Cooperation Between Multiple Markers in Augmented Reality

Ling Du

44181033-0

Master of Engineering

Supervisor: Prof. Jiro Tanaka

Graduate School of Information, Production and Systems Waseda University

February 2020

Abstract

Nowadays based on the development of image recognition and AR(Augmented Reality) technology, people can detect some real images, add and show 3D virtual contents above images. However common system usually has two limitations which strongly influence user experience, the first is lack of cooperation between different virtual contents, the second is lack of interaction between users and AR contents.

This research aims to improve user experience while using marker-based AR system by building a new system which can enhance the cooperation between multiple markers and their contents in Augmented Reality.

Our system consists of two main parts, controller and module. Firstly system will recognize different designed markers which contain virtual contents and extra information from the feature points, and show their virtual contents. Next, it generates variable interfaces from controller and module markers for users to connect them with main object. Last, user can do directly manipulate on AR device or physical markers to achieve supplementary functions like movement control and modules combination, system will show the change of main virtual object in real time.

We have invited some participants to test the usability and efficiency of our system, the results of preliminary user study have shown the positive feedback.

Keywords: Augmented Reality, Multiple Markers, Cooperation, Interactive

Acknowledgements

In this part, I want to express my sincere gratitude to all the people who helped me in the progress of researching and writing master thesis.

First and foremost, I want to show my great appreciation to my supervisor, Prof. Jiro Tanaka, for the support on my study and research. Thanks to his patient guidance and affable encouragement, I had a chance to open my mind and start an academic journey in HCI field, especially Augmented Reality. Without suggestions and devices got from him, it should be impossible to complete this study.

Secondly, I would like to thank all members of IPLAB, they are good partners in research and life. Thanks so much for many helps in my life and research, the brain-storming always give me lots of inspiration and pleasure.

And Last but not least, I want to thank my parents and all my friends. For my parents, they give me financial support and warm mental support throughout my master study life. My friends always give me courage and push me ahead, I will keep going on and making progress to return all those people.

Contents

List of figures

List of tables

Chapter 1

Introduction

1.1 Introduction

In recent years, AR (augmented reality) system has become very popular[2]. AR system has been improved to support more and more information which is usually displayed as 3D-virtual content. One of the most common AR system is the marker-based AR system[8]. Under a marker-based AR system, through a camera, the device can detect the spatial information of physical markers or images as a input. Then the system will access to database to get the related virtual content, and show the combination of real word and computer-generated graphics as the output through the AR display devices (such as mobile phone or AR glasses).

With the development of the Augmented Reality, users need AR system to do more things, which is not restricted to display more diverse information and adapt to more situations, sometimes users want to control the virtual contents or edit the existing object without remaking the physical marker, especially for multiple markers and content. This is essential to flexibility and augmentability of marker-based AR system and also beneficial to user experience. There should be some methods to support cooperation among different virtual contents generated by multiple markers, and improve the user experience in a more attractive and intuitive way.

In this study, we tend to use multiple markers or images to make some cooperation for more attractive and abundant user experience in marker-based AR system.

1.2 Organization of the thesis

The rest of this thesis is organized as follows: Chapter 2 introduces the background about the thesis. Chapter 3 will talk about the related work. Chapter 4 will describe the existing problems, research goal and the approach. Chapter 5 is the system design part, where the design concept and ideas will be introduced in detail. Chapter 6 presents the system implementation part which is about the detailed devices, environment and implementation. Chapter 7 will be about the experiments, we will talk about the performance of our approach. The last part, Chapter 8, will be conclusion and future work part, where we will conclude the previous content and talk about the future possibilities.

Chapter 2

Background

2.1 Augmented Reality

2.1.1 Definition

Augmented Reality (AR) was defined in 1997 by a researcher who is called Ronald T. Azuma[1]. From that time, it's considered as a field of computer science research that combines real world and digital data.

In Ronald's survey, he has given three key characteristics:

• Combines real and virtual content,

• Interactive in real time,

• Registered in 3D.

Generally speaking, augmented reality is a new approach or technology which can make the computer interface invisible and enhances user interaction with the virtual and real world[8]. These three characteristics also define the technical requirements of an AR system, namely that it has to have a display that can combine real and virtual images, a computer system that can generate interactive graphics the responds to user input in real time, and a tracking system that can find the position of the users viewpoint and enable the virtual image to appear fixed in the real world.

So it means AR system need to be based on real world, and it cannot completely replace it. Ideally, users seem to have virtual and real objects coexisting in the same space[1], the elements which are be enhanced including vision, hearing, touch, body and smell[9].

Since the emergence of AR technology, AR has been widely used in many different fields, for example medical visualization, maintenance and repair, annotation, robot path planning, entertainment, military aircraft navigation and targeting and so on.

Fig. 2.1 A simple traditional AR example

The figure2.1 shows a simple example of traditional Augmented Reality, the paper, desk and laptop are real ones, the pumpkin is a virtual one. These two kinds of things can be put together and shown at the same time.

So under augmented reality, users can interact with the real world at the same time as the virtual images, bringing the benefits of VR interfaces into the real world and facilitating very natural collaboration[12].

2.1.2 Techniques

For the display and manipulation of AR, it needs some latest mobile computing devices, such as smartphones and tablets, containing cameras and MEMS sensors that includes accelerometers, GPS, solid state compasses and others, are the appropriate augmented reality platforms[10]. The newest device is Head-Mounted Display (HMD), it is worn on the head or as part of a helmet, and has a small display optic in front of one or each eye. HMD can display the virtual content in the real world, and support user to do some manipulation on those virtual ones by many methods, such as extra controller, gesture, voice command, eye tracking and more.

That is the support of augmented reality at the hardware level, surely it needs some support at the software level. So an important issue of the AR system is how to truly combine virtual objects with the real world. In order to deal with this, some software development kits (SDKs) are made and developed quickly, which can enable rapid development of augmented reality applications. As the fig2.2 shows, there are some various AR SDKs: Vuforia, Wikitude, ARCore, and ARKit.

(a) Vuforia (b) Wikitude (c) ARCore (d) ARKit 3

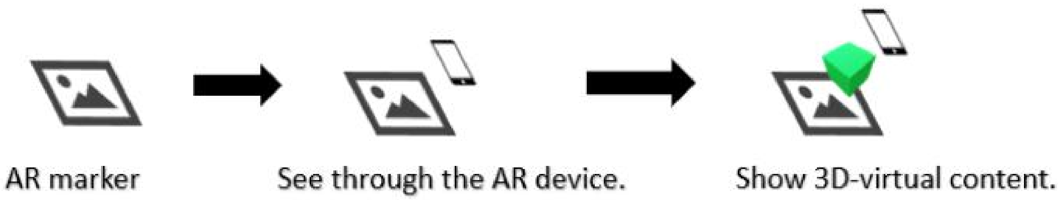
Fig. 2.2 Common AR SDKs

2.1.3 Marker-based Augmented Reality System

In Augmented Reality, we need to anchor the digital world to the real world, and add some 3Dvirtual contents into the real world, for that it is necessary to know whether user is pointing the camera at that particular environment or not, like that human moving eyes on something or not. In current AR system, this need can be achieved by placing a distinctive picture or shape and scanning.

After setting and scanning, the distinctive pictures will be recognized and system will display the virtual contents immediately, tracked to the appropriate place in the real world. The user can also move the physical picture around and see the virtual world "stick" to the real surface. We call the distinctive picture that can be recognized by the device, the AR marker, and the system is usually called marker-based AR system[13].

AR markers are widely used in the existing AR systems. For example, ArToolKit, Auto-assembly, Outdoor tracking, CyliCon, ArLoc, CyberCode and other more. They all use visual marker based systems for motion tracking and pose estimation[14].



(a) System flow

(b) 3D virtual model shown above marker

Fig. 2.3 Marker-based Augmented Reality System

Nowadays Maker-based AR system has been used for many applications, especially something can enrich printed media, for example, Esquire magazine published an augmented reality issue in December 2009, in that case, users were able to see AR content when they showed the magazine to a PC webcam[8].

And in 2010, *Süddeutche Zeitung* released their first issue with AR content. In the case of that, users could see the content with a mobile phone after downloading the application. In Finland, *Katso* and *TVSeiska* magazines used AR in cooperation with VTT in advertising a new animated children series called *Dibitassut* in April 2010. Brazilian newspaper *O estado de Sao Paulo* has featured regular AR content since 2009[5].

