Department of
Computer Science
香港城市大學
City University of Hong Kong

CS3483 Multimodal Interface Design

Group Project Final Report

Message Animation: a Facial Animation based Instant Messaging App

Tutorial (21:00-21:50 Monday) Group21

Wang Yucheng	**Yao Shenglong**	**Tam Kam Chuen**
56198686	**56198698**	**56204415**
Zhou Yukun	**CHAN Hiu Leong**	
56198649	**55820320**	

Table of Contents

1. Introduction

In this project, we develop an instant messaging software that can transfer users' vocal or textual messages into a cartoon animation.

Nowadays, people usually chat using pure text and voice on common messaging apps like WhatsApp, WeChat and Messenger. We observe that users will not have the expected response when receiving the textual or voice messages from others. Instead, they enjoy innovative communication that takes advantage of multi-modal technologies such as short videos and animation; iMessage💬iMessage is one of the representatives. The app tries to implement innovative communication methods such as creating a cartoon character to present itself. We observe that people tend to share and engage in more emotional communication, such as facial expressions instead of plain text, especially during the pandemic. While most of these apps do not have a function to express their emotion to others, we adopt a more straightforward way of expressing the feelings generated from the voice and texts using a cartoon character. The facial expression of the virtual character can be essential to improve the experience of interpersonal communication.

Based on it, our group came up with an idea to help senders express their emotions and facial details and help receivers have a more engaging experience when receiving the message. For now, people use emojis to express their feelings in the existing chatting application, which might slightly differ from the user's real feelings. The emoji cannot show the facial details, the opponent might know your feeling, but they can only guess the degree of emotion because the emoji is highly abstract and cannot show the details of facial gestures. By contrast, for our design, we aim to complete the nonverbal interpersonal communication, which contains facial expressions, tone, and voice speed. The cartoon character will speak out the words with an appropriate tone and facial expression. As this feature allows the user to make a personalized animated character to express their feelings, we believe the idea we propose will become an innovative and brand-new communicative element.

In this report, a detailed analysis of the user community will be discussed first. After that, the final product including the structure of interfaces, the layout of the important interfaces, detailed design justification with several rules will be discussed. The alternative modality of interaction and possible future extensions will be mentioned at last.

2. Analysis of the user community

A detailed analysis of the related user communities is necessary for valuing and investigating the idea we presented in this project. Before developing the software, we have to understand how the idea can bring about beneficial changes to specific target users. Specific target users should be identified and analyzed from the aspects including age and characteristics. **Message Animation** is an application used for communication, and primarily it uses animated characters to express.

Therefore, the major target users are the people using social applications frequently in daily life and who are willing to try the latest messaging application. Besides, the minor target users are those who suffered from Asperger syndrome since they could take the advantage of the animation. Details will be explained in this chapter.

2.1 Reason to use an animated character

Before talking about the statistics of the target user group, it should be clarified and illustrated why users will be attracted and willing to use animated characters or communication.

1. Importance of facial expression

People use verbal and written language to communicate with each other in daily life. According to Singh [1], only 7% of communication consists of words themselves in human communication, whereas 93% are body language and paralinguistic cues. Body language, including facial expression, plays a vital role to express human thought. However, behind current online communication, most body language cannot be expressed through the internet and screen. It can only be delivered by text or voice only.

Besides that, Eckhard Hess says the eyes may reveal and be accurate of all human communication signals because they are a focal point on the body [2]. Therefore, the eye, together with the face, is an important emotion indicator. If there is no facial expression, people may not be able to recognize the real meaning of the message.

2. Hit of the animated message

Apart from the importance of facial expression, animated messages have become more and more popular recently. According to a study by Robert Williams [4], it is observed that users are willing to use visual expression messages. We believe that animated characters as a communication tool will also become attractive to the users.

2.2 Major user groups

The idea of **Message Animation** we proposed is an instant message communication platform with a visual expression message. To identify the user groups, a survey on communication platforms of different ages is conducted.

According to the report from The London School of Economics and Political Science [3] ,Snapchat, one of the famous communication platforms for mainly visual expression messages, has the primary group of people below 25 ages, teenagers and young adults (figure 1). This age group is more interested in using new trends and new communication platforms, which will be more attracted to the visual message.

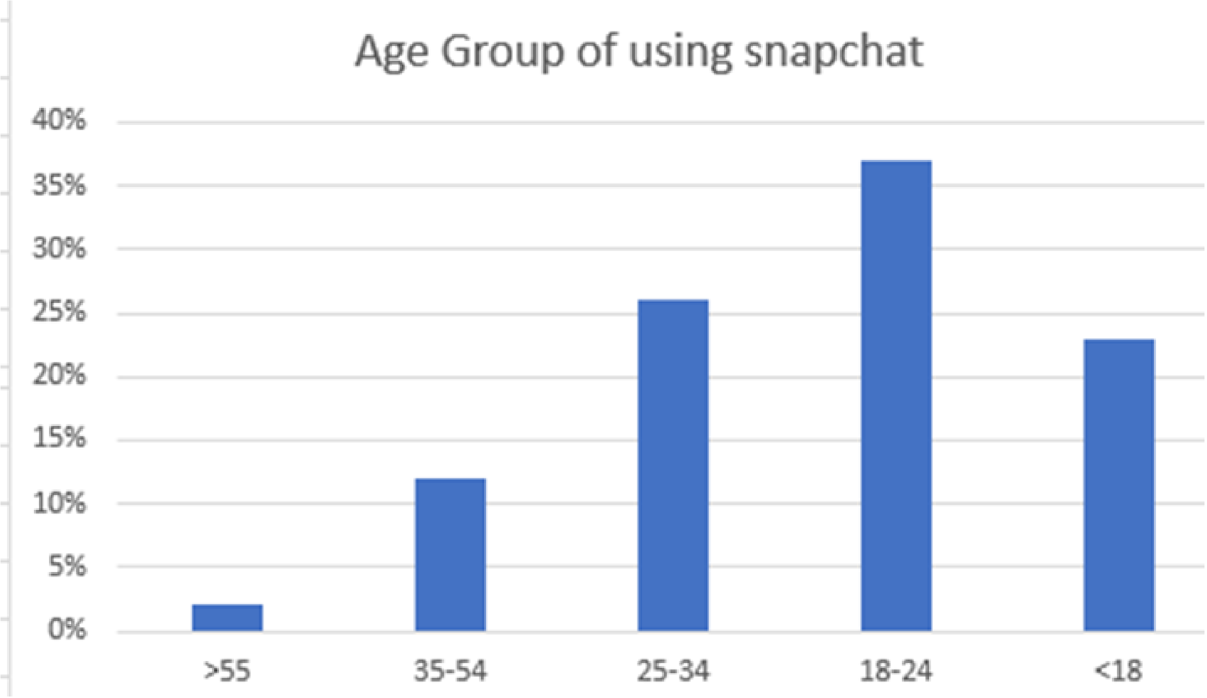


Figure 1 - Age Group of using snapchat

Besides, according to Tenor [4], a mobile platform for visual messages such as .gif, it shows that 77% of adults think visual expression improves their communication (figure 2). Tenor also points out that 65% of visual expressions in messaging express their personality or emotion (figure 3).

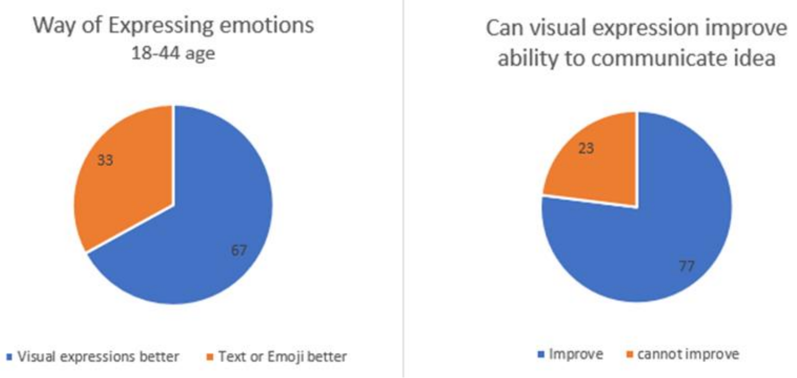


Figure 2 - Opinion on visual expression

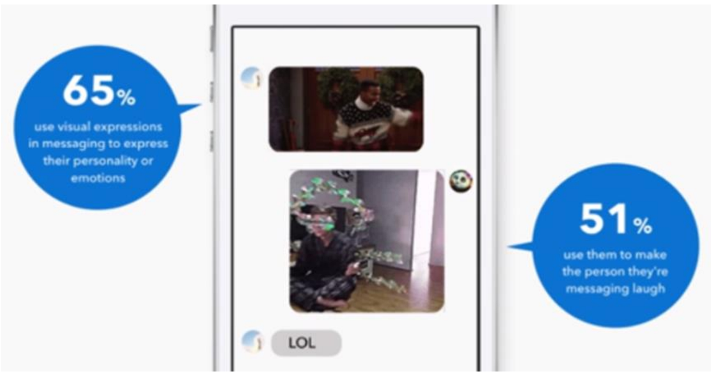

65% use visual expressions in messaging to express their personality or emotions
LOL
51% use them to make the person they're messaging laugh

Figure 3 - Data of visual expression usage

Based on these two points, we figure out that teenagers and young adults are more willing to use new platforms. There are the majority of people who will use visual expression to express their emotions and personality. We can conclude that the Message Animator would target smartphone users with ages between 15~29.

2.3 Design for Target User group

As said, teenagers and young adults from fifteen to late twenty are our target users. The design of software should meet their age-ground characteristics and requirements.

2.3.1 Innovative way of communication

After covid 19, online communication became more popular as most of the people cannot communicate face-to-face, especially in the young generation. Users cannot be satisfied with textual and emoji. This group of people enjoys visual messages such as .gif and some facial expression tools. Message Animation provides a platform with innovative communication technology which makes use of artificial intelligence. It allows an animated character to express their facial details, which is brand new and innovative.

