CS427 - 3D Visualization and Game Development

FINAL PROJECT REPORT

Minh-Nhut Nguyen, Gia-Han Diep

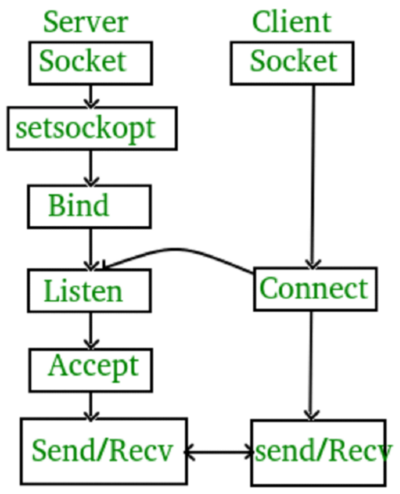
Brief

Hologram has long appeared on screen of sci-fi films, especially in form of hologram-chatting that can show the speaker's expression, simulating the conversation between people.With the advancement of 3D techniques, hologram has come from sci-fi films to real life and become an interesting topic. For that reason, our group choose 'Hologram 3D to express facial expression in video*' as our topic. To-be-processed video is uploaded with TCP socket and user will get a hologram avatar with their facial expression changing over time as in the video (the video must be recorded in close-up so that all parts of the face is visible)

*: the video must contain one and only human face.

Implementation

We use MVC model and TCP Client-Server model as follows:



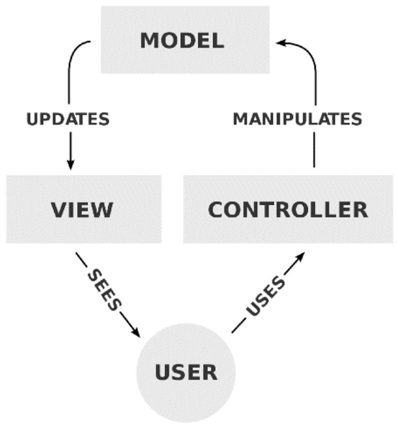


Fig 1: TCP socket Server-Client Fig 2: MVC model

Server creates a TCP socket and bind it to a server address (we use localhost) on port 8888. After that, the server runs in passive mode, listening for connection request from client. Once connection is established, the server will stream data to the client.

As for client, it first creates a TCP socket, and as soon as it is ready, send a 'hand-shake' request to the server. The server then sends packets that contains facial expression data in the video, which the client parses and displays it in the hologram.

In our project, Unity acts as client and Python is responsible for server. Video is first passed into a facial recognition model in server. The server uses the model to determine expressions in the video and waits for client to request.

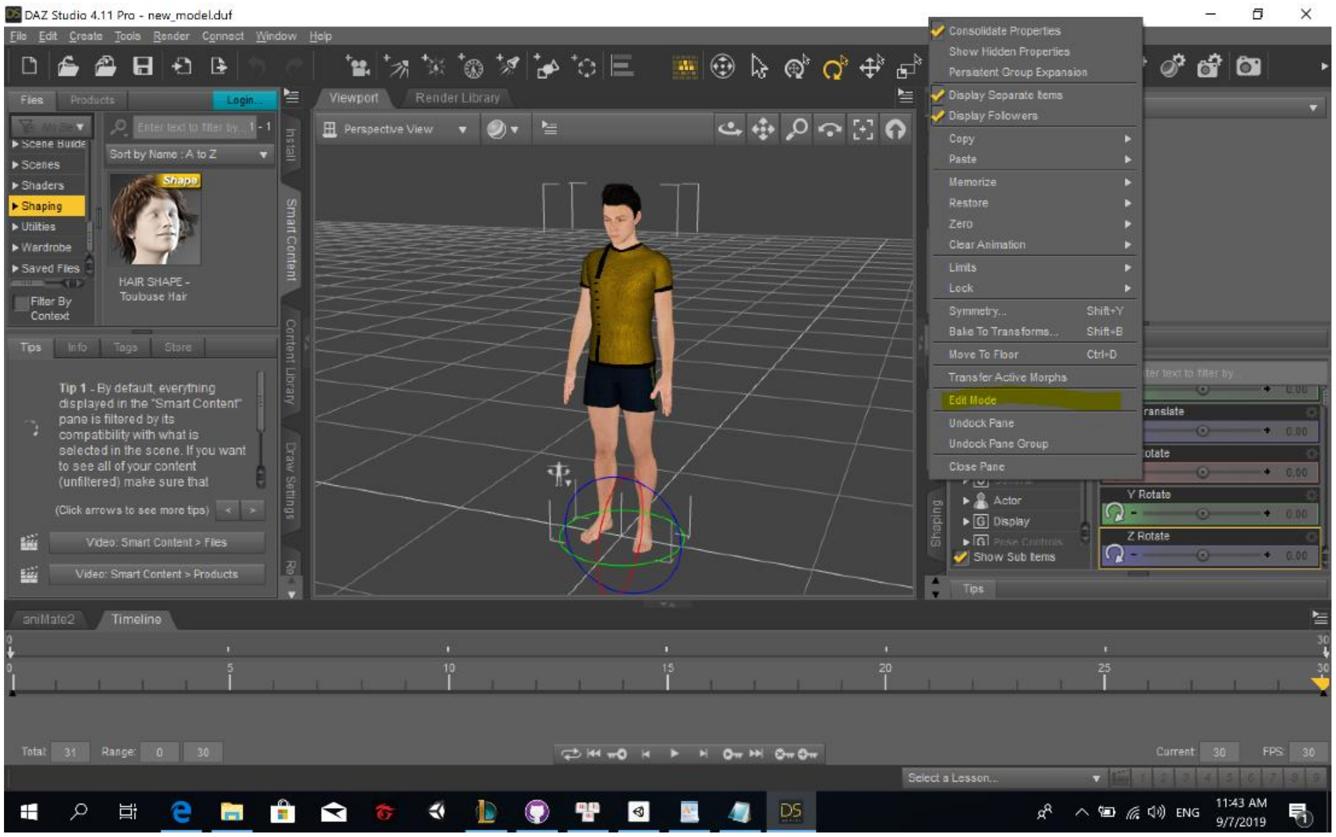
We use 3D model designed with Daz Studio to show facial expression.