Chapter 3

Research goal and Approach

3.1 Problem

In current system, the AR contents generated by different markers are usually independent[8]. The independence is shown as following two aspects:

1. Just show their own contents.
2. Contents cannot cooperate with each other.

Fig. 3.1 Multiple independent AR contents

Usually marker-based AR system just shows the virtual contents above markers separately and independently, as figure3.1 shows, two different markers shows two different warriors but there is no relationship between these two models.

If user wants to combine some single objects for a further operation, the current system cannot provide a general method without specific programming and interface in software. These reasons make marker-base AR very limited and not attractive enough.

3.2 Research Goal

According to the problem in current marker-based AR system, the goal of our research is to design a system which can enhance the cooperation between multiple AR markers and contents.

Generally speaking, the system should achieve the following function:

When user put several designed AR markers together, they will not only show the original contents, but also cooperate with each other.

It should allow user to design and enjoy the stimulating virtual contents by switching between real world and augmented world.

3.3 Research Approach

In order to support various cooperation methods among different markers and improve user experience in marker-based AR system, we defined two special markers (controller and module) to cooperate with main virtual object for two main functions:

1. Additional controller generated from controller markers to control main objects

2. Additional modules generated from module markers to be equipped into main objects.

The types of markers will be distinguished by different tags ("module" and "controller") stored with virtual contents in database.

3.3.1 Controller Part

To design a controller part, we have to decide some framework.

System needs a database for designed markers to confirm type of markers first when user is using this system. After getting controller markers and connecting them to main object, user can use a general controller generated by controller marker to control movement or action of any available connected main object.

It's possible for users to change main objects and design specific control methods as they want, which can make interaction among user and multiple markers more abundant and attractive.

3.3.2 Module Part

The module part is a little similar to controller part, the framework of it is also based on some designed markers and general logic in system.

This part provides user a function to choose and edit a main virtual object model by putting various module markers and pressing buttons for connection. After system scanned module marker and show its original virtual content above image, which is just like the object marker, user can put available modules close to some main objects, system will generate extra interface for them to connect each other. Through the interface generated with models by module markers, user can easily edit the model of main object shown without changing object marker or doing extra manipulation in software.

Module part is designed to make interaction among models that are generated by some different markers, it provides user a more intuitive and attractive way to interact with static images in marker-based AR system, and can be combined with controller part to improve experience further.

3.4 Novelty

The novelty of our research mainly reflects in these aspects:

1. Design additional markers and connection logic to make it possible for markers to interact with each other. In current system, AR contents generated by different markers are usually independent but sometime we need to combine them for a further operation, current system is not enough for that.

2. Add and implement two main functions for users to edit virtual contents and customize methods to control it as they want. They don't need any complicated adjustments in a software, they just change markers and do some direct manipulation on smartphone, so the user experience can be improved.

Chapter 4

Related Work

Related works will be introduced in this session. Here we will briefly describe some of the works that are mostly relevant to marker-based augmented reality system and multiple markers.

Based on development of AR(augmented reality) research, user have chances to combine physical real environment and virtual elements, and Marker-based AR system is a kind of traditional system like that. Marker technology is an enduring research theme in augmented reality[17]. In various different researches, researchers have invented a variety of AR markers and many methods to use them for a variety of different functions.

Khushal Khairnar et al.[18] put a system which uses physical markers to show and edit virtual furniture in real world. It provided some methods to place virtual furniture and adjust layout based on designed markers. That paper also discussed some placement of different 3D models when they were put very close.

Moreover, Angeline Lee Ling Sing et al.[19] explored a multi-marker detection technique in interactive augmented reality colouring book, they used some color cards to change the color of one AR content, and also support some simple interaction among multiple contents of colouring book.

Rafaá Wojciechowski[20] has already presented a system that allows museums to build and manage Virtual and Augmented Reality exhibitions based on 3D models of artifacts. For the AR part, they placed several AR markers in front of each artworks. The user scans those AR-markers to generate the corresponding 3D model. They even designed virtual buttons for users to interact with 3D models. In addition, Yuan-Ping Luh[21] proposed a marker-based customized AR system, which builds a systematic framework to support the design customization of footwear for children, and it identified three modules related to shoe styling: shoe surface, shoe bottom, and accessory. They designed a new module, shoe cloth, which is created to allow a quick change of shoe appearance, that makes consumers have a chance to specify various design attributes in each module, including color, texture, embroidery, and shape.

There are some research which are talking about the relationship between design and user experience, such as gamification, and proposing some basic elements in the applications under the guidance of gamification. Among these typical game design elements[16], are points, badges, leader-boards, performance graphs, meaningful stories, avatars, and teammates. The research in this field inspired us to find more possibility of marker's cooperation, pushed us to design the attractive cooperation methods among multiple markers and users in marker-based augmented reality system.

Combined with multiple markers and attractive user experience, the research of AR game is also a meaningful and popular topic. It is often used as an interesting demonstration of new attempts in the field of Augmented Reality. Some of the earliest research in AR games was conducted by Ohshima and his colleagues to demonstrate advances in underlying hardware and software infrastructure, by creating the multiplayer air hockey[22] and first-person shooter[23] games from scratch. In contrast, AR Chinese Checkers [24]utilizes optically-tracked fiducial markers (printed black and white patterns whose pose and identity can be determined automatically) as input devices, in conjunction with attached wireless physical buttons for precise selection tasks. These research described many simple but useful interaction methods for physical markers and users' manipulation under marker-based AR system.

In summary, there are several researches about markers recognition technique and virtual contents manipulation techniques. The device, algorithm, user cases have many differences but also share some commons and give us lots of inspiration. Since all of the prior works have focused on single marker display and one-to-one manipulation, whereas few researches have explored possibilities of cooperation among multiple markers in marker-based AR system. What's more, most of the studies on the combination of different virtual contents did not support users to make a flat and attractive interaction with them, which reduced user experience and sense of participation. Therefore, designing and developing the cooperation methods of multiple markers is of crucial importance.

Chapter 5

System design

In this chapter, we will introduce our system design and each point of our approach. We will divide this chapter into two parts:

- Part 1 is the system overview. This part will roughly introduce how this system works.
- Part 2 is about the main composition of this marker-based AR system in details. We also divided it into two main parts, controller part and module part, and will explain the design in details.

5.1 System Overview

Mainly this system consists of two parts: Controller part and module part. Both parts are based on Vuforia database and Augmented Reality technology. Controller part can support user to choose one object and design methods to control it. Module part can support user to edit model by putting and removing module markers directly. Controller part and module part are designed to promote each other, when combining this two parts, system will enhance the cooperation among different markers and their virtual contents, make the interaction more attractive and improve the user experience in Augmented Reality. The figure5.1 shows how the system looks like

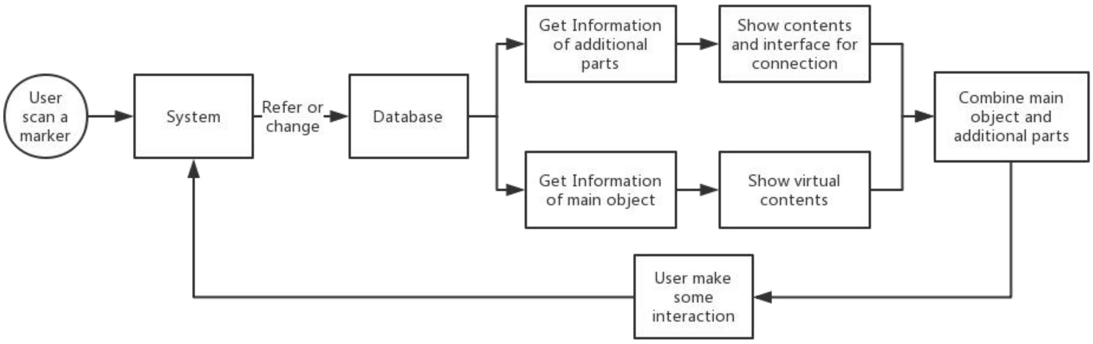


Fig. 5.1 System Overview

As figure 5.1 shown, when user views target markers or images through an AR device (smartphone, webcam with PC and so on), system will query the Vuforia database to confirm the type of marker, and generate different interfaces for object and additional parts (controller and module). If it's a type of object image, just generate virtual object and waiting for connection, if it's module image or controller image, generate its content with special interface, user can equip the module and connect the controller to the main object. After combining object, module and controller, system will show the combined virtual object with controller. This system will provide a general platform for user to change the model and design control methods by putting or removing the markers. The object marker, controller marker and module marker will do different jobs as follows:

- Object: main part, show content and ready for connection.
- Controller: additional part, generate virtual controller to control the main object.
- Module: additional part, generate virtual modules which can be attached to main object.