2.3.2 High speed, high consistent and convenient

The major target users are the generation who are affected by fast-food culture. They may not have much leisure time to slow down and have been used to processing the information at high speed. They want a fast, consistent, and convenient platform to perform online communication. Therefore, Message Animation needs to be fast to access, especially for the transition of the animated character. Also, the feature should be consistent and convenient. Therefore, for the animated character, the scanning method should be accurate and stable. When sending it to the users, the processing time should not be too long.

2.4 Minor User Community

Apart from the major user group, the app also caters for universal usage since the application is a communication software. It is our responsibility to accommodate the needs of the minor user community, such as the people who suffer from neurodevelopmental diseases like Asperger syndrome.

Aspeger was estimated to affect 37.2 million people globally [5]. People who are diagnosed with it may be incapable of understanding the hidden message of non-verbal expressions, such as emotions, which makes it hard to share the same sentiment with people. By taking advantage of the features, they can practice a correct facial expression to comprehend the meaning of the speech and learn the correct way to express their feelings. By matching the text or voice combined with the text, they can practice if they are receiving and expressing the same feeling and meaning of text and speech.

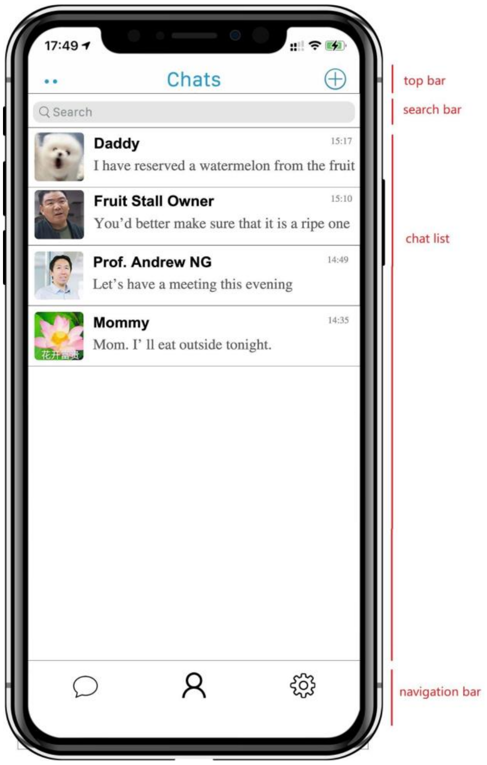
3. Interface relationships and associated features

3.1 Chat List Screens

3.1.1 Relationships

This chat list interface screen is the home page when the user enters the app. Users can access the dialog interface of each chat by tapping the corresponding chat and accessing the face capture interface through the figure setup option after clicking the add button..

3.1.2 Feature



17:49

Chats

Search

Daddy 15:17
I have reserved a watermelon from the fruit

Fruit Stall Owner 15:10
You'd better make sure that it is a ripe one

Prof. Andrew NG 14:49
Let's have a meeting this evening

Mommy 14:35
Mom. I' ll eat outside tonight.

Figure 4 - Chat list interface

Top

At the top of the chat list interface, there is an "edit" button on the left side and an "Add" button on the right side. The search section with a placeholder for inputting is located below the title bar. The search bar occupies 95% of the width of the screen.

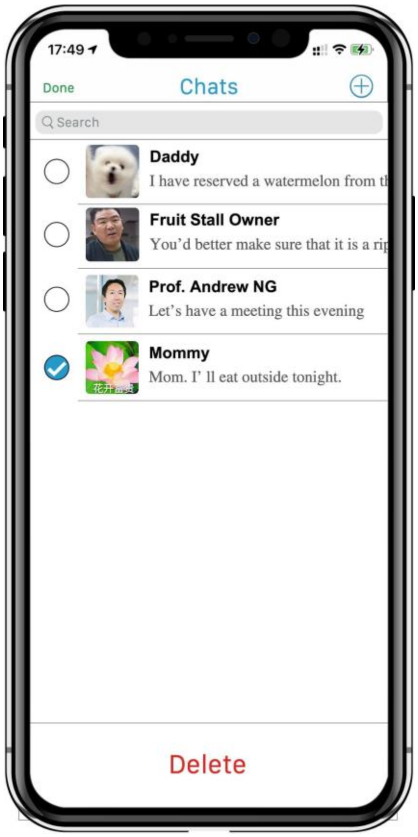
Center

For the central message area, we adopt the design of listing the chat tabs from top to down. Each tab consists of the following information, including the profile picture, user name, last message, and the last message's sending time. Each chat is a rectangular shape that occupies 100% width and 13% height of the screen. The profile picture is shown on the left of the rectangular box. It makes use of 15% screen width and 90% height of the rectangular shape.

The chat name is regarded as the friend's name set by the user, shown on top of the chat description. The description area makes up around 80% screen width. For the time, it is at the same level as the username and placed on the rightmost of the rectangular shape.

Bottom

As the main component of the bottom of the interface, the navigation bar comprises three well-known logos (💬 , 👤 and ⚙), which is chats, contacts and settings respectively. Each of them occupies around 33% screen width. The height of the navigation bar is approximately 13%.



17:49

Done Chats ⊕

Search

○ Daddy
I have reserved a watermelon from th

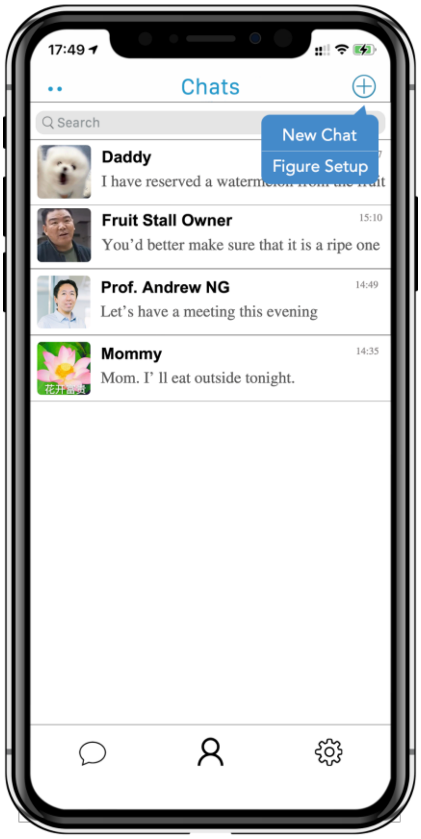
○ Fruit Stall Owner
You'd better make sure that it is a rip

○ Prof. Andrew NG
Let's have a meeting this evening

✓ Mommy
Mom. I' ll eat outside tonight.

Delete

Figure 5 - •• is clicked



17:49

•• Chats ⊕

Search

New Chat
Figure Setup

Daddy
I have reserved a watermelon from the fruit

Fruit Stall Owner 15:10
You'd better make sure that it is a ripe one

Prof. Andrew NG 14:49
Let's have a meeting this evening

Mommy 14:35
Mom. I' ll eat outside tonight.

💬 👤 ⚙

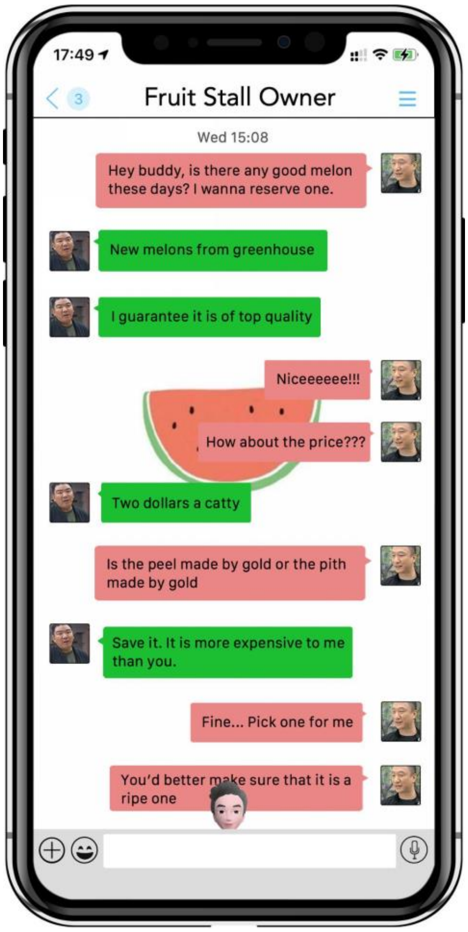
Figure 6 - ⊕ is clicked

We put two buttons there for the left and right of the top bar, including edit (••) and more option (⊕). Regarding the edit button, it provides a multiple delete chats function when it is clicked (figure 5). As for the more option button, it consists of two implementations of the requirements, figure setup and opening a new chat. A selection list would pop out when this button is clicked (figure 6).

3.2 Dialogue Interface Screen

3.2.1 Relationship

This dialog interface provides functions for users to switch to other interfaces and can be accessed from the chat list interface by tapping the corresponding chat and back to the chat list interface with a return button at the left side in the title bar.



Wed 15:08

Hey buddy, is there any good melon these days? I wanna reserve one.

New melons from greenhouse

I guarantee it is of top quality

Niceeeeee!!!

How about the price???

Two dollars a catty

Is the peel made by gold or the pith made by gold

Save it. It is more expensive to me than you.

Fine... Pick one for me

You'd better make sure that it is a ripe one

Figure 7 - Dialog interface

3.2.2 Features

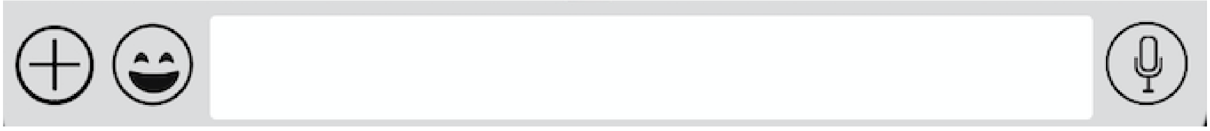
Title

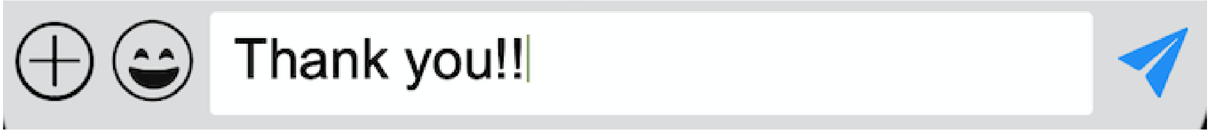
The title bar (figure 7) includes the name of the person you chat with, a return button in the top left corner and an options button to provide some necessary functions such as returning to the previous page and checking the detailed information about this dialog.

