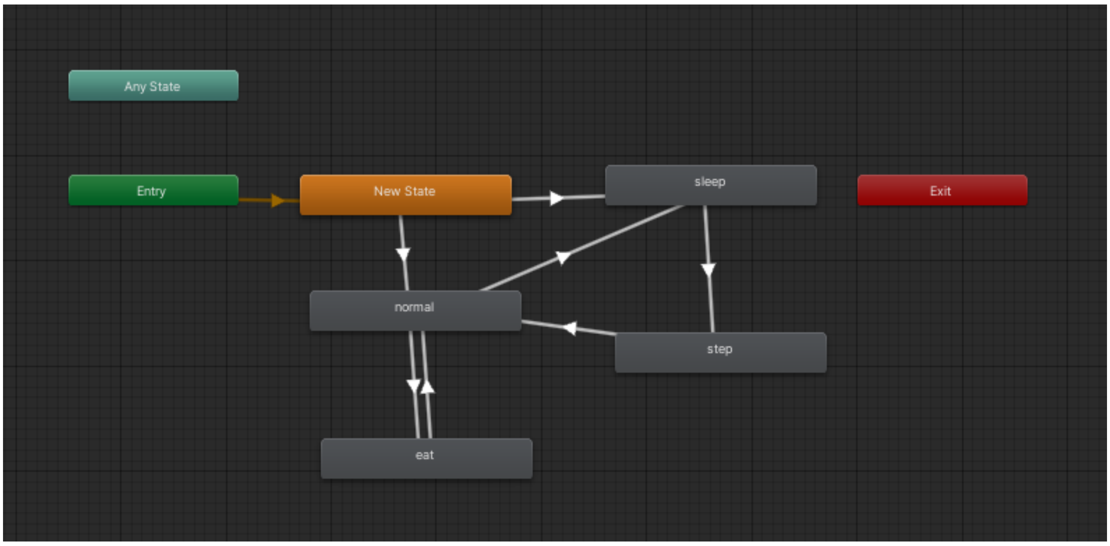


Fig. 6.13 Animator Controller for squirrel

Here, in Figure 6.13, we have implemented part of the process as shown in the flowchart (Figure 5.6). Among them, the Entry module is the entrance of the animation. When the model is activated, it will automatically start from the Entry module. After that, you will enter the StatusJudge module, which contains two Transitions. They are controlled by the variable AirConSwitch shows in Figure 6.14 .

Fig. 6.14 Variable AirConSwitch

If the air conditioner is turned on, then AirConSwitch is true, it will enter the normal state (as shown in the Figure 5.5a. If the air conditioner is turned off, then AirConSwitch is false, it will enter the sleep state(as shown in the Figure 5.5c. When the squirrel is in the sleep state, if the user touches the squirrel model and turns on the air conditioner, the variable AirConSwitch will become true, and the squirrel will go through a transitional state step to become the normal state. When the squirrel is in the normal state, if user turns down the air conditioner, the variable AirConSwitch will become false, and the squirrel will go directly to sleep. When the squirrel is in the normal state, that is to say, the air conditioner is turned on.

If the user feeds the squirrel, which is the collision between squirrel and Apple in our system, then Apple's GameObject will be destroyed, and the trigger eat will be activated, the state will be changed from normal to eat, and the squirrel will make an eat action, and then automatically change back to normal.

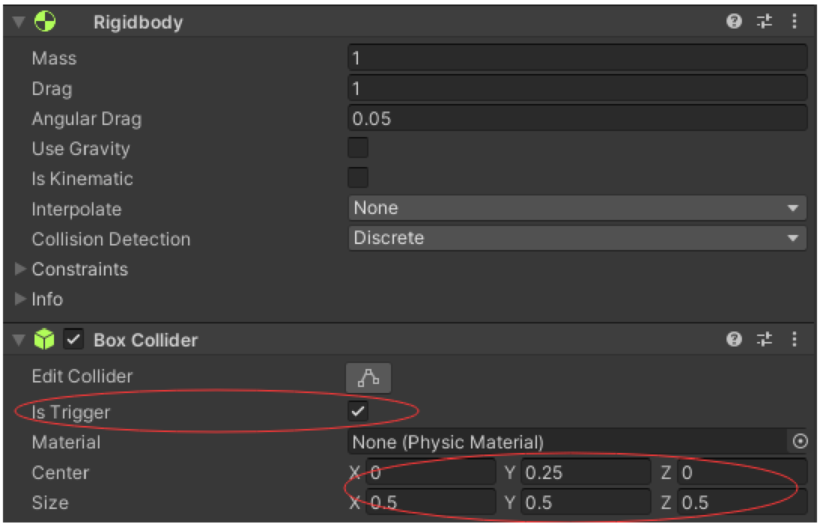


Fig. 6.15 Unity components "Rigidbody" and "Box Collider"