5.2 Controller Part

We use Vuforia database and Augmented Reality technology in this part. We designed some images as general markers. After scanning controller markers and object markers, the user can use general controllers generated by controller markers to control movement of any available connected main object. According to the user's choice, the control methods and target object can be different.

5.2.1 Usage Scenario

We designed a simple usage scenario to show how controller part is working in this system. At first system needs some designed markers for virtual objects and controllers stored in the database.

(a) Knight (b) Helicopter (c) Arrow-Key (d) Up-Down Key

Fig. 5.2 Designed object and controller markers with their feature points

User firstly uses this system to design movement control methods of virtual knight and helicopter, he needs to get four designed markers shown in figure5.2.

System can scan (a)Knight marker to generate a virtual knight, and scan (b)Helicopter marker to get a virtual helicopter, each model means main object. After getting a main object, user can scan additional control markers like (c)Arrow Key marker to generate a general joystick for horizontal movement of the main object that he chose, and scan (d)Up-Down Key marker to get a Up-Down key for vertical movement.

5.2.2 Interaction Design

In this part, we will introduce the interaction design in our system. It includes how to use it and why we design it in this way.

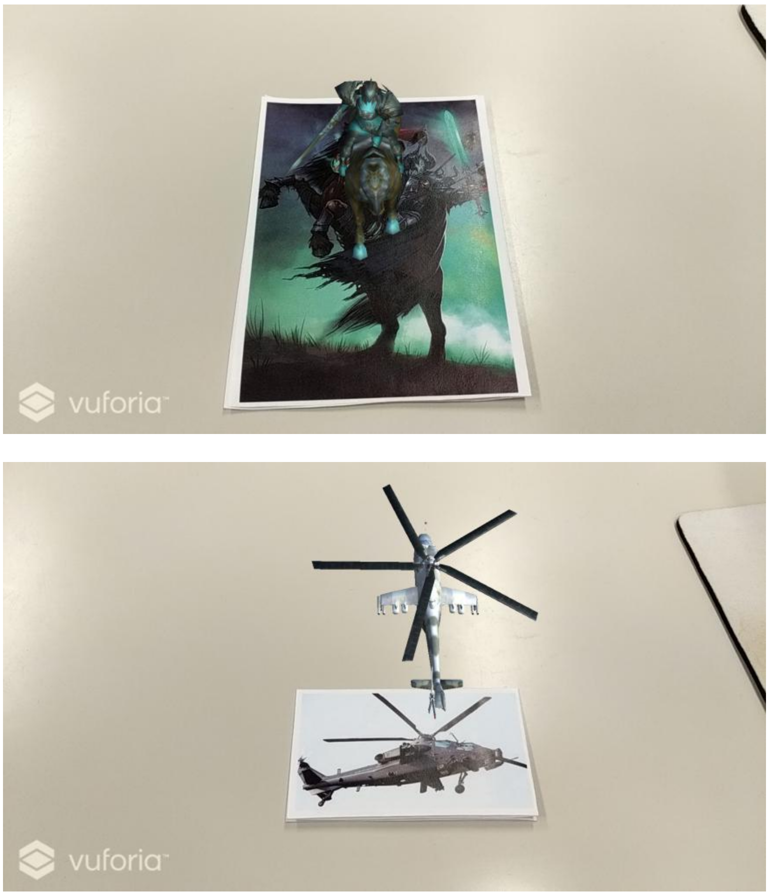
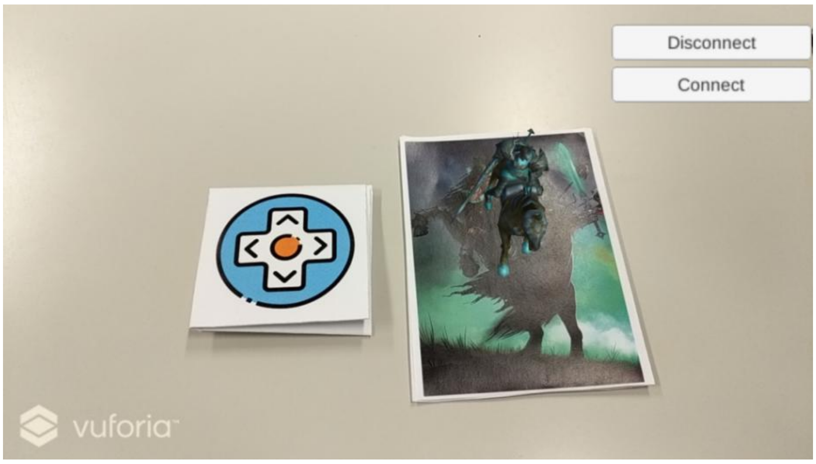
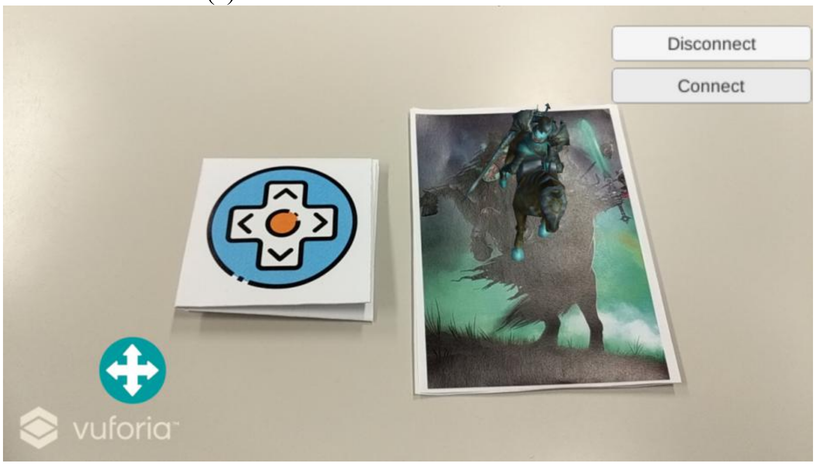


Fig. 5.3 Single object above its marker: knight and helicopter

The figure 5.3 shows two main objects. The single object marker shows a 3D virtual model of death knight or helicopter above the marker, which is waiting for connected with other additional markers. In this stage system just shows the original 3D model, there will be no interface for user to control the model, it needs extra controller. The background is the scene of camera.



(a) Connect or disconnect button



(b) Connect general joystick

Fig. 5.4 Controller interface of Arrow-Key I

(c) Move knight by joystick

Fig. 5.5 Controller interface of Arrow-Key II

The Fig.5.4 shows the main interface of controller part of this system, it's an example of arrow-key connected to virtual knight. Put and scan arrow-key marker, system will recognize it as a controller marker, and generate two buttons.

There are two kinds of icons for controller in this interface after scanning:

• Buttons for connection options: "Connect" and "Disconnect" buttons appear in the upper right corner of the AR device's screen, like figure5.4 (a) shows. User can press "Connect" button to generate a general joystick and connect it to the available main object, like figure5.4 (b) shows.

• Joystick: generated joystick can be used to control the horizontal movement of main object, as figure5.5 shows, user can touch screen to use joystick for controlling knight to move horizontally.

Before changing object, user needs to press "Disconnect" button to release the used general controller.

(a) Change object and add controller

(b) Horizontal and vertical movement

Fig. 5.6 Controller interface of Arrow-Key and Up-Down Key

System supports to change the main object which can be controlled. The figure5.6 shows that user change the object marker from knight to helicopter, and connect Arrow-Key to it. For helicopter, only horizontal movement is not enough, user can add an extra Up-Down Key to get a general up-down controller for vertical movement of flying objects like helicopter.

5.3 Module Part

Module part is designed to make interaction between user and AR system more attractive and interesting, it can be combined with controller part to improve experience.