Dialog Display

The dialog display area (figure 7) takes the whole middle part as its content display area and organizes the messages in a good manner. The contact person and the user himself are separated to two sides of the area, corresponding to each message's origin. Each message is presented as a rectangular massage bubble originating from the corresponding user's head portrait. Moreover, this rectangular massage bubble can easily adjust its width to fit the message content and have different background colors for corresponding messages. We adopt 70% of the whole screen width as the maximum message bubble width.

Input





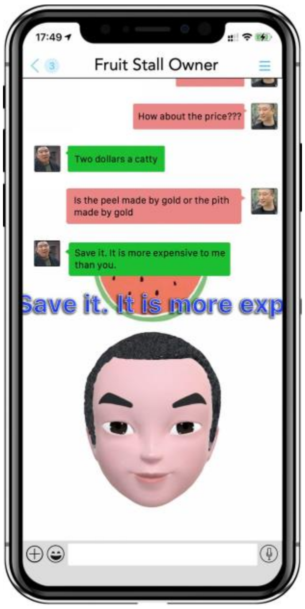
Thank you!!

Figure 8 - Text input bar

The input box is in the middle of the input area (figure 8) and the whole input area is located at the bottom of the screen as the base area (figure 7).

The microphone button is at the right side of the input box and the user can tap it to switch between text input mode and voice input mode. Moreover, after typing or speaking, the right-hand side icon will change to a highlighted send button for the user to confirm sending. There is an add button for other format input and an emoji button on the left side of the middle input.

<u>Floating Cartoon Widget</u>



How about the price???

Two dollars a catty

Is the peel made by gold or the pith made by gold

Save it. It is more expensive to me than you.

Save it. It is more exp

Figure 9 - An audio is played

The cartoon widget is the 3D head portrait character set by the user in the face capture interface.

The widget (figure 9) is floating at the upper limb of the input area. In the automatic default setting, when the user receives a message from the contact person, this cartoon character will automatically jack up the message to occupy the empty space at the bottom. It will express the message with the scrolling striking caption, the contact person's original voice, suitable emotional tone and facial expression extracted from the input message. What's more, the animation effect can also be woken up manually by double-tapping the corresponding message bubble, and if you tap on the widget itself, it will just spring up and stand at the place. Correspondingly, users can also swipe down to minimize the character whenever they want.

3.3 Face Generating Interface

3.3.1 Relationships

This face capture interface can be accessed from the figure setup option in the add menu. It can be accessed by clicking the add button in the chat list interface. The output 3D model will be used in the dialog interface.

3.3.2 Features

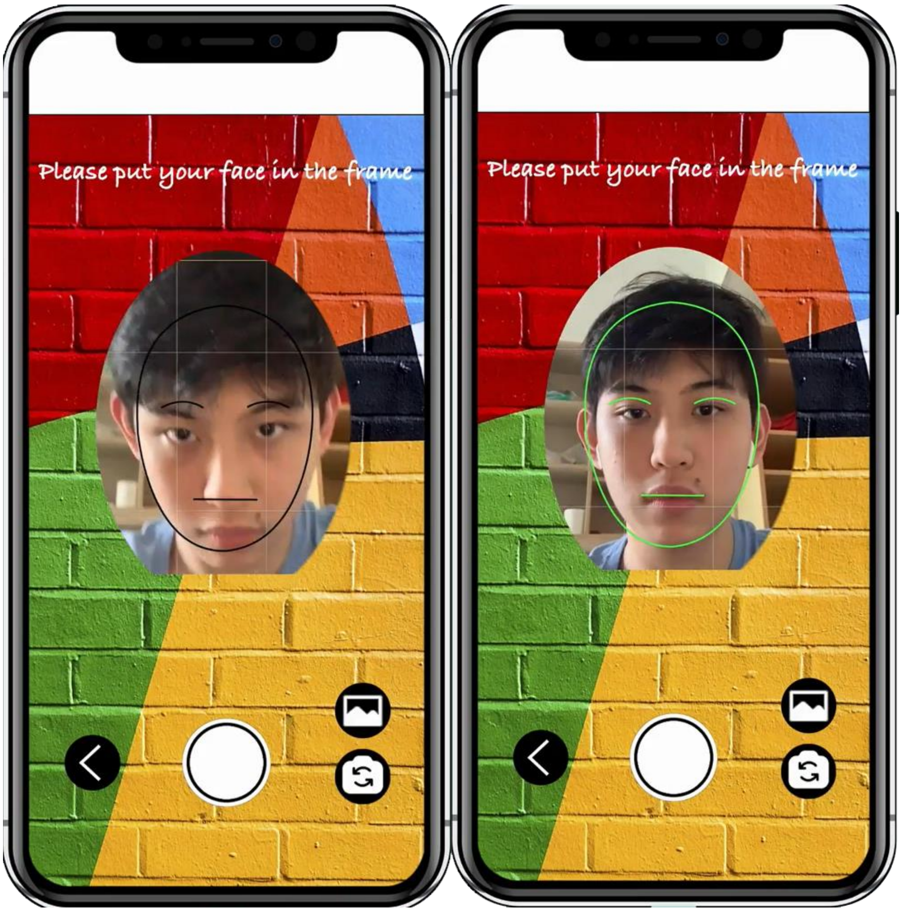

Please put your face in the frame
Please put your face in the frame

Figure 10 - Character generating interface (photo taking)

Capture

For the primary capture interface, most space is taken up to show the camera's view, which is the most significant thing when people open the camera and take a selfie. The black frame guides users to put their face within and makes sure the taken photo can cover the users' face clearly. A clear photo can make the model easier to generate the 3D model with a better effect. The frame will turn green if the face is detected at a proper position. The shutter button is in the middle of the bottom. Less essential widgets like the images gallery and the flip camera button are also in the bottom but in the right part, a little bit far from the middle. The former is for choosing the previously-stored picture, and the latter is for the front and back camera flipping.

The cancel button with black arrow is on the left bottom corner.

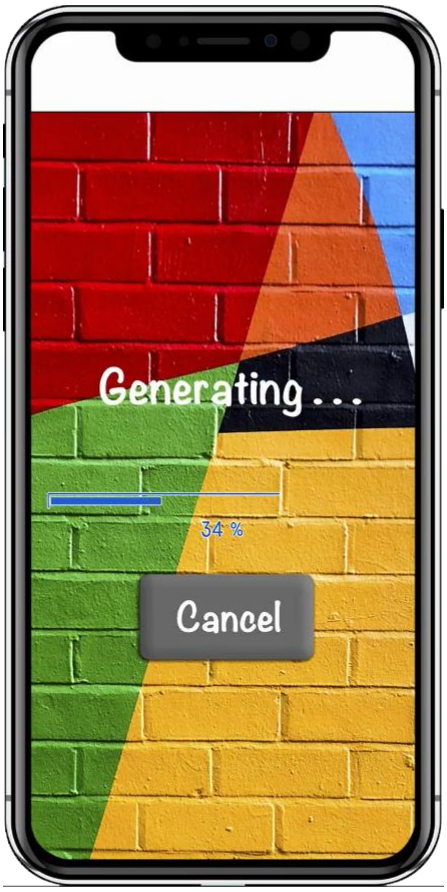


Figure 11 - Character generating

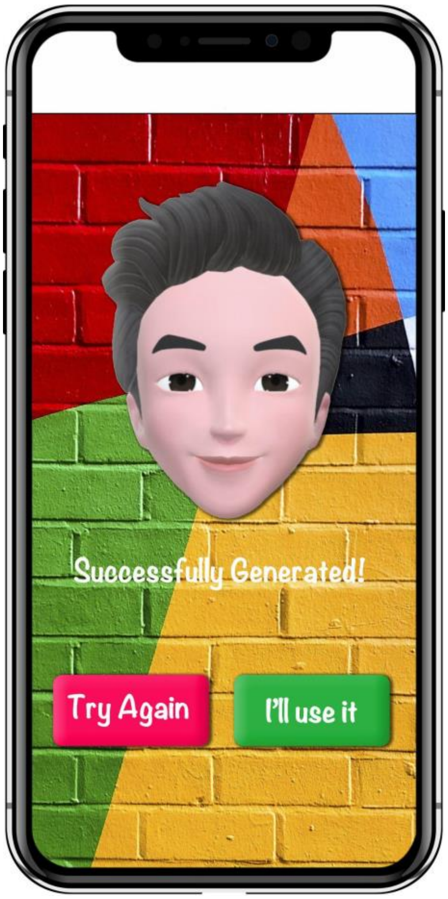


Figure 12 - Character generation is done

Generating

After getting the picture input of the user, to show the progress of the generating process, a progress indicator is equipped (figure 11). A progress bar with a clear percentage presentation is located in the middle of the interface. Below the indicator is a grey cancel button for users to cancel the operation and exit the page whenever they want.

Complete

To show the completion of the generating, this interface (figure 12) presents an inspiring success sentence with the previewed face model.