In order to detect the collision between squirrel and apple, we need to add 2 components "Rigidbody" and "Box Collider" as Figure 6.15 shows to both objects. Adding a Rigidbody component to an object will put its motion under the control of Unity's physics engine. a Rigidbody object will react to collisions with incoming objects when "Collider" component is also present. Here we have selected the "is Trigger" in the "Box Collider" component, so that when the apple and the squirrel collide, they will not knock each other away. In addition, we also set the collision volume and collision center for them that match the model.

6.6.2 Activation Control

In the vase and light scene, the number of flowers in the vase will be changed, as said before, we use 3ds Max to divide the flower vase into a vase and 12 flowers, and we need to display different numbers of flowers according to different conditions of the light. And also, in the air conditioner and squirrel scene, the squirrel will display the text "zzz" only when it sleeps. In order to achieve these effects, we control the activation state of the flower and text "zzz" by C# script.

All objects that appear in the Unity scene are GameObjects, and GameObjects have activation properties. Each item in the Figure is a GameObject, the one that is lit is the active state, and the one that is gray is the inactive state. The actived GameObject will be displayed in the scene, and the components mounted on it will also work. The GameObject that is not activated will not be displayed, and the components mounted on it, such as scripts, will not run.

And in the scene of using the locomotive, we use a script to let him change the resource of the texture at a certain time interval to realize the function of flickering. So, when we need it to flash, we activate this script, otherwise we make this script inactive. This script will run automatically after being activated.

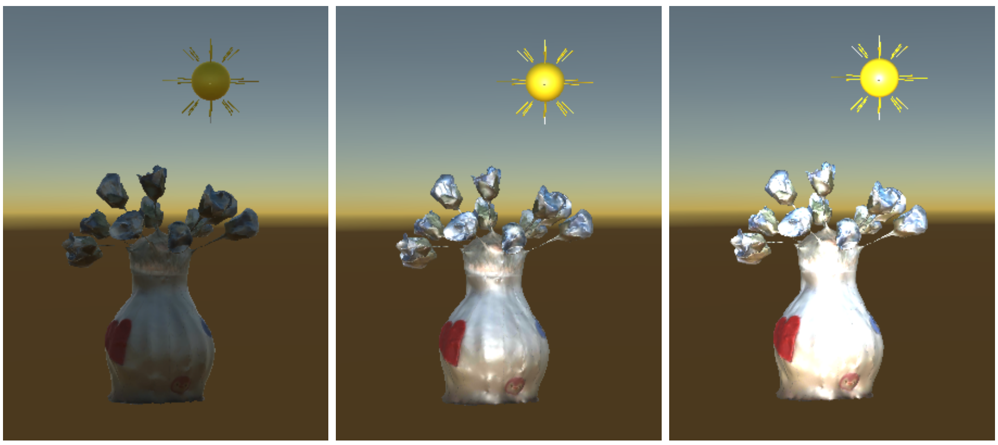
6.6.3 Lighting Intensity Control

Directional lights are very useful for creating effects such as sunlight in your scenes. All objects in the scene are illuminated as if the light is always from the same direction. The distance of the light from the target object is not defined and so the light does not diminish. The light source has many parameters, including color, mode and intensity.

In this system, we change the brightness of the vase by controlling the intensity of the Directional light to remind the user of the illumination in the home. Among them, the relationship between the intensity of the Directional light and the illumination is set to be linear, as in formula.

$$Y_{intensity} = kX_{illumination}$$

The intensity of Directional light $Y_{intensity}$ ranges from 0 to 2.08, the range of illumination $X_{illumination}$ is 0 to 160, and k is a constant with a value of 0.013. As Figure 6.16 shows, the default value of the intensity of Directional light is 1 .At this time, its lighting is normal. If it is greater than 1, it will appear very bright, and if it is less than 1, it will appear dark. We set the value of the intensity of Directional light to 0 2.08 because this range will not cause abnormal illumination, and it can also express the changes in illumination vividly. Illumination is the current illumination data we get from Nature Remo. It may range from 0-160. It is 160 when the curtain is opened during the day and the light is turned on, and it is 0 when the curtain is closed and the light is not turned on at night.