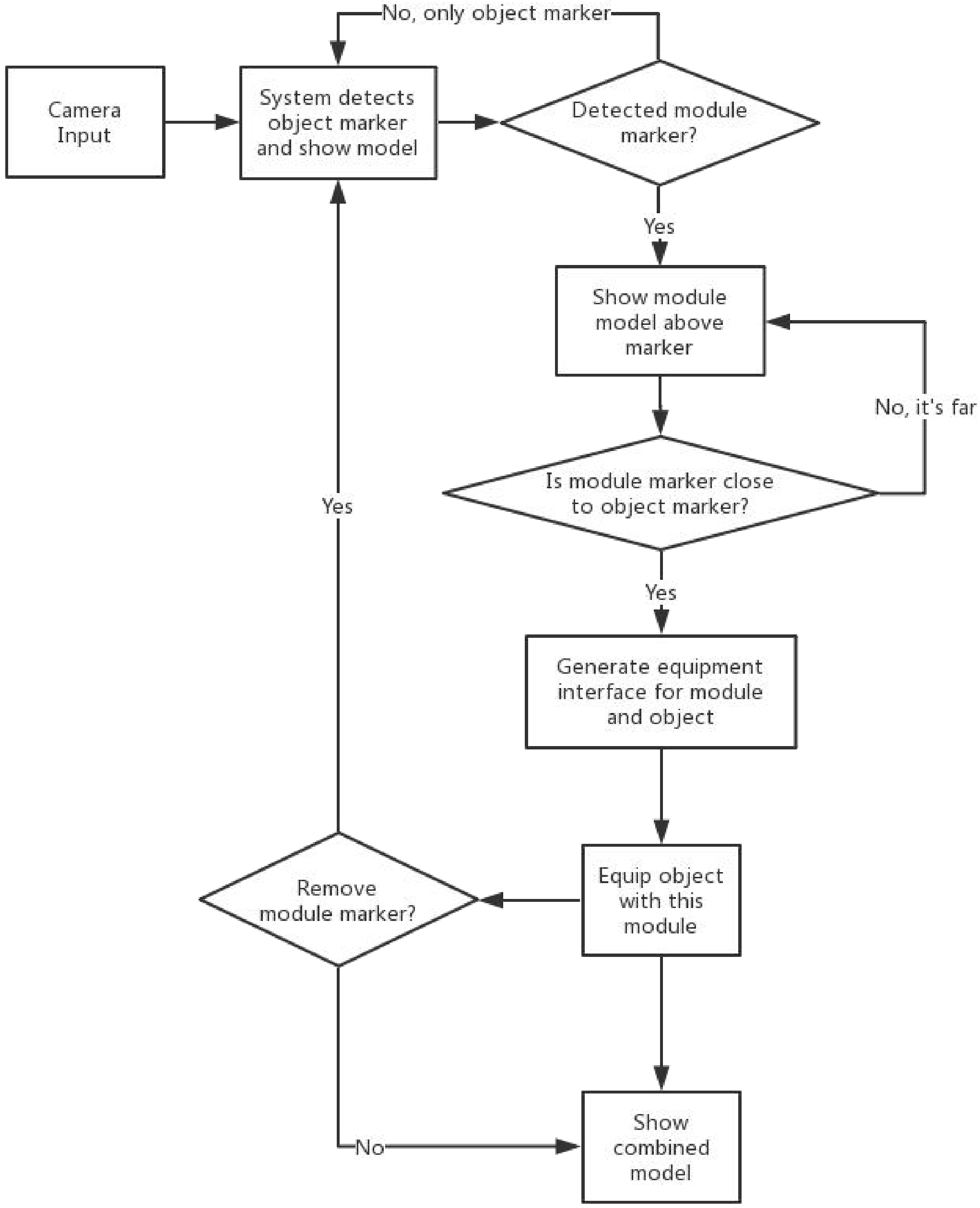


Fig. 5.7 Structure of module part

As figure5.7 shows, module part of this system will support user to choose some modules equipped to main object. Before really using this system to edit model by changing markers in real time, designer or user needs to design the equipment methods in advance, because of the 3D model conflict. Only available modules can be attached to a main object. These methods and related modules should be stored in database, and system will check them while judging whether generate equipment interface or not.

5.3.1 Usage Scenario

We also designed a simple usage scenario to show how module part is working in this system. Similar to controller part, system needs some designed markers for virtual objects and modules stored in the database.

(a) Robot (b) Backpack (c) Rifle

Fig. 5.8 Designed object and modules markers with their feature points

User uses this system to view 3D virtual model of robot, and edit it by putting and connecting available modules, and the change of model will be shown in real time. User needs to get 3 designed markers shown in figure5.8. System can scan (a)Robot marker to generate a virtual robot which can be attached some modules, and scan (b)Backpack marker and (c)Rifle marker to get a virtual backpack and a virtual rifle module which are available for this robot. After scanning objects and modules, user can view a interface generated by system to choose which module should be connected to the main object. In this scenario, it will be shown as robot and its optional equipment.

5.3.2 Interaction Design

Different with direct UI on the screen in the controller part, in order to get more attractive and intuitive experience when user are using these markers, in the module part of system, the optional buttons and some feedback are mainly shown around the virtual objects and modules in AR. we will introduce the interaction design in module part of system. It includes how to use it and why we design it in this way.

Fig. 5.9 Single object above its marker: robot, backpack and rifle

The figure 5.9 shows three single virtual contents, a robot, a backpack and a rifle, but there is a relationship among these three ones. The backpack and the rifle are modules which can be equipped to the virtual robot. Before user connect modules to objects, their original contents are independent. The background is the scene of camera.

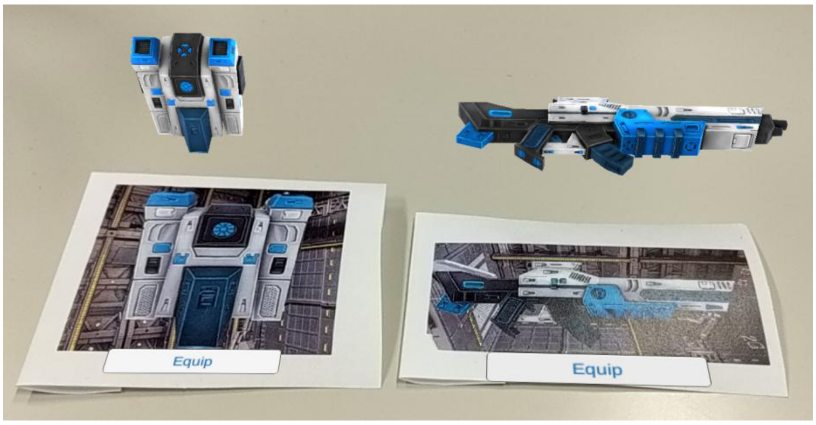


Fig. 5.10 Virtual buttons around modules

If system recognizes modules which are close to the main objects are available for the selected main object, system will generate a virtual button around the image, which can be switched between "Equip" and "Unequip", which is like what shown in the figure5.10.

(a) Robot equipped with backpack (b) Interface for equipped backpack

Fig. 5.11 Switched buttons and green tick after equipping

Like what shown in figure5.11 (a), when user presses “Equip” button above the backpack marker, system will equip main object robot with the module backpack, just like figure5.11 (b), then system will witch button to “Unequip” and show a green tick for equipped module. In the module part, the equipping method should be designed in advance for solving conflict among different 3D models, so in this case, we cannot make a virtual knight holding a rifle.

Fig. 5.12 Robot equipped with multiple modules

Our system also supports main objects to equip with multiple modules, as figure5.12 shows, a virtual robot can equip backpack and rifle at the same time. Of course system can support more than two modules if user or system developer designed more modules in advance. Every equipped module will be shown with switched buttons and green ticks, and the information of equipment will be shown in the upper left corner of screen to tell the user what modules he has chosen to equip. These interfaces can coexist with the interfaces generated by controller markers. It will provide a basis to combine two parts for some applications.

Fig. 5.13 Remove markers after equipping

Usually AR device cannot always aim at the something for a long time, and the range and accuracy of scanning will be influenced by the distance between camera and markers. When putting several markers which are hard to be captured by a single camera, it may also reduce the user experience while using many markers.

So the system needs to keep the connection status of main object after removing the markers out of screen. As figure5.13 shows, we removed backpack marker and rifle marker after equipping them. The models of modules are still shown on the robot model, and the information of equipment is also still displayed in the upper left corner of screen.

Chapter 6

System Implementation

6.1 Hardware and Programming Environment

We used some hardware and programming tools to create the system.

• Hardware Device for using system: Mobile with camera.

• Programming Environment: Unity 3D 2018.4.8f1, Pycharm 2018.

• Android SDK with Java for front-end building on smartphone.

• Vuforia for AR-marker detection and AR contents display.

We established this system and build the 3D models we need in Unity 3D 2018.4.8f1, and used C SHARP as the development language. And for realizing system, the Android platform should be Android 7.0 or higher, so we used a Google Pixel 2 like figture6.1 as the device, and made some front-end building for the system. In order to recognize the image or marker and add virtual contents on it, we used Vuforia-Unity SDK, and use a PC like table6.1 to support programming.