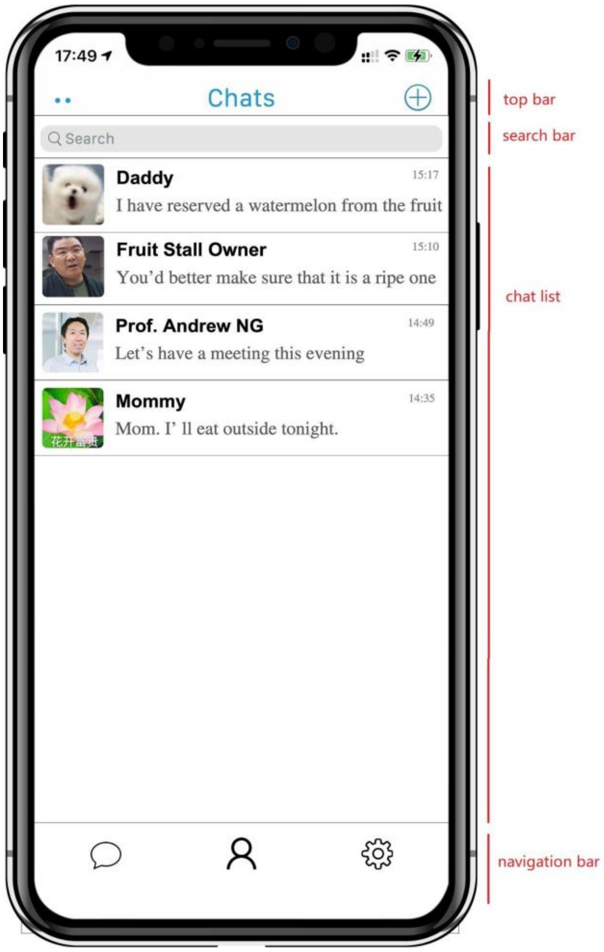
If the users are not satisfied by the character, there is a retry function with a red color to warn that this operation will cancel all processes beforehand and redo. The right hand side green “I'll use it” button is an ack button to confirm and finish the capturing process.

4. Interface Design Principles

To achieve interface design goals and develop a mobile application that provides users with effective and satisfactory interactions. We adopted several design principles and characteristics, which will be discussed in the following section.

4.1 Gestalt principle

1. Proximity principle



Daddy 15:17
I have reserved a watermelon from the fruit

Fruit Stall Owner 15:10
You’d better make sure that it is a ripe one

Prof. Andrew NG 14:49
Let’s have a meeting this evening

Mommy 14:35
Mom. I’ ll eat outside tonight.

Figure 13 Chat list interface

The proximity principle states that elements placed close to each other would be considered one group.

We embrace this principle in the chat box area of the chat list interface. As shown in the figure 13, each of the chat boxes is separated by two grey lines, indicating that each is an independent component. Besides that, the profile picture, chat name, description of time and the last message sent/received are firmly placed close to each other, further demonstrating that the above information belongs to the same chat. This helps users recognize a chat in the chat list interface and reduces the time to find relative information of a single chat.

2. Similarity principle

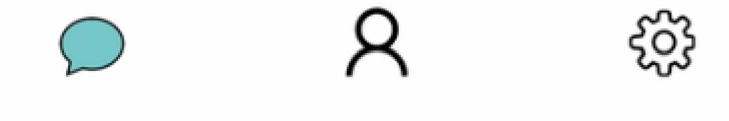


Figure 14 Navigation bar

The similarity principle says that objects with similar visual characteristics, such as size, shape and color will be seen as one group, and share similar functionality.

We adopted this principle in the navigation bar (figure 14). The navigation bar comprises three parts, the chat button 🗨, contact button 👤 and the setting button ⚙. They are placed on the bottom of the screen closely. When one of the buttons is clicked, the screen would be changed to the corresponding interface. Due to the proximity principle, users would portray a relationship between ‘change to another interface’ and ‘click the button on the navigation bar’. This reduces the time to learn the operation, which satisfies the goals of interface design.

3. Continuity principle

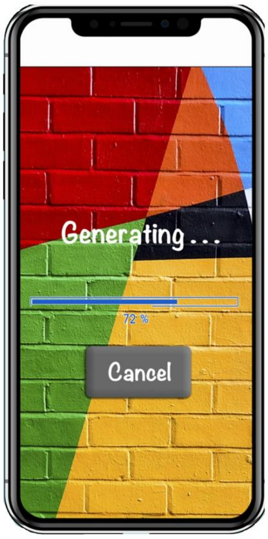


Figure 15 Generate character interface

The continuity principle expresses that users tend to see smooth, continuous representations.

This principle is performed in the character generating interface (figure 15). After users take the photo, a progress bar would be shown to let the users know the percentage of generating the personalized character. The progress animation grows smoothly from 0% to 100%, which makes users feel more comfortable. It also satisfies the goal of user interface design since it could increase users' subjective satisfaction.

4. Closure principle

The closure principle indicates that humans tend to see things as complete objects even though there might be gaps in the shape of the objects.

In our design, all the chat tabs and message bubbles are all complete and closed figures (figure 7). Take the message bubbles as an example. Since these shapes are all complete shapes, when the message bubbles are intercepted by the upper border of the interface, users can easily imagine the whole shape of the bubble. Therefore, they will realize that more messages are hidden above naturally.

5. Prägnanz principle

The Prägnanz principle shows that we tend to perceive things based on the simplest and most stable or complete interpretation. Nowadays, the design of icons has a trend to be more flat instead of the practical 3-D design which was prevalent a few years ago.

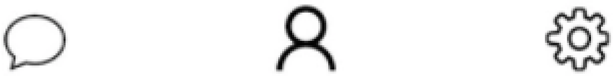


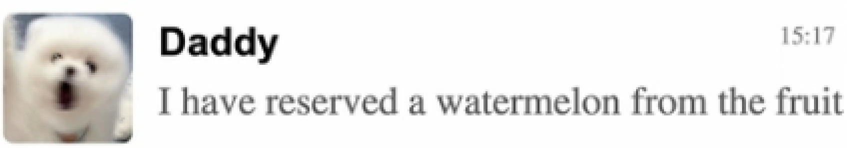
Figure 16 Navigation bar

As shown in figure 16, our design of the icon of the navigation bar conforms to this principle to show a more stable shape instead of applying a rather complicated design.

4.2 Other design characteristics

Apart from the Gestalt principles, there are some important design principles that we have used in the application. Such as design of stimulus intensity, cater for eye movement and reducing short-term memory usage. They will be discussed in this chapter.

1. Design of stimulus intensity



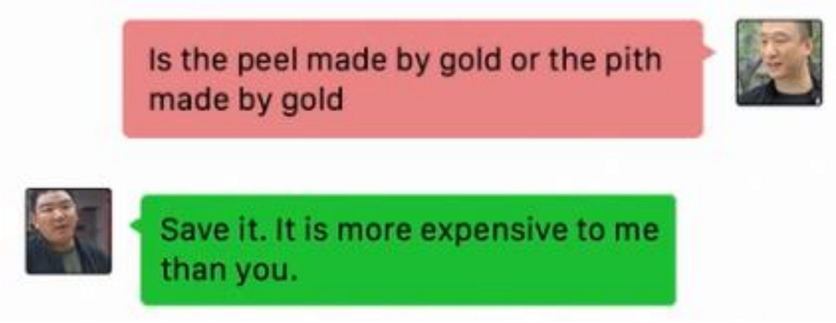
Daddy 15:17

I have reserved a watermelon from the fruit

Figure 17 Chatbox in the chats interface

The chat box design in the chat list complies with the stimulus intensity. As users tend to respond first to the intensity, we bold the chat name and make it slightly larger than the chat description (figure 17). By doing this, users could be attracted by the pop-out effect and reduce the time of differentiating chats.

2. Cater for the eye movement



Is the peel made by gold or the pith made by gold

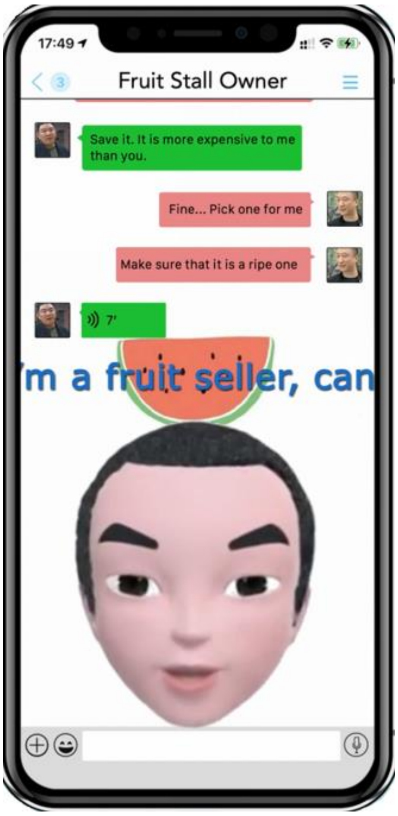
Save it. It is more expensive to me than you.

Figure 18 Example of chats

When users are reading the text, their eye gaze tends to return to the position they start to read. Thus, all the texts start in the same position (figure 18) when a new line is needed to accommodate this eye movement instinct.

4.3 Users with special needs

People with special needs are common in the world. Thus, the application also accommodates their needs, such as providing the text and the animation of the audio. Two groups of special users will be discussed in this chapter.



Save it. It is more expensive to me than you.

Fine... Pick one for me

Make sure that it is a ripe one

7'

'm a fruit seller, can

Figure 19 Chat interface with character

1. People with neurodevelopmental diseases

As mentioned in section 2.4, one of the target users of Message Animation is those who suffer from neurodevelopmental diseases like Asperger syndrome. As they may not fully understand the hidden emotion of a text, the application provides an animation of the audio.

When the audio button is clicked, a personalized character would pop up and speak like a human (figure 19). The character could express precise emotion by moving the face components. By doing this, people with neurodevelopmental diseases could easily understand what other users are talking about and interpret the hidden message of the text. They could also share the sentiment with other users and practice how to express the sentiment in the face in a correct way.

2. People with hearing impairments

WHO mentioned that 5.3% of the world population has hearing loss, which comprises about 360 million people [6]. It is a huge population that we need to take care of. Since those hearing impaired may not hear the audio, the application generates a caption for them. As shown in the figure 19, the application would display the captions on top of the animation. This design solves hearing difficulties as they could understand the audio by reading the caption.