3D model design

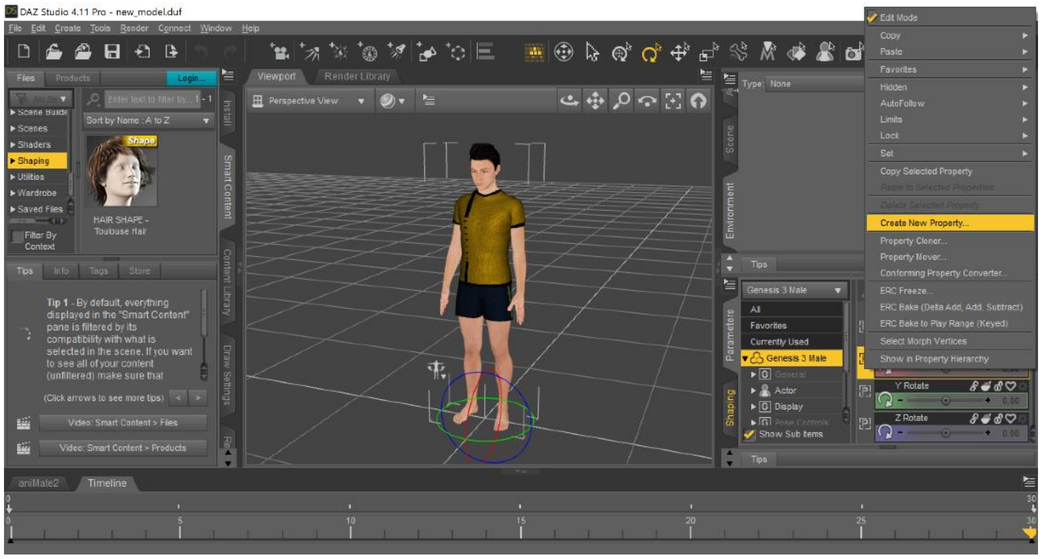
Our project's model can show 7 expressions: Angry, Disgust, Fear, Happy, Sad, Surprise and Neutral.

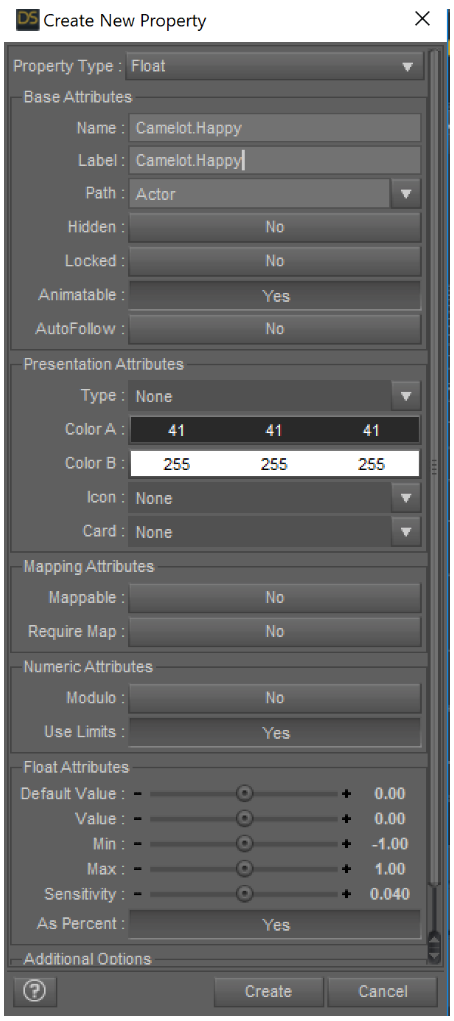
We use Daz Studio to create blendshape properties which Unity can control. Daz Studio is an easy-to-use tool which provides not only Genesis Model but also common blendshapes (or Morph in Daz Studio) which we can combine to create facial expression.