(a) Intensity = 0 (b) Intensity = 1 (c) Intensity = 2

Fig. 6.16 Different intensity of Directional light on the vase

6.6.4 Fast moving judgment

When we are in the control of the locomotive model, the user can move the model quickly to simulate the action of cleaning the floor. In order to simulate this kind of action judgment, we use the two parameters of user's moving distance and moving speed to judge whether the user is performing this action. The Unity program is executed every frame, and a function written in Update will be executed once every frame. This is because the picture that the user sees needs to be refreshed every frame.

$$Distance_{\,1} = \text{Distance}_0 + \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2 + (z_1 - z_0)^2}$$

In this system, we set a unit to calculate the distance from when user clicks. After the user clicks, the distance will continue to be accumulated until the user releases the finger. And then when user's finger leaves the screen, the distance will return to 0. Because the position of the Object in Unity3D is represented by a three-dimensional coordinate, the distance can be calculated using formula.

Among them, $Distance_0$, x_0, y_0 and z_0 respectively refer to the distance and coordinates of the previous frame, $Distance_1$, x_1, y_1 and z_1 represent the time of this frame. In the code, in order to calculate the square and root sign, we use the Maths package. Then, in order to calculate the speed, we need to know the interval time we use Time.deltaTime. This is a read only data that give the interval in seconds from the last frame to the current one. So, he velocity is just like the formula.

$$velocity = \frac{\sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2 + (z_1 - z_0)^2}}{Time.deltaTime.}$$

6.6.5 Moving Object

In order to allow users to place the selected model in any position on the detected plane, we have implemented the moving function. For this, we first need to add components

"Rigidbody" and "Box Collider" as mentioned in 5.6.1. Then, we add C# scripts for each object that needs to be controlled. The core code in the script is shown in the Figure 6.17.

```
if (Physics.Raycast(ray, out hitObject) && !isObjectSelect)
{
    moveObject = hitObject.collider.gameObject;
    moveObjectName = hitObject.transform.name;
    isObjectSelect = true;
}
if (name == moveObjectName)
{
    Frame.Raycast(positionX, positionY, raycastFilter, out Thit);
    temp = Thit.Pose.position;
    transform.position = temp;
}
```

Fig. 6.17 Core code of moving object

As shown in the code in Figure 5.15, "isObjectSelect" is a Boolean value that determines whether an object has been selected which is initially be false. "hitObject" represents the hit object, and "Thit" represents information about hitting the detection plane.If no object is selected and the ray is successfully emitted, the value of "isObjectSelect" is changed to true to indicate that the object hit by the ray is selected. Then it is judged whether the object hit by the ray is the current object, if so, the point where the ray hits the detection plane is regarded as the new position of the object. In this way, the object is moved.

When we want to move the sun, it needs to be moved up and down within a certain range. So, we use the core code shown in Figure 6.18 to control it.

```
y = positionY - recordPY;

if (transform.position.y - flower.transform.position.y < MAX && flower.transform.position.y - transform.position.y < MIN)
        temp = new Vector3(transform.position.x, transform.position.y + 0.001f * y, transform.position.z);
        transform.position = temp;
}
recordPY = positionY;
```

Fig. 6.18 Core code of moving in y axis

Among them, "recordPY" is the recorded position of y axis, it records the position of the sun in previous frame. "positionY" is the y axis coordinate of the sun in the current frame. The variable "y" represents their difference. If the current range of the sun relative to the flower is within a certain value, let the sun move in the direction (up or down) specified by the user. Finally, record the current y-axis position as the "recordPY" of the next frame.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

In this thesis, we designed and implemented an AR smart house system that allows user use customized AR 3d model to control IoT devices which are used in their house to provide a more intuitive feedback and a more interesting interaction. In recent years, IoT devices are used by more families to build their own smart houses. Although the existing smart speaker can meet many needs of users, it is difficult for users to use in some scenarios. Also, these related systems cannot provide a good feedback. This research uses AR technology to provide a new experience for the smart house control. And, we aim to provide a system using customized 3d model in AR to give user a more intuitive feedback and a more interesting interaction.