4.4 Object Action Interface Model (OAI)

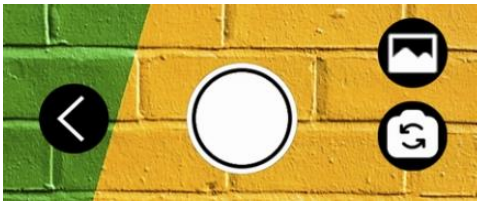
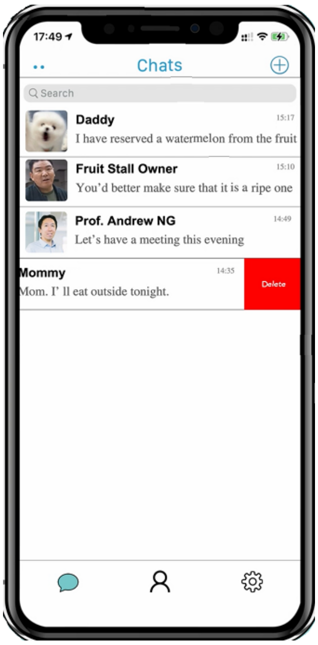


Figure 20 Button group of the face capture interface

The object action interface model states that an object should have a metaphoric meaning of the real-world object. The Message Animation applies OAI on the icons such that users could easily build a mapping between the icons and the specific action to be performed. For example, the 💬 icon of the navigation bar (figure 16) represents the 'chat' action, ⚙ represents the 'setting' action, 🖼 in the face capture interface (figure 20) means 'select photo' action.

4.5 Keystroke level model (compare deletion operation with WhatsApp)



17:49

Chats

Search

Daddy 15:17
I have reserved a watermelon from the fruit

Fruit Stall Owner 15:10
You'd better make sure that it is a ripe one

Prof. Andrew NG 14:49
Let's have a meeting this evening

Mommy 14:35
Mom. I' ll eat outside tonight.

Delete

Figure 21 Delete chat - Message Animation

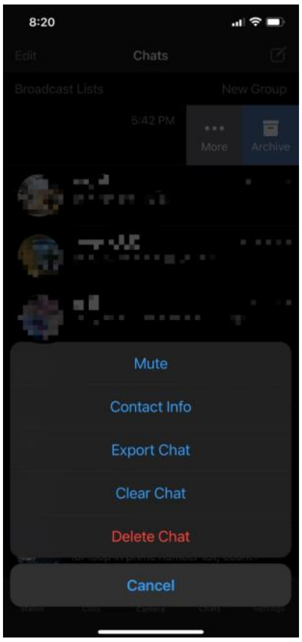


Figure 22 Delete chat - WhatsApp

The keystroke level model is also applied to predict the time of chat deletion operation between Message Animation and WhatsApp. Users could scroll the chat to the left in Message Animation and click the delete button to delete the chat from the chat list. While in WhatsApp, users need to scroll chat to the left and then click the 'more' button. After that, they can select the delete chat button.

It is assumed that both users are intermediate-level users and have average typing speed. The detailed keystroke level model is as follows:

Message Animation	**WhatsApp**
Mentally Prepare - delete chat (0.5s)	Mentally Prepare - delete chat (0.5s)
Position for finger to the chat (0.4s)	Position for finger to the chat(0.4s)
scroll the chat into left (0.4s)	scroll the chat into left (0.4s)
Mentally Prepare - select “delete” button (0.5s)	Mentally Prepare - select “more” button (0.5s)
press the delete button (0.2s)	Press the more button (0.2s)
release finger (0.2s)	release finger (0.2s)
	Homing - put finger back to the original place (0.4s)
	Mentally Prepare - select “delete” button (0.5s)
	Press the delete button (0.2s)
	release finger (0.2s)
Overall: 2.2s	Overall: 3.5s

Table 1 Keystroke level model between Message Animation and WhatsApp

By applying the keystroke level model, it is observed that the time required for deleting a chat in Message Animation is fewer than the time in WhatsApp.

4.6 Eight golden principles of interface design

1. Aim for consistency

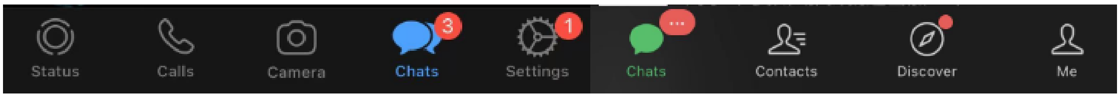


Figure 23 Navigation bar of WhatsApp (left) and WeChat (right)

Message Animation adopts similar icons and layout from WhatsApp and WeChat (figure 23). These two applications are the most commonly used instant messaging application around the globe. By doing so, users could easily understand the icon's meaning since they could apply the mapping between a specific icon and action.

2. Cater to universal usability

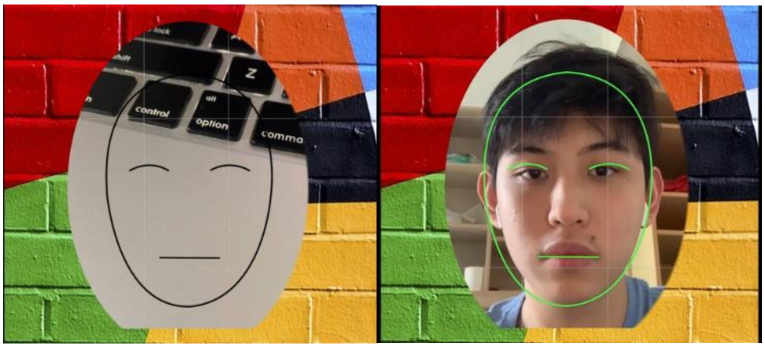
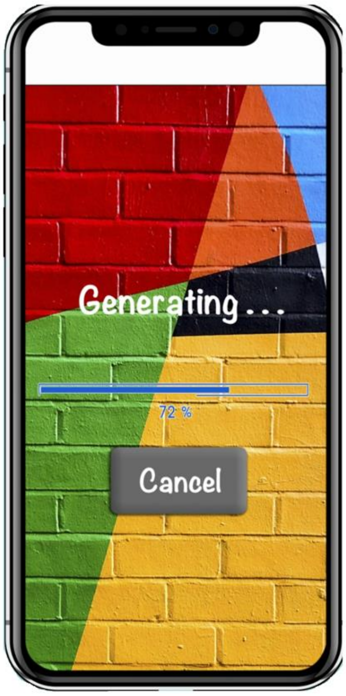


Figure 24 Face landmark guideline for novice users in character generating interface

The application also caters to universal usability. It is considered that there would be first-time users who do not know how to generate character by taking a selfie. In such a case, we provide a face landmark on the camera view (figure 24). When users' face match the face landmark, the landmark will become green. Thus, novice users would recognize an appropriate place to take the photo. For those intermittent users and expert users, they could turn off the face landmark in the setting interface.

3. Offer error prevention and error handling



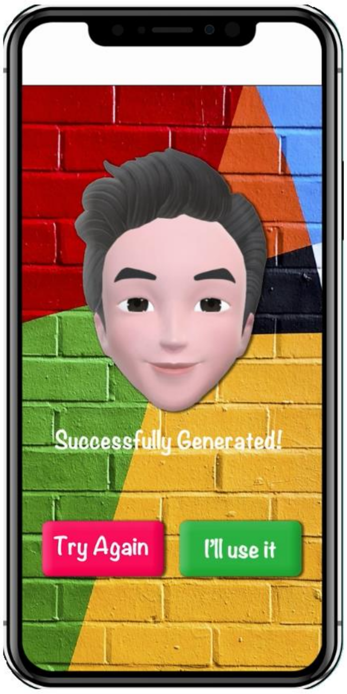


Figure 25 Character generating progress

Figure 26 Character generation is done

Message Animation offers error prevention and error handling mechanisms to the users, one of the examples is in the character generation interface. If the users have the intention to take another photo after taking one, they can click the cancel button while generating the character (figure 25), if they missed the cancel button. They could still press the 'Try Again' button in the last step of the character generation process (figure 26).

4. Offer informative feedback

The application also offers informative feedback to the users. For example, when users click the 🗨 in the navigation bar, it’s color would turn to blue 💬 (figure 14), indicating that the current interface is the chat list interface. Besides that, when users want to create a character, we would provide a progress bar to inform the users of the creation progress (figure 25).

5. Design dialogs to yield closure

We have organized a sequence of actions to yield closure in the character generation process. When users take an photo and press the ○ icon in the photo-taking interface (figure 20), the progress begins. After that the users would be guided to the middle stage of the progress (figure 25). At the end, we provide two options for the users to decide whether to use the character or not (figure 26). The above three stages could give users a satisfaction of accomplishment when the stage is finished.

6. Permit easy reversal of actions

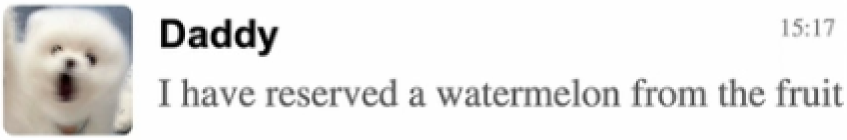
Similar to other instant messaging applications, Message Animation permits easy text reversal. When users sent a text and wanted to withdraw it, they could press the text for two seconds, then an option list would pop up. The users could press the ‘withdraw’ button to delete the message.

7. Allow users to be in control

Message Animation allows users to be in control. For example, users can set up the audio received not to be played automatically. In this case, it would start playing only when users click the audio. Besides that, when the character is performing animation in the chat interface (figure 19), the users could stop playing it by tapping on the chat area or swiping down the screen.

Another example is during the process of character generating. When they want to discard the generating process and want to retry or exit this interface, they can tap the cancel button, avoid being stuck in that page.

8. Reduce short-term memory load


Daddy 15:17

I have reserved a watermelon from the fruit

Figure 27 Chat box example

Researchers indicated that users’ short-term memory is limited compared to long-term memory [7]. Similar to the mainstream instant messaging applications such as WhatsApp and WeChat. We provide a description of the last text in the chat box (figure 27) to reduce short-term memory usage. Therefore, the users could easily recall the contents of the last chat with the corresponding person.

5. Guidelines

User interface design guidelines are important to the interface design, it makes learning easier since if the application follows the guideline's standard, users could pay only a little effort to learn how to perform specific operations. We have adopted several guidelines in the Message Animation, they will be discussed in this chapter.