The information of device which we used are shown in following table and figures:

Operation System	Microsoft Windows 10
CPU	Intel(R) Core(TM) i7-7700HQ @2.80GHz
Graphics Card	NVIDIA Geforce GTX1060
Ram	8 GB
Software	Unity 2018.4.8f1

Table 6.1 Information of PC

Fig. 6.1 Google Pixel 2

Also we need a USB type-c cable to connect our phone to our computer.We use the Unity 2018.4.8f1 or later with Android Build Support selected during installation. The Android SDK 7.0 (API Level 24) or later is needed to be installed using the SDK Manager in Android Studio.

After the preparation, we need to create a new project and import the SDK:

• 1.Open Unity and create a new 3D project.

• 2.Unity 2017.2 or later:

1) File > Build Setting > Platform > Android.

2) Player Settings > XR Settings > Vuforia Augmented Reality.

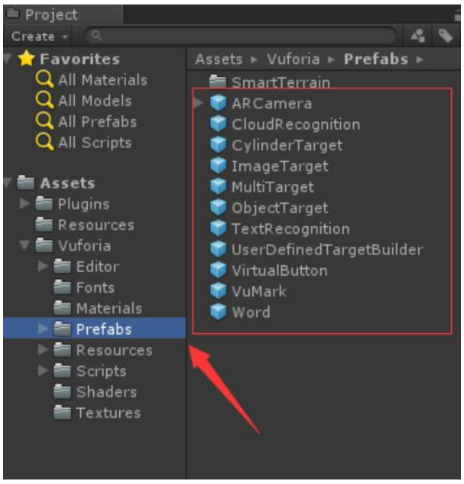
• 3. Import the Vuforia SDK for Unity before 2017.2:

1) Select Assets > Import Package > Custom Package

2) Select the vuforia-unity-6-2-10.unitypackage that you downloaded.

3) In the Importing Package dialog, make sure that all package options are selected and click Import.

(a) Unity 2017.2 or later



(b) Common materials under Vuforia > Prefabs

Fig. 6.2 Vuforia SDK import

6.2 System Diagram

The figure6.3 shows the system diagram. It shows how the framework of this system looks like.

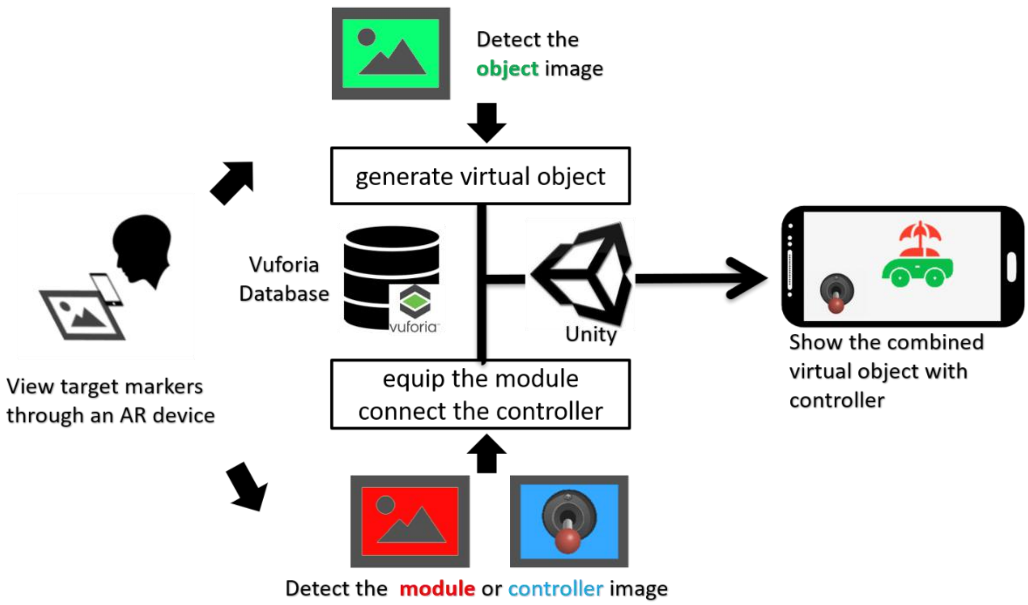


Fig. 6.3 System Diagram

In this research, the system has two main parts, one is controller connection and the other one is module connection. At first user scan target markers via some AR devices, which can be smartphone, PC with webcam and AR glasses. System will start to work after getting information of markers correctly, it will recognize the marker and query Vuforia database to confirm what contents the marker are linking with. System will show the virtual content for each marker and generate interfaces for additional parts, support user to connect modules or controllers with objects. The result of combination will be shown in Augmented Reality in real time, when user interacts with objects through general interfaces or physical markers, the system will update the data in the database.

6.3 3D virtual content modeling

In the demo, system will display these 3D virtual contents to the users. So we need to create those fictitious 3D virtual content in advance. We use Unity 3D to accomplish this part of work.

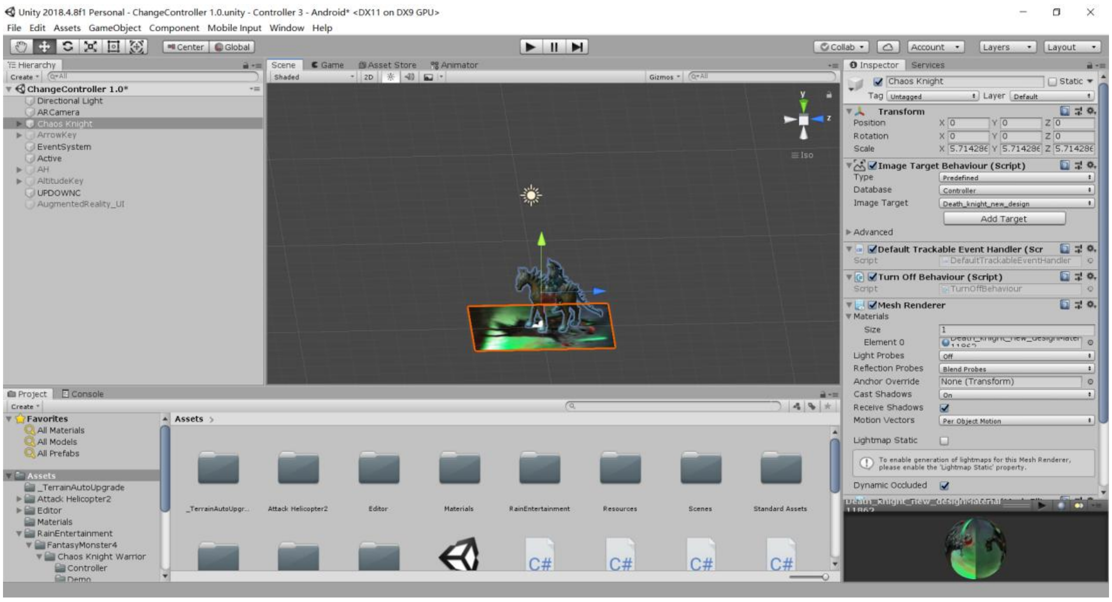
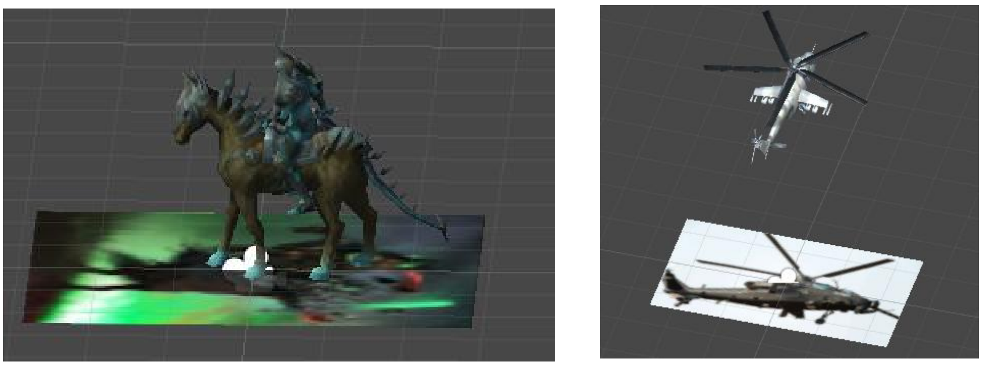


Fig. 6.4 Unity 3D

We have created markers to show the 3D virtual model (Available for free in Unity Assets Store). The 3D models shown in figure6.5 are the objects and modules in our system, which can be connected each other and controllers.



(a) Death Knight model (b) Helicopter model

(c) Robot model (d) Rifle model

(e) Backpack model

Fig. 6.5 3D models of objects and modules in the system

The five models above are linked with specific markers, and they are shown as common virtual content when user didn’t choose to connect them.