The system could be divided into 2 main parts:

We use multiple photos with normal angles taken by mobile phone to reconstruct the 3D model. Then, bind the bones for them to make the model movable, and finally make them controllable by the users. In order to put the model into the AR environment, we need to recognize the environment. So, we introduce ARCore to recognize the environment.

System can connect the appliances that only support infrared remote control to change them to the IoT devices by using Nature Remo. We connect the AR model with the corresponding IoT device, and use these models designed and implemented different intuitive interaction and feedback methods for different appliances. User can control the IoT device through the control model, and the IoT device can also give feedback to the user through the model.

In general, we designed and achieved an AR-based smart house control system using customized interactive 3d model. The system allows users to reconstruct a 3d model using photos from multiple angles of the object that they prefer. Multiple models can be created, and can be connected to the corresponding IoT devices. Then users can put these models into the AR environment via mobile phone, and control the IoT device by interacting with model, and the model can reflect the state of environment and the device through unique actions.

7.2 Future Work

Our system is a new attempt at the human-machine interface of smart house. We have designed and implemented some typical and very commonly used scenarios, but there is still room for expansion in the usage scenarios, including more appliances and more interactions. According to some previous work, we can see that the use of cute 3D models can promote people's behavior to a certain extent. Studying the role of these effects in the AR smart house system may be a future research direction. Technically, because of the Internet transmission speed, the system has some non-negligible delays. Solving this problem may also be a topic in the future.

References

[1] Mark Billinghurst, Adrian Clark, and Gun Lee. A survey of augmented reality. 2015.

[2] Pranay P Gaikwad, Jyotsna P Gabhane, and Snehal S Golait. A survey based on smart homes system using internet-of-things. In *2015 International Conference on Computation of Power, Energy, Information and Communication (ICCPEIC)*, pages 0330–0335. IEEE, 2015.

[3] Google. Google assistant, your own personal google. https://assistant.google.com/, 2015.

[4] Philips. Philips hue: Smart lighting. https://www.philips-hue.com/en-us, 2015.

[5] Ronald Azuma, Yohan Baillot, Reinhold Behringer, Steven Feiner, Simon Julier, and Blair MacIntyre. Recent advances in augmented reality. *IEEE computer graphics and applications*, 21(6):34–47, 2001.

[6] Paul Milgram and Fumio Kishino. A taxonomy of mixed reality visual displays. *IEICE TRANSACTIONS on Information and Systems*, 77(12):1321–1329, 1994.

[7] Sonia Schechter. What is markerless augmented reality? | marxent. https://www.marxentlabs.com/what-is-markerless-augmented-reality-dead-reckoning/, 2014.

[8] Wolfgang Hürst and Joris Dekker. Tracking-based interaction for object creation in mobile augmented reality. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 93–102, 2013.

[9] Sunny Consolvo, David W McDonald, and James A Landay. Theory-driven design strategies for technologies that support behavior change in everyday life. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 405–414, 2009.

[10] Sora Inomata, Kosuke Komiya, Koya Iwase, and Tatsuo Nakajima. Ar smart home: a smart appliance controller using augmented reality technology and a gesture recognizer.

[11] Kang Bing, Liu Fu, Yun Zhuo, and Liang Yanlei. Design of an internet of things-based smart home system. In *2011 2nd International Conference on Intelligent Control and Information Processing*, volume 2, pages 921–924. IEEE, 2011.

[12] TechTarget Contributor. Smart speaker. https://whatis.techtarget.com/definition/smart-speaker, 2017.

[13] Dong Woo Seo, Hyun Kim, Jae Sung Kim, and Jae Yeol Lee. Hybrid reality-based user experience and evaluation of a context-aware smart home. *Computers in Industry*, 76:11–23, 2016.

[14] Gary White, Christian Cabrera, Andrei Palade, and Siobhán Clarke. Augmented reality in iot. In *International Conference on Service-Oriented Computing*, pages 149–160. Springer, 2018.

[15] Dongsik Jo and Gerard Jounghyun Kim. Iot+ ar: pervasive and augmented environments for “digi-log” shopping experience. *Human-centric Computing and Information Sciences*, 9(1):1–17, 2019.

[16] Laehyun Kim, Wanjoo Park, Hyunchul Cho, and Sehyung Park. An universal remote controller with haptic interface for home devices. In *2010 Digest of Technical Papers International Conference on Consumer Electronics (ICCE)*, pages 209–210. IEEE, 2010.