5.1 Menus

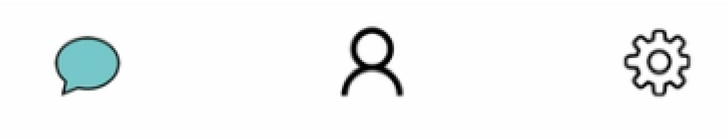


Figure 28 Navigation bar

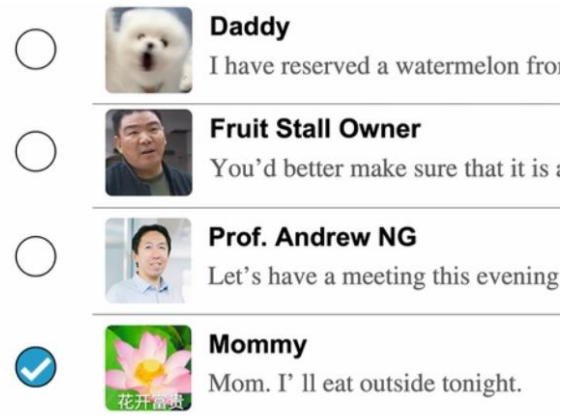
Characteristic:

The navigation bar (figure 28) follows the menu design guidelines. We apply the OAI model to the icon design so that users do not have to recall the meaning of each icon. They could simply operate it by recognition. It matches the characteristic of the menu "emphasis recognition over recall."

Item presentation sequence:

We adopt the "most frequently used items first" rules for designing the sequence of the navigation bar. It is clear that in an instant messaging application, the chat page would be called most frequently, followed by the contact page and then the setting page. Therefore, we emphasize the importance of the chats and set it as the first item of the navigation bar.

5.2 Checkboxes



○ **Daddy**
I have reserved a watermelon fro

○ **Fruit Stall Owner**
You'd better make sure that it is

○ **Prof. Andrew NG**
Let's have a meeting this evening

✓ **Mommy**
Mom. I' ll eat outside tonight.

Figure 29 A checkbox is clicked in the delete chat operation

The multiple chats deletion operation follows the checkbox design guidelines. When the choice is set, a check mark appears in the checkbox (figure 29). Also, each checkbox is accompanied by a chat on the right side, users would know the checkbox is for that specific chat. It matches the guidelines of the checkbox “appears with an accompanying label” and “shows a checkmark when it is selected.”

5.3 Progress indicator

The progress bar of the character follows the progress indicator guidelines. The bar (figure 25) consists of a solid segmented rectangular bar filled from left to right. It is very useful for showing the percentage of completion of the character generation.

5.4 Controls

The general guidelines of controls are to provide a label to identify the purpose of a control and increase the area of the hot zone.

- Provide a label to identify the purpose of a control

The application strictly follows this guideline. For example, in figure 26, two controls are displayed when the character generation process is done. Both of them have the English label on the control buttons, namely “I’ll use it” and “Try Again”.

- Increase the area of the hot zone

Message Animation also complies with this guideline. We take 1.35s as the maximum time needed for clicking a button from the starting point since it is the average time demonstrated by researchers [8]. Below shows an example of the clicking time of the buttons by Fitts’ law.

Example:

The goal is to estimate the time of clicking on the center of the “I’ll use it” button (figure 26) from the thumb's current position.

By Fitts’ law, t = a + blog2(D/S +1)

Where D is the distance from the current finger position to the center of the target, and S is a measure of target size (which is min (width, height)).

For the constants a and b, we assume that they are equal to 50 msec and 150 msec respectively. Regarding the target size S, the width and height of the “I’ll use it button” is 40% and 10% respectively. Besides, we assume the user is using iPhone 13 with a screen size of 13.15cm * 6.42 cm. Therefore, the corresponding target size S of both designs would be:

*S = min (6.42*40%, 13.15*10%) = min (2.568, 1.315) = 1.315cm*

As for the initial distance between the chat and the finger, we assume that the users are right-handed since Scharoun and Bryden[9] state that right-handedness is the majority. Using this assumption, users would grab the phone with their right hand and the thumb would be placed at the bottom right corner of the phone. Therefore, the distance D from the thumb to the center of the “I’ll use it” button would be:

$$D = \sqrt{width^2 + height^2} = \sqrt{1^2 + 4^2} = 4.123cm$$

Sub those variables into the Fitts’ law equation, the estimated time would be:

50 + 150log2((*4.123*/1.315) + 1) = 357ms, which is lower than the throughput 1.35s.

6. Alternative Modality of Interaction

As for the alternative modality of Interaction in the system, we have applied many cutting-edge technologies in the computer vision area. On the input side, apart from the traditional mobile input methods such as text input or voice input, we also utilize the camera to realize the picture or video input for the cartoon head portrait generation. On the output side, we use animation or in other words video outputs to greatly enrich the user's interaction with the computer and their contactor, which is a good supplement to the traditional output formats such as text or audio.

As for the detailed multi-modal interaction of our system, in one word, it will transfer the user's text or voice input into a piece of animation, in which a cartoon character generated by the user's picture will talk to the user with the sender's voice and corresponding appropriate face movements. In a more vivid description, it is just like a cartoon character talking in front of you in the representation of the contactor.

6.1 Input side:

6.1.1 Camera Guidance

Here we want to provide some additional guidance for the user when they capture their face to generate the cartoon character. That is when the user's face in the camera coincides with the reference face landmarks on the screen, changing the color of the reference face landmarks to guide the users.

Methods of implementation

Here we implement this effect by traditional face detection technology. We use p5.js and ml5.js library to implement a simple demo. The detection results of ml5.js face detection API contain the corresponding face landmark information of detected faces. We compare the coordinates of these detected landmarks with the reference landmarks and when they are similar in value, the reference facial landmarks will turn green.

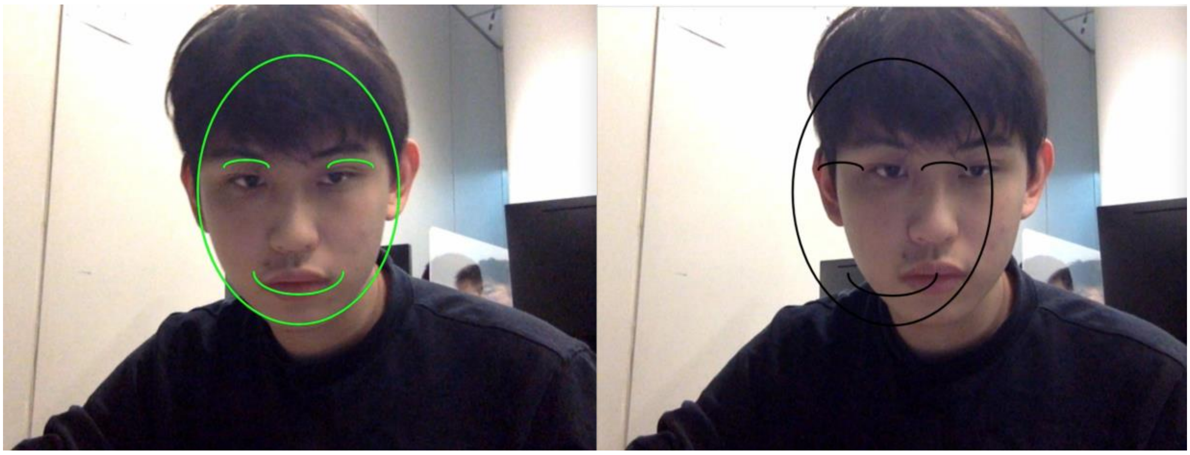


Figure 30 The demonstration of the implemented capturing guidance

6.1.2 Cartoon Character Generation

In this step, we want to generate a picture of the cartoon character based on the camera face capture of the user. In other words, it is a face to image transformation.

Related Work

In recent years, there are many related computer vision technologies that have realized this effect. Most of them are based on the GAN. CycleGAN [10] is the pioneer in this area which realizes image-to-image translation without using a training set of aligned image pairs. It meets our requirements where we only have one side input face image. StyleGAN [11] is a more advanced method provided by Nvidia which utilizes transfer learning technology and has a better performance than CycleGAN [10]. Newly released AgileGAN [12] focused on generating characters of specific styles, which perfectly meets our target of generating cartoon-style characters. It has a higher resolution output and less transfer time, which is also suitable for later mobile device deployment.

Methods of implementation

After comparing many existing technologies, we decided to choose DeepAI’s Toonify API [13]. First of all, it is a newly proposed model and has a similar performance as the AgileGAN [12] mentioned before. Moreover, it is deployed on the central server and accessed through a network API, which hardly has requirements for hardware's computational power and is suitable for our mobile device deployment. Finally, its API is simple and suitable for many popular programming languages, which also ease the difficulty of development.

In our implementation, we write a piece of python code to call the API and get the generated face image back, which can be used in the later cartoon animation generation. The detailed python code can be seen in the source code[1].

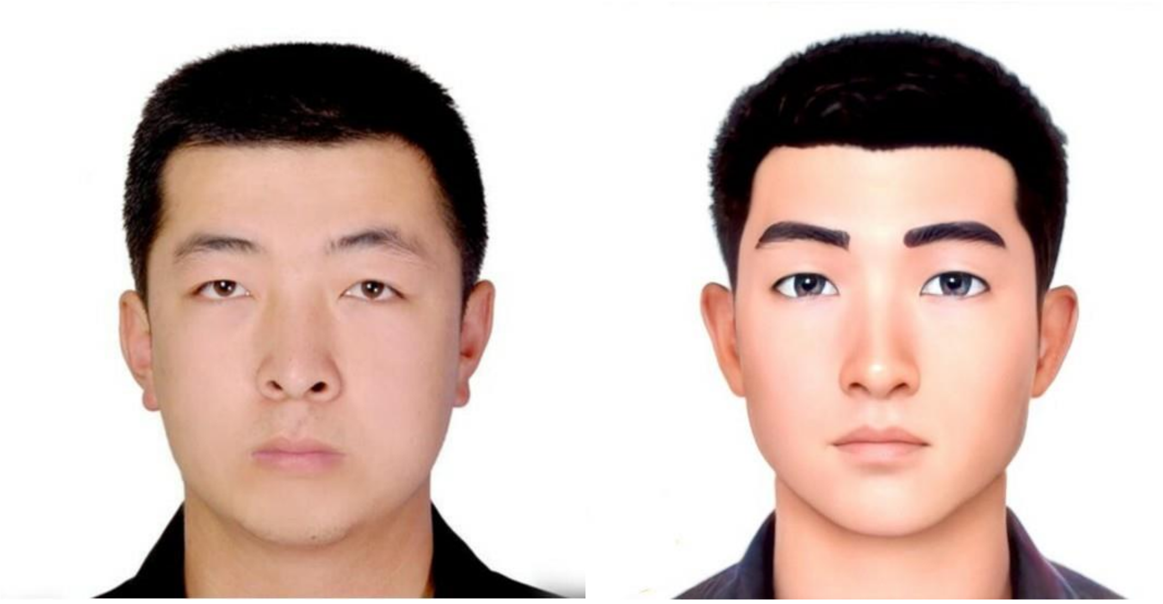


Figure 31 Demonstration of the effects of the toonify API

[1] https://github.com/DELTA-DoubleWise/Message_Animation

6.2 Output side

6.2.1 Text/Voice to Cartoon Animation

On the output side, we want to transfer the user’s textual or vocal input into a piece of cartoon animation video which can be played on the receiver side of the contactor.