6.4 AR marker Recognition

Based on Vuforia service, our system can preset and and identify AR markers in the user’s view through AR devices or PC with webcam. We apply for a license of Vuforia service on its official website, then we can pick the designed pictures as AR markers to upload to Image Target database. Vuforia will help us to recognize those AR-markers in the user’s view.(Like what shown in Fig.6.6 and Fig.6.7).

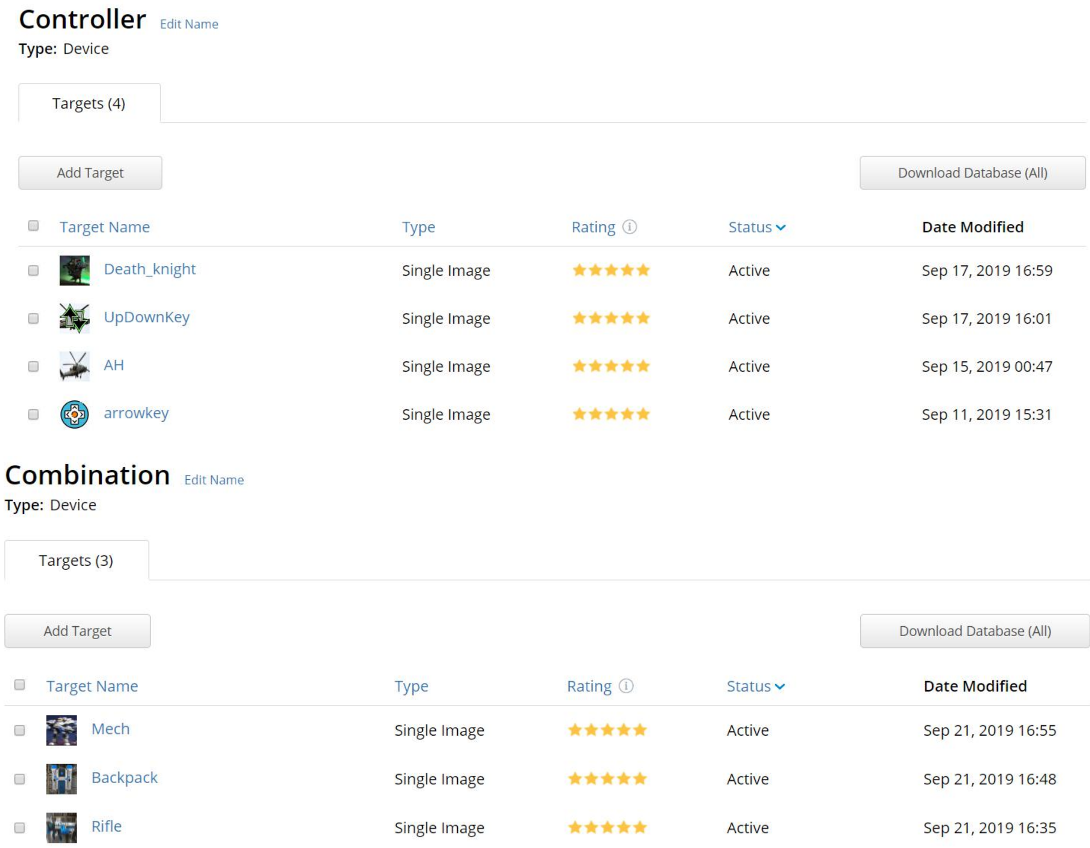


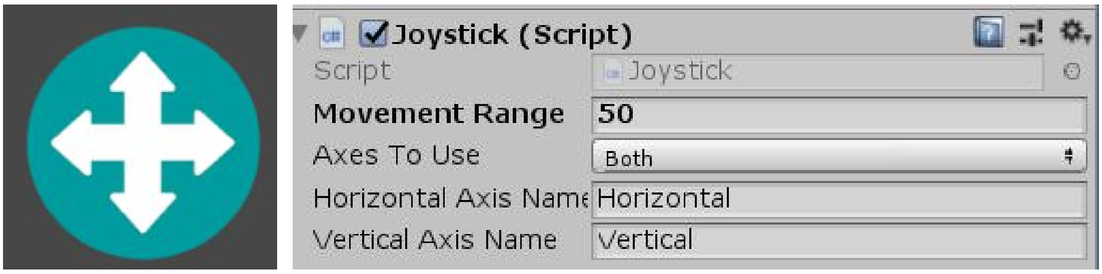
Fig. 6.6 Vuforia Image Target Database

Fig. 6.7 Image Target setting in Unity 3D

After uploading designed pictures and packaging the Image Target database into the Unity 3D, and choosing 3D model bonding to related marker, we can use these markers to show AR contents in our project.

6.5 General Controller Implementation

For the movement of object, we need to design a general controller, we design a joystick like Fig6.8 (a), and create a script for it with some general settings, after using the namespace "UnityStandardAssets", we need system to arrange and read the coordinates of virtual objects, as figure6.8 shows, we need to create virtual axes, it's a core part of general controller to which can be connected to some 3D models in this AR system.



(a) Joystick (b) Script setting for virtual joystick

Fig. 6.8 Image and script for general horizontal controller(Joystick)

```
void CreateVirtualAxes()
{
    // set axes to use
    m_UseX = (axesToUse == AxisOption.Both || axesToUse == AxisOption.OnlyHorizontal);
    m_UseY = (axesToUse == AxisOption.Both || axesToUse == AxisOption.OnlyVertical);

    // create new axes based on axes to use
    if (m_UseX)
    {
        m_HorizontalVirtualAxis = new CrossPlatformInputManager.VirtualAxis(horizontalAxisName);
        CrossPlatformInputManager.RegisterVirtualAxis(m_HorizontalVirtualAxis);
    }
    if (m_UseY)
    {
        m_VerticalVirtualAxis = new CrossPlatformInputManager.VirtualAxis(verticalAxisName);
        CrossPlatformInputManager.RegisterVirtualAxis(m_VerticalVirtualAxis);
    }
}
```

Fig. 6.9 Create virtual axes for general controller

As figure6.10 shows, we get the axis from virtual joystick on the screen of AR device, and use vector3(x, 0.0f, y) to change horizontal movement, and set the velocity based on Rigidbody component. System will check the change of joystick, if it's touched and moving, the change of coordinate will be sent to the main object, and refresh the coordinates of the main object per frame, it can be shown like the virtual object moving smoothly. And depending on the state of joystick input, object will do different animation.

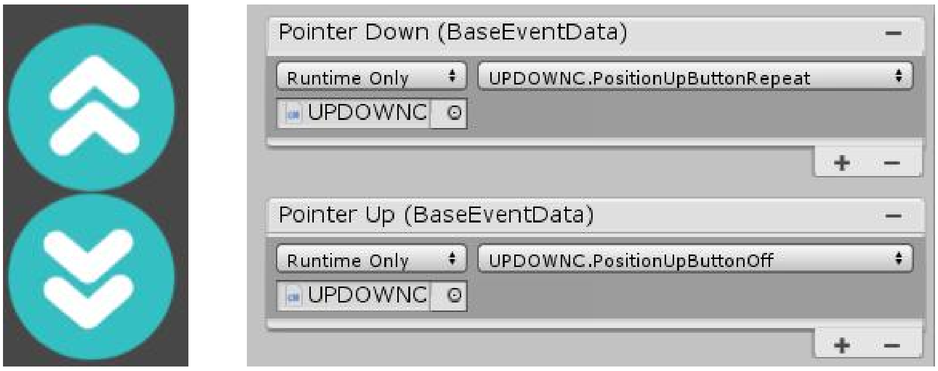
```
void Update()
{
  float x = CrossPlatformInputManager.GetAxis("Horizontal");
  float y = CrossPlatformInputManager.GetAxis("Vertical");

  Vector3 movement = new Vector3(x,0.0f,y);
  rb.velocity = movement * 4f;

  if (x!=0 && y != 0)
  {
    transform.eulerAngles = new Vector3(transform.eulerAngles.x, Mathf.Atan2(x, y) * Mathf.Rad2Deg, transform.eulerAngles.z);
  }
  if (x != 0 || y != 0)
  {
    anim.Play("Walk");
  }
  else {
    anim.Play("Stand");
```

Fig. 6.10 Parameter and animation of movement by general horizontal joystick

For something needs vertical movement, we design an logic to control the model's to only move in vertical direction, and set the parameter similar to horizontal controller, when user press the Up-Down button, system will keep the movement until releasing, which can make the movement more smooth.