[17] Dongyu Wang, Kazunori Sugiura, and Yui Murase. Design and implementation of user-centered home appliance controlling service environment. In *Proceedings of Workshop on Mobile Video Delivery*, pages 1–6, 2014.

[18] Abdulla AlHammadi, Aisha AlZaabi, Bashayer AlMarzooqi, Suhail AlNeyadi, Zayed AlHashmi, and Maad Shatnawi. Survey of iot-based smart home approaches. In *2019 Advances in Science and Engineering Technology International Conferences (ASET)*, pages 1–6. IEEE, 2019.

[19] Michal Luria, Guy Hoffman, and Oren Zuckerman. Comparing social robot, screen and voice interfaces for smart-home control. In *Proceedings of the 2017 CHI conference on human factors in computing systems*, pages 580–628, 2017.

[20] Stephen S Intille. Designing a home of the future. *IEEE pervasive computing*, 1(2):76–82, 2002.

[21] Erik Kučera, Oto Haffner, and Štefan Kozák. Connection between 3d engine unity and microcontroller arduino: A virtual smart house. In *2018 Cybernetics & Informatics (K&I)*, pages 1–8. IEEE, 2018.

[22] Rafael Giménez, Marc Pous, et al. Augmented reality as an enabling factor for the internet of things. In *Proceedings of the W3C Workshop: Augmented Reality on the Web, Barcelona, Spain*, pages 15–16, 2010.

[23] BinaryBanana. smartarhome. http://smartarhome.com/press/, 2018.

[24] Rina Umeyama and Hidekazu Suzuki. ihac: Smart appliance controller using ar technology. In *2017 IEEE International Conference on Consumer Electronics (ICCE)*, pages 168–169. IEEE, 2017.

[25] Dongsik Jo and Gerard Jounghyun Kim. Ar enabled iot for a smart and interactive environment: a survey and future directions. *Sensors*, 19(19):4330, 2019.

[26] Ching-Hu Lu. Iot-enhanced and bidirectionally interactive information visualization for context-aware home energy savings. In *2015 IEEE International Symposium on Mixed and Augmented Reality-Media, Art, Social Science, Humanities and Design*, pages 15–20. IEEE, 2015.

[27] Maribeth Gandy and Blair MacIntyre. Designer's augmented reality toolkit, ten years later: implications for new media authoring tools. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, pages 627–636, 2014.

[28] Shinya Mihara, Kohei Kawai, Hideki Shimada, and Kenya Sato. Evans 3: Home appliance control system with appliance authentication framework using augmented reality technology. In *2013 IEEE 10th Consumer Communications and Networking Conference (CCNC)*, pages 849–850. IEEE, 2013.

[29] Atieh Mahroo, Luca Greci, and Marco Sacco. Holohome: an augmented reality framework to manage the smart home. In *International Conference on Augmented Reality, Virtual Reality and Computer Graphics*, pages 137–145. Springer, 2019.

[30] Keonhee Cho, Hyeonwoo Jang, Lee Won Park, SeungHwan Kim, and Sehyun Park. Energy management system based on augmented reality for human-computer interaction in a smart city. In *2019 IEEE International Conference on Consumer Electronics (ICCE)*, pages 1–3. IEEE, 2019.

[31] Philipp A Rauschnabel, Alexander Rossmann, and M Claudia tom Dieck. An adoption framework for mobile augmented reality games: The case of pokémon go. *Computers in Human Behavior*, 76:276–286, 2017.

[32] Isaac Wang, Jesse Smith, and Jaime Ruiz. Exploring virtual agents for augmented reality. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2019.

[33] Nahal Norouzi. Augmented reality animals: Are they our future companions? In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pages 563–564. IEEE, 2020.

[34] Telmo Zarraonandia, Alvaro Montero, Paloma Diaz, and Ignacio Aedo. " magic flowerpot": An ar game for learning about plants. In *Extended Abstracts of the Annual Symposium on Computer-Human Interaction in Play Companion Extended Abstracts*, pages 813–819, 2019.

[35] Lele Feng, Xubo Yang, and Shuangjiu Xiao. Magictoon: A 2d-to-3d creative cartoon modeling system with mobile ar. In *2017 IEEE Virtual Reality (VR)*, pages 195–204. IEEE, 2017.

[36] NATURE INC. Homepage | nature remo. https://en.nature.global/, 2021.

[37] Google. Build new augmented reality experiences that seamlessly blend the digital and physical worlds. https://developers.google.com/ar, 2021.