Related Work

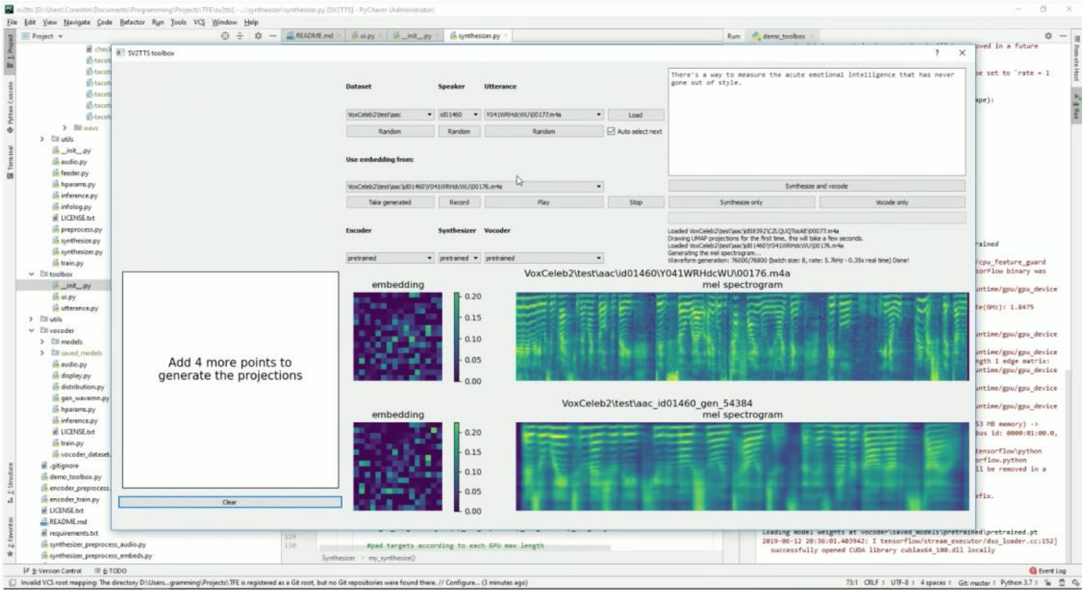
To implement this amazing multimodal interaction, we do some literature reviews to find some cutting-edge technology to implement this effect. Netease’s Flow-guided One-shot Talking Face Generation [14] can synthesize high visual quality facial videos with reasonable animations of expression and head pose, and just utilize arbitrary driving audio and arbitrary single face image as the source, which perfectly meets our requirements to synthesize a piece of animation video according to vocal input. Although it is currently not open source, it can still be used as an industry-leading reference standard. Netease’s Text-based Emotional and Rhythmic Talking-head Generation [15] can generate talking head videos based on textual input, which perfectly meets another requirement. It can synthesize high-fidelity facial expressions and head motions in accordance with contextual sentiments as well as speech rhythm and pauses. This work presents an innovative idea of adjusting the talking head’s facial expression according to the sentiments analyzed from the textual input, which we also take into consideration and hope to implement it in future work. However, similar to the previous one, it is also not open source, regretfully.

Methods of Implementation

After reviewing several existing technologies, we propose a novel and applicable method to realize our target features. This methodology mainly contains three phases: text to voice, audio to face, and deep fake face change.

Text to Voice:

For the text input, we need to transfer it to a piece of voice audio with the contacter’s vocal features for later animation generation. Here we use the Multispeaker Text-To-Speech Synthesis (RTVC) [16] technology which can transfer a piece of text to speech that is similar to the target speaker. This technology is able to synthesize natural speech from speakers unseen during training. It only needs a piece of 5 seconds speech audio of the target speaker as a reference to synthesize any other speeches based on the reference. It uses a speaker encoder network, a sequence-to-sequence synthesis network based on Tacotron 2, and an auto-regressive WaveNet-based vocoder network to realize the training set encoding and target speech synthesis.



```
There's a way to measure the acute emotional intelligence that has never gone out of style.
```

```
Loaded VoxCeleb2\test\aac\id08392\CZLQUQTssAE\00077.m4a
Drawing UMAP projections for the first time, this will take a few seconds.
Loaded VoxCeleb2\test\aac\id01460\Y041WRHdcWU\00176.m4a
Generating the mel spectrogram...
Waveform generation: 76000/76800 (batch size: 8, rate: 5.7kHz - 0.35x real time) Done!
```

```
Loading model weights at vocoder\saved_models\pretrained\pretrained.pt
2019-06-12 20:36:01.403942: I tensorflow/stream_executor/dso_loader.cc:152]
  successfully opened CUDA library cublas64_100.dll locally
```

Invalid VCS root mapping: The directory D:\Users...gramming\Projects\TFE is registered as a Git root, but no Git repositories were found there. // Configure... (3 minutes ago)

Figure 32 Interface of the original RTVC toolbox by Jemine Corentin's team

In our actual implementation, we utilize RTVC's [16] open-source code and make some modifications to make it better fit our one command execution and simple IO requirements. The original model is implemented as a GUI-based application, which does not meet our application scenario. We shield the GUI part in the source code and redirect the input-output stream to the command line. Moreover, we add several command-line arguments to let the user set the input text and output audio. In addition, in consideration of some of the devices that do not have a suitable GPU for acceleration, we also add an option for the CPU to work. Finally, we use anaconda to construct the required python environment and use a simple command to realize easy input-output of text and speech. The detailed demonstration and modification can be seen in the demo video and source code[2].

Audio to Face:

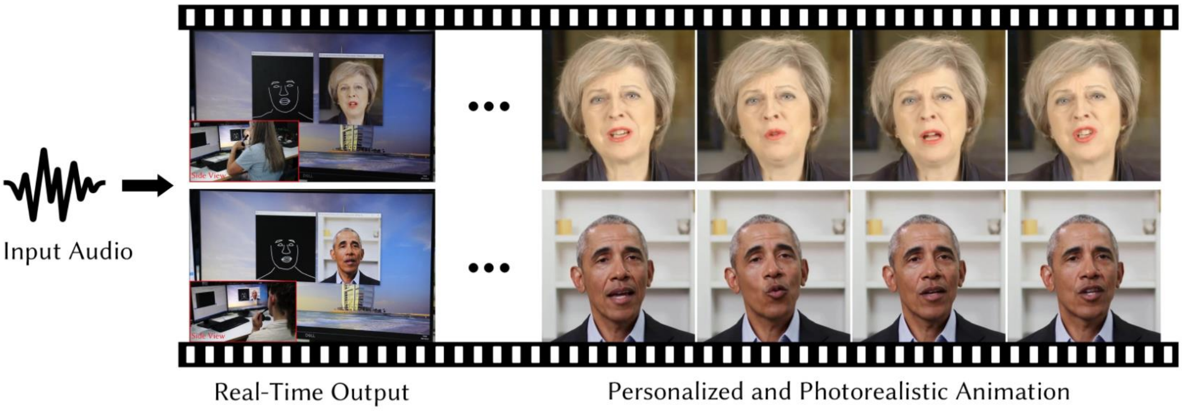


Figure 33. Example of the LSP implementation

[2] https://github.com/DELTA-DoubleWise/Message_Animation

This is the core technology in our system that applies the cross-modal associations learning, which constructs associations between people's voice and face. Here we can transfer a piece of a person's voice audio (recorded or synthesized) into a piece of video that contains corresponding appropriate face movements such as lip movements or eyes movements.

Due to the limitation of our computational power, we focused on finding some pre-trained models to realize this effect. First is the previous Netease Flow-guided One-shot Talking Face Generation [14], which only needs one shot as input and perfectly meets our requirements. However, it is a newly invented method and is not open source. Another choice is Nvidia's Omniverse Audio2face [17]. It is a pre-trained 3D model which can be driven by the user input audio with synchronized face movement. Its advantage is that it is a 3D model. However, it also has some disadvantages. First, it only has one pre-trained model, and the creator needs to manually map key facial features such as mouth and eyes with the pre-trained model, which is not suitable in real industry deployment. Another problem is that it is a GUI application which is not easy for automation script in our application scene and it needs Nvidia RTX graphic cards as a hardware requirement, which is also not applicable for more devices.

Finally, due to the limitation of these cutting-edge modals, we decided to choose the Live Speech Portraits (LSP) [18] to realize our implementation. Although it needs no more than one target face picture to train the model, it has several pre-trained models and this problem can be solved by the later deep fake face change method. Moreover, its real-time feature perfectly meets our requirements and it is fit for our later requirements.

This LSP [18] method uses a deep neural network that extracts deep audio features along with a manifold projection to project the features to the target person's speech space and then learns facial dynamics and motions from the projected audio features.

Here we use the demo.py as the core to transfer our input audio to a pre-chosen person's output video with the pre-trained model. In our actual implementation, the core problem we need to solve is which pre-trained model to choose as LSP provides several pre-trained models trained with different parameters and datasets. In the original scenario, we decide to choose the dataset which has the same gender (male) as our target character to get better compatibility for later deep fake face change. However, all the male pre-trained models such as Obama and Nadella have the problem that their lip movement is too small, which makes the output results of the final deep fake look strange and perform badly. Finally, we decide to choose the female dataset May, whose lip movements look more natural and smooth although its gender is different. In addition, as this step should be integrated with the next deep fake step, we add the execution command of this LSP part to the python notebook of the deep fake part to realize the one command input-output effect of the final two steps.