(a) Up-Down key (b) Event trigger script for vertical movement

Fig. 6.11 Image and script for general vertical controller(Up-Down key)

```
public float translationSpeed = 5.0f;


bool repeatPositionUp = false;
bool repeatPositionDown = false;

public void PositionUpButton()
{
    GameObject.FindWithTag("ALT").transform.Translate(0, translationSpeed * Time.deltaTime, 0);
}

public void PositionDownButton()
{

    GameObject.FindWithTag("ALT").transform.Translate(0, -translationSpeed * Time.deltaTime, 0);
}
public void PositionDownButtonRepeat()
{
    repeatPositionDown = true;
}
public void PositionUpButtonRepeat()
{
    repeatPositionUp = true;
}
public void PositionUpButtonOff()
{
    repeatPositionUp = false;
    Debug.Log("Off");
}
public void PositionDownButtonOff()
{
    repeatPositionDown = false;
    Debug.Log("Off");
}
```

Fig. 6.12 Extra limited parameter of general vertical controller I

```
// Update is called once per frame
void Update()
{
  if (repeatPositionUp)
  {
    PositionUpButton();
  }

  if (repeatPositionDown)
  {
    PositionDownButton();
  }
```

Fig. 6.13 Extra limited parameter of general vertical controller II

In addition, we need an extra method to distinguish whether the model can be moved vertically or not, because not all objects should have a chance to move with change of altitude. We add the “ALT” tag to the all objects which can move vertically in advance in Unity, and before controlling the current main object, system will query the tag information, and only do the operation on the ones with “ALT” tag, which is shown as “GameObject.FindWithTag(“ALT”)” in figure6.12.

Something like death knight model without “ALT” tag in our user scenario, will not be permitted to connect this Up-Down controller, and something like helicopter model will be able to do that.

• translationSpeed = 5.0f : the speed of movement is set to 5.0 by temporarily, which can be changed by user or system developer if they want.

• translationform.Translate : decide the parameter of movement only in vertical direction, the coordinate change is 5.0 * duration of pressing button.

PositionDownButtonRepeat and PositionUpButtonRepeat will repeat the movement automatically when user keep pressing button. System will do the update once per frame.

6.6 Module Equipment Implementation

This section will show the final effect of the system. Before really using this function, designer should design the module equipment methods in advance for decreasing conflicts among different 3D models.

(a) Raw robot

(b) Robot holding rifle

(c) Robot equipping backpack

(d) Full armed robot

Fig. 6.14 Pre-designed equipment methods among 3 models

In Unity we decided how the modules will be equipped to the main object like above figure6.14, when user scan related marker and choose to equip module, the system will automatically add the module model to the designed position and combine the new module and main object into a new whole.

In the module part, user should be provided a chance to change the model with or without modules freely, which can be switched between "Equip" and "Unequip", as the figure6.15 shows, it's the example code of UI and module switch for rifle module. When user is handling the equipment of rifle, this system will divide the situation into two cases, the case1 is raw robot, the case2 is robot holding rifle.

```
// public method to switch avatars by pressing UI button
public void SwitchAvatar()
{

  // processing whichAvatarIsOn variable
  switch (whichAvatarIsOn)
  {

    // if the first avatar is on
    case 1:

      // then the second avatar is on now
      whichAvatarIsOn = 2;

      // disable the first one and enable the second one
      rifle.SetActive(true);
      Tick.SetActive(true);
      equipbutton.SetActive(false);
      unequipbutton.SetActive(true);
      anim.Play("Shoot_single");
      break;

    // if the second avatar is on
    case 2:

      // then the first avatar is on now
      whichAvatarIsOn = 1;

      // disable the second one and enable the first one
      rifle.SetActive(false);
      Tick.SetActive(false);
      equipbutton.SetActive(true);
      unequipbutton.SetActive(false);
      break;
  }

}
```

Fig. 6.15 Rifle example code of UI and module switch

When in case1, system can show “Equip” button to make rifle module active, show a green tick, and make it clear to show the animation “Shoot_single”.

When in case2, system will show “Unequip” button to make rifle module inactive, hide the green tick, and cancel the animation about shooting, turn it to standing.

Chapter 7

Preliminary Evaluation

7.1 Experiments

7.1.1 Participants

Invite 10 participants to use our system, ranging in age from 22 to 25 and including 2 female and 8 male. All participants have basic computer and smartphone skills, 8 of them have experience or knowledge about marker-based AR system.

7.1.2 Method

All participants are given a brief introduction of this system. Before each study, we introduced basic operation methods of system and the difference with normal marker-based AR system. Then each participant needs to use our system with some designed controller and module markers.

After that, the participants will be asked to fill in a questionnaire. The questionnaire has following 5 questions and these questions use the 5-point Likert scale.

- The system is easy to use.
- The cooperation methods among multiple markers is interesting or useful.
- I can easily understand meaning and usage of each marker and its content.
- I can feel free to combine manipulation methods and edit models.
- The system can provide an attractive user experience.

7.1.3 Questionnaire

The questionnaire is showed in Figure7.1. We plan to investigate the basic information of each participant and get their feedback.

QUESTIONNAIRE

Name: Age: Gender: F / M Date

Questions:

The questions are based on 5-point scale.
Answer the following questions by circling the most appropriate answer.

1. **The system is easy to use.**
   Strongly Disagree Disagree Neutral Agree Strongly Agree
2. **The cooperation methods among multiple markers is interesting or useful.**
   Strongly Disagree Disagree Neutral Agree Strongly Agree
3. **I can easily understand meaning and usage of each marker and its content.**
   Strongly Disagree Disagree Neutral Agree Strongly Agree
4. **I can feel free to combine manipulation methods and edit models.**
   Strongly Disagree Disagree Neutral Agree Strongly Agree
5. **The system can provide an attractive user experience.**
   Strongly Disagree Disagree Neutral Agree Strongly Agree

How could the system be improved?

____

Fig. 7.1 Questionnaire

7.2 Results

After collecting the results given by the participants, the evaluation of using the system to enhance cooperation among multiple markers and improve user experience can be carried out. All the participants are asked to rate on a Likert Scale ranging from 1 to 5.

Question	1	2	3	4	5
Q1: The system is easy to use.			2	7	1
Q2: The cooperation methods among multiple markers is interesting or useful.			1	4	5
Q3: I can easily understand meaning and usage of each marker and its content.		1	1	7	1
Q4: I can feel free to combine manipulation methods and edit models.		1	2	6	1
Q5: The system can provide an attractive user experience.				5	5

Table 7.1 Answers Statistics of Investigative Questions

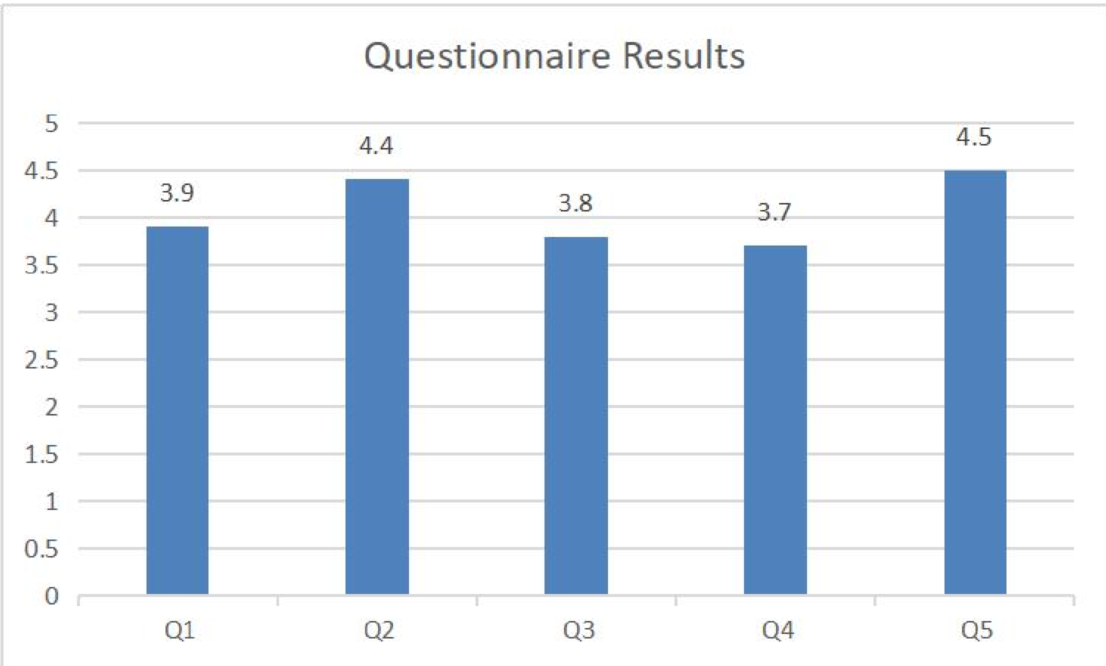


Fig. 7.2 Questionnaire Results

The Table 7.1 and Figure7.2 show the results of questionnaire.