First, drag Genesis Male 3 from Figure on the left. Then we need to create a property. Right click on Parameter tabs and choose Edit mode:



In Edit mode, right-click again and choose Create new property … :





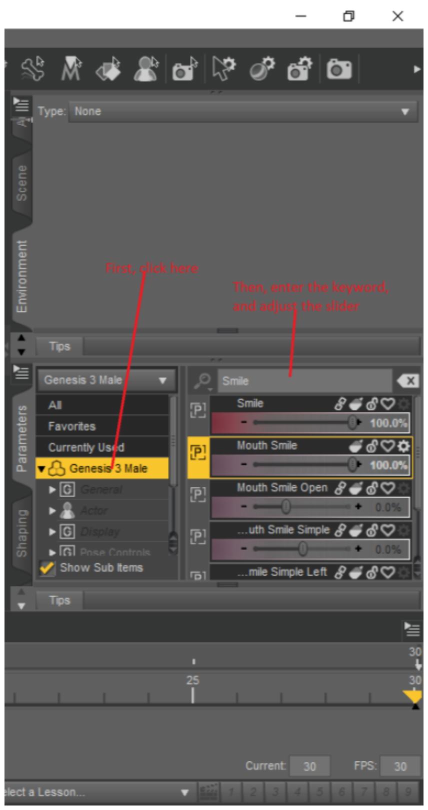
Create New Property

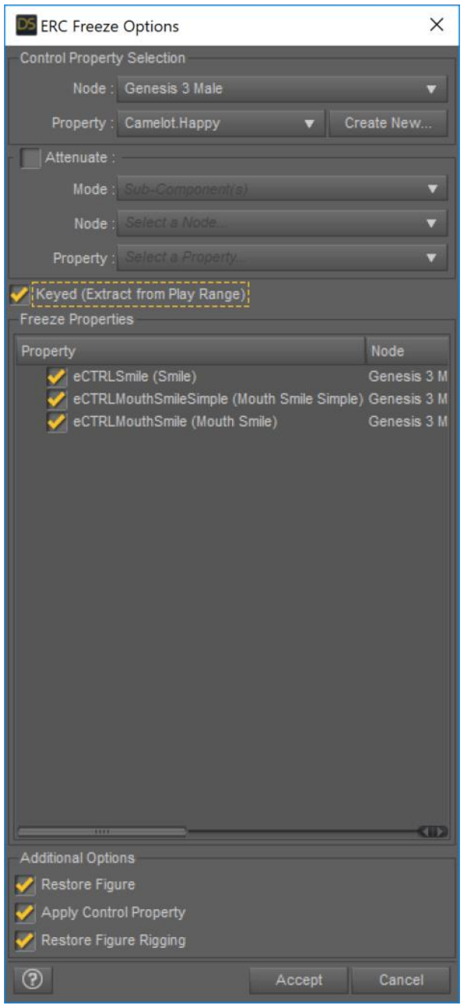
Property Type :	Float
Base Attributes	
Name :	Camelot.Happy
Label :	Camelot.Happy
Path :	Actor
Hidden :	No
Locked :	No
Animatable :	Yes
AutoFollow :	No
Presentation Attributes	
Type :	None
Color A :	41 41 41
Color B :	255 255 255
Icon :	None
Card :	None
Mapping Attributes	
Mappable :	No
Require Map :	No
Numeric Attributes	
Modulo :	No
Use Limits :	Yes
Float Attributes	
Default Value :	0.00
Value :	0.00
Min :	-1.00
Max :	1.00
Sensitivity :	0.040
As Percent :	Yes
Additional Options	

Create Cancel

Name the new property with some rule. We will use that rule when exporting to FBX file to filter out unnecessary properties. In our case, we use the rule Camelot.*: Camelot.Happy, Camelot.Sad

After that, in the timeline below, switch to frame 30, and change parameter of the model such that it shows the desired expression.




ERC Freeze Options

Control Property Selection
Node : Genesis 3 Male
Property : Camelot.Happy
Attenuate :
Mode : Sub-Component(s)
Node : Select a Node...
Property : Select a Property...
[✓] Keyed (Extract from