Deep Fake Face Change:

As the audio2face [17] part only has one pre-trained model and the creator needs to manually map the key face points of the target model with the pre-trained model, this does not meet our requirements. This problem is similar for the LSP [18], as we only have one picture of the target character, we cannot use it to train another model with LSP's method. To solve this problem, we decided to apply the one picture deep fake face change technology to the output video to change the face of the pre-trained model to our target face image. To be more specific, this is just like letting the AI do the key face points mapping work automatically for us.

There are also many methods in deep fake technology. DeepFaceLab [19] is the leading technology for this area, however, it needs several target face frames (can be extracted from a piece of video of the target face) to form the new video, which does not meet our application scene where there is only one input picture.

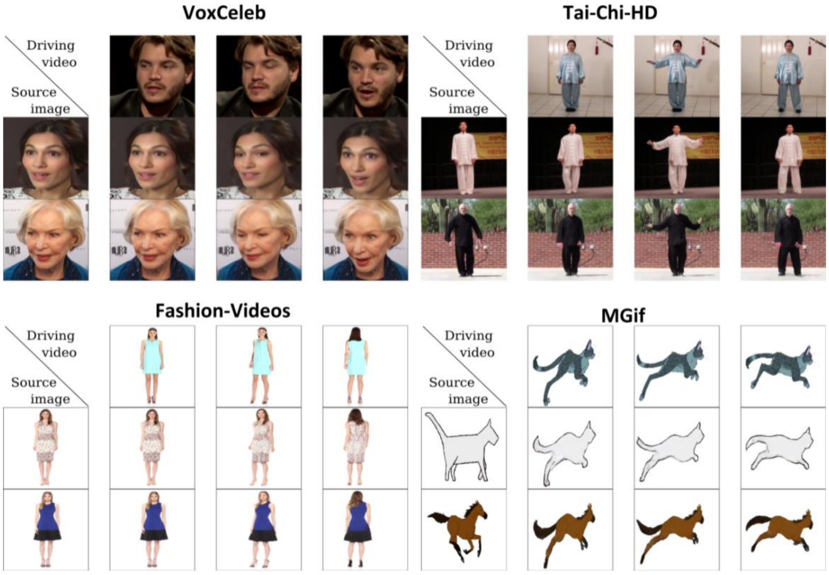


Figure 34 Example animations produced by first order motion models

We apply the First Order Motion Model for Image Animation’s [20] method, which only takes one face picture as input. Moreover, it is open source and also implements as a python notebook, which is easier for our later application. Moreover, this Image animation is not restricted to faces, it can also be extended to any other objects, which means that we can promote our face animation to a whole-body animation. This technology uses self-supervised formulation and a generator network to form the output video with the input picture.

Here we also use its open-source python notebook [20] to realize the procedure and form the final video. In our actual implementation, first, we need to resize our input picture to fit the input requirements. The original image input and resize method of the First Order Motion Model has exceptions in some cases. So we change it to a new image IO method and resolve the exception. Moreover, another core problem is the choice of the animation-making parameters. After several experiments, we decided to choose the relative=True parameter which makes the movement of the face landmarks more smoothly and discarded the adapt_movement_scale=True which adjusts the movement scale of the image. Finally, together with the previous LSP audio to face step, we construct a python notebook with which one click realizes all the procedures and gets our final output.

6.3 Conclusion

With the technologies and implementations described above, we can realize the desired multimodal interaction. The whole system can be deployed on a high-performance central server and accessed through network IO, so even if the mobile devices may not have enough computational power, the whole system can still be functional, though nowadays some mobile devices already have enough computational power to deal with such a task.

7. Future Plan

7.1 Message Animation Adaption to Group Chat

In our current design, the cartoon character message animation only works for a one-to-one chat. However, adapting it to group chat is also a good idea as our message animation can also make boring text group chat more vivid. In the group chat, instead of showing one character to say the message to the user, several characters will jump out together and start a discussion, which looks just like a real-time in-person face-to-face round-table conference.

7.2 Change 2D character to 3D character

In our current design, the cartoon character is a 2D image, which looks not so stereoscopic and vivid. However, as we have mentioned before, Nvidia's audio2face [8] already has the ability to drive a 3D human face model to speak with input audio. In our future plan, we plan to implement our Message Animation in a 3D model. There are already some technologies that can transfer the camera face capture to a 3D face model; the only problem that needs to be solved is the automatic key face feature landmarks mapping.

7.3 Change head portrait to whole-body character

In our current design, the cartoon character is only a head portrait, the First Order Motion Model [11] gives us the inspiration that we can also drive a whole-body character to animate as the First Order Motion Model [11] can transfer the whole-body motion from one video to another. The problem that needs to be solved is how to map the audio features to body movement just like the facial expression.

7.4 Character facial expressions adjustment based on contextual sentiments

In our current design, the cartoon character's facial expression is only based on the audio, which does not have corresponding facial expressions based on emotion. Netease's Write a Speaker [6] gives us the inspiration that we can map between the textual contextual sentiments with the facial expression. As Netease has already realized this technology [6], we think that it will not be difficult to deploy it to our system if it is open source later.

7.5 Voice tone and rhythm adjustment based on contextual sentiments

Similar to the previous facial expression adjustment, the textual contextual sentiment can also be used to adjust the tone and rhythm of the output voice. This technology is called emotional voice conversion (EVC) [12], which has already been realized recently. It can convert the spectrum and prosody to change the emotional patterns of speech while preserving the speaker's identity and linguistic content. However, due to the limitation of training computational power, we decided to put it in the future plan.

8. Responsibilities

WANG Yucheng: Video Demonstration, System Implementation, Overall Review

YAO Shenglong: Alternative Modality of Interaction, System Implementation, Future Plan

TAM Kam Chuen: Interface Design Principles, Interface Design Guidelines

ZHOU Yukun: Interface Relationships and Associated Features

CHAN Hiu Leong: Overall Introduction, Analysis of the User Community

9. Reference

[1] Singh, Vijendra. (2018). Language and Body Language.

[2] V. Campion-Vincent, “The Tell-Tale Eye,” *Folklore (London),* vol. 110, no. 1–2, pp. 13–24, 1999, doi: 10.1080/0015587X.1999.9715977.

[3] The London School of Echonomics and Political Science, “A guide to social media platforms and Demographics.” [Online]. Available: https://info.lse.ac.uk/staff/divisions/communications-division/digital-communications-team/assets/documents/guides/A-Guide-To-Social-Media-Platforms-and-Demographics.pdf.

[4] R. Williams, “Study: 71% of mobile users add emojis and gifs to messages,” Marketing Dive, 28-Jun-2017. [Online]. Available: https://www.marketingdive.com/news/study-71-of-mobile-users-add-emojis-and-gifs-to-messages/445987/.

[5] M. Arora *et al.*, “Global, regional, and national incidence, prevalence, and years lived with disability for 310 diseases and injuries, 1990–2015: a systematic analysis for the Global Burden of Disease Study 2015,” *The Lancet (British edition)*, vol. 388, no. 10053, pp. 1545–1602, 2016, doi: 10.1016/S0140-6736(16)31678-6.

[6] Multi-country assessment of national capacity to provide hearing care. Geneva: World Health Organization; 2013. Available from: https://www.who.int/pbd/publications/WHOReportHearingCare_Englishweb.pdf

[7] N. Cowan, “The magical number 4 in short-term memory: A reconsideration of mental storage capacity,” *The Behavioral and brain sciences*, vol. 24, no. 1, pp. 87–114, 2001, doi: 10.1017/S0140525X01003922.

[8] M. F. Roig-Maimó, I. S. MacKenzie, C. Manresa-Yee, and J. Varona Gómez, “Fitts’ Law: On Calculating Throughput and Non-ISO Tasks,” Revista Colombiana de Computación, vol. 19, no. 1, pp. 7–28, 2018, doi: 10.29375/25392115.3226.

[9] S. M. Scharoun and P. J. Bryden, “Hand preference, performance abilities, and hand selection in children,” Frontiers in psychology, vol. 5, pp. 82–82, 2014, doi: 10.3389/fpsyg.2014.00082.

[10] Jun-Yan Zhu, Taesung Park, P. Isola, and A. A. Efros, “Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks,” in 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2242–2251, doi: 10.1109/ICCV.2017.244.

[11] T. Karras, S. Laine, and T. Aila, “A Style-Based Generator Architecture for Generative Adversarial Networks,” IEEE transactions on pattern analysis and machine intelligence, vol. 43, no. 12, pp. 4217–4228, 2021, doi: 10.1109/TPAMI.2020.2970919.

[12] G. Song et al., “AgileGAN,” ACM transactions on graphics, vol. 40, no. 4, pp. 1–13, 2021, doi: 10.1145/3476576.3476684.

[13] J. N. M. Pinkney and D. Adler, “Resolution Dependent GAN Interpolation for Controllable Image Synthesis Between Domains,” 2020.

[14] Z. Zhang, L. Li, Y. Ding and C. Fan, "Flow-guided One-shot Talking Face Generation with a High-resolution Audio-visual Dataset," 2021 IEEE/CVF

Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 3660-3669, doi: 10.1109/CVPR46437.2021.00366.

[15] L. Li, “Write-a-speaker: Text-based Emotional and Rhythmic Talking-head Generation”, AAAI, vol. 35, no. 3, pp. 1911-1920, May 2021.

[16] Y. Jia et al., “Transfer Learning from Speaker Verification to Multispeaker Text-To-Speech Synthesis,” 2018.

[17] “United States : NVIDIA Announces Platform for Creating AI Avatars,” MENA Report, 2021.

[18] Y. Lu, J. Chai, and X. Cao, “Live Speech Portraits: Real-Time Photorealistic Talking-Head Animation,” 2021.

[19] I. Perov et al., “DeepFaceLab: Integrated, flexible and extensible face-swapping framework,” 2020.

[20] A. Siarohin, S. Lathuilière, S. Tulyakov, E. Ricci, and N. Sebe, “First Order Motion Model for Image Animation,” 2020.

[21] K. Zhou, B. Sisman, and H. Li, “Transforming Spectrum and Prosody for Emotional Voice Conversion with Non-Parallel Training Data,” 2020.