The average score of Q1 is 3.9. Q2 is 4.4 and Q3 is 3.8. The three questions are used to judge performance of system itself and the interface UI design. From the results we can find that most of participants think the system is easy to use, users can easily understand meaning and usage of marker while looking at the physical image, and interface way actually makes them feel interesting and useful. But there are still few participants who think the system is a little complex.

The average score of Q4 is 3.7. We can get that 7 of 10 participants think the existing cooperation methods in this system is free and good , three participants consider it is still somewhat inadequate.

The average score of Q5 is 4.5. The high score shows that the system is attractive enough for them and the participants agree that this system can effectively improve the user experience.

Overall, we got a positive feedback through the preliminary user study. We also get some comments and suggestions from participants:

1. “The cooperation methods are not enough, if add more kinds of markers and cooperation methods, this could be mush more fantastic.”
2. “The marker can be designed to be more intuitive, maybe with some short text description.”
3. “Smartphone is still limited, being improved in more powerful AR platform should find more possibilities.”
4. “I hope for a muti-player battle game based on this system!”

Chapter 8

Conclusion

8.1 Summary

In general, this paper expounded problems of the traditional marker-based AR system, for that we proposed a system that can provide a general platform to support cooperation among multiple markers and their contents. It allows user to design and enjoy the stimulating virtual contents by switching between real world and augmented world, which is useful to improve the user experience.

In order to enhance the interaction among multiple markers and user, two main components were designed to build this proposed system.

For the controller part, the system can detect and distinguish object marker and controller marker by querying the database, then automatically generate interface for user to choose whether to generate some additional controllers from controller markers for controlling main objects or not. It's possible for users to change main objects and design specific control methods as they want, which can make interaction among user and multiple markers more abundant and attractive.

For the combination part, the system is also designed to distinguish object marker and additional module marker based on database. The additional modules will be generated from module markers which can be equipped into main objects. This part provides user a function to choose and edit a main virtual object model by putting various module markers and pressing virtual buttons rather than changing object marker or doing extra manipulation in software.

Several experiments were performed to verify whether our system is effective or not in terms of improving cooperation among multiple markers and enhancing user experience. And the feedback is positive.

8.2 Future work

Although we have proposed a prototype system which can support cooperation among multiple markers, such as additional controller and module function, there are still some limitations and future possibilities to develop more functions for a better user experience. For the rapid development of AR technology and user requirement, future system should support to distinguish abundant and complex image input and provide more interaction methods, such as, supporting complex combination of controllers and modules with multiple users and different devices at the same time.

Reference

[1] Ronald Azuma. A survey of augmented reality. Presence: *Teleoperators & Virtual Environments*, 6(4):355-385, 1997.

[2] Dimitris Chatzopoulos, Carlos Bermejo, Zhanpeng Huang, and Pan Hui. Mobile augmented reality survey: From where we are to where we go. *IEEE Access*, 5:6917-6950, 2017.

[3] Shriram K. Vasudevan, T. Naveen, K.V. Padminy, J. Shruthi Krithika, and P. Geethan. Marker-based augmented reality interface with gesture interaction to access remote file system. *International Journal of Advanced Intelligence Paradigms (IJAIP)*, 10(3), 2018.

[4] Yuna Kang and Soonhung Han. An alternative method for smartphone input using AR markers, *Journal of Computational Design and Engineering*, 1(3):153-160, 2014.

[5] Sanni Siltanen. Theory and applications of marker-based augmented reality. *VTT Technical Research Centre of Finland*, 2012.

[6] Charles Woodward, Janne Lahti, Jukka Rönkkö, Petri Honkamaa, Mika Hakkarainen, Jani Jäppinen, Kari Rainio, Sanni Siltanen, and Jouko Hyväkkä. Virtual and augmented reality in the Digital building project. *International Journal of Design Sciences and Technology* , 14(1): 23-40, 2007.

[7] Sanni Siltanen. Texture generation over the marker area. *IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR)*. 22–25 October 2006. Page 253.

[8] Mark Billinghurst, Adrian Clark, and Gun Lee. A survey of augmented reality. *Foundations and Trends® in Human–Computer Interaction*, 8(2-3):73–272, 2015.

[9] Patrick Schueffel. The concise fintech compendium. *Fribourg: School of Management Fribourg/Switzerland*, 2017.

[10] Rachel Metz. Augmented reality is finally getting real technology review. *Internet WWW-page, URL: http://www.technologyreview.com/news/428654/augmentedreality-is-finally-getting-real/(10.07. 2013)*, 2012.

[11] Jannick P. Rolland, Yohan Baillot, and Alexei A. Goon. A survey of tracking technology for virtual environments. *Fundamentals of wearable computers and augumented reality*, page 67, 2001.

[12] Mark Billinghurst and Hirokazu Kato. Marker tracking and hmd calibration for a videobased augmented reality conferencing system. *Proceedings 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR'99), 20-21 Oct.* IEEE, 1999.

[13] Marco Paladini. 3 different types of ar explained: Marker-based, markerless and location. *Internet WWW-page, URL: https://www.blippar.com/blog/2018/08/14/ marker-based-markerless-or-location-based-ar-different-types-of-ar*, 2018.

[14] Karan Kalra, Anuroop Katiyar, and Chetan Garg. Augmented reality and improved ransac. pages 55-62. IJETSR, 1998.

[15] Amon Rapp. Designing interactive systems through a game lens: An ethnographic approach. *Computers in Human Behavior*, 71:455 – 468, 2017.

[16] Michael Sailer, Jan Ulrich Hense, Sarah Katharina Mayr, and Heinz Mandl. How gamification motivates: An experimental study of the effects of specific game design elements on psychological need satisfaction". *Computers in Human Behavior*, 69:371 – 380, 2017.

[17] Myoung-Hee Kim and Jiyoung Park. Smart pico-projected ar marker. In *SA'12 SIGGRAPH Asia 2012 Posters Article No. 19*. ACM, 2012.

[18] Khushal Khairnar, Kamleshwar Khairnar, Sanketkumar Mane, and Rahul Chaudhari. Furniture Layout Application Based on Marker Detection and Using Augmented Reality. *International Journal of Engineering Science and Computing*, pages 4201 - 4206, 2016.

[19] Angeline Lee Ling Sing, Awang Asri Awang Ibrahim, Ng Giap Weng, Muzaffar Hamzah, and Wong Chun Yung. Design and Development of Multimedia and Multi-Marker Detection Techniques in Interactive Augmented Reality Colouring Book. *International Conference on Computational Science and Technology*, pages 605-616, 2019.

[20] Martin White, Rafaá Wojciechowski, and Krzysztof Walczak. Building virtual and augmented reality museum exhibitions. In *Copyright © 2004 by the Association for Computing Machinery, Inc*. ACM, 2004.

[21] Yuan-Ping Luh, Jeng-Bang Wang, Jin-Wan Chang, Shun-Ya Chang, and Chih-Hsing Chu. Augmented reality-based design customization of footwear for children. In *24 February 2012 / Accepted: 3 April 2012 / Published online: 15 April 2012 © Springer Science+Business Media, LLC 2012*, pages 905-917. ACM, 2012.

[22] Toshikazu Ohshima, Kiyohide Satoh, Hiroyuki Yamamoto, and Hideyuki Tamura. AR2Hockey: A case study of collaborative augmented reality. *VRAIS '98: Proceedings of the Virtual Reality Annual International Symposium*, pages 268-275, 1998.

[23] Toshikazu Ohshima, Kiyohide Satoh, Hiroyuki Yamamoto, and Hideyuki Tamura. *RV-Border Guards: A Multi-player Mixed Reality Entertainment*. *Transactions of the Virtual Reality Society of Japan*, 4(4): 699-705, 1999.

[24] Maria Dahlquist, Simon Mann, Hannah Slay, Joanne Zucco, Smith-Ross, Bruce Thomas, Nicholas Cooper, and Aaron Keatley. Augmented reality chinese checkers. In *ACM SIGCHI International Conference on Advances in computer entertainment technology.* Pages 117–126, 2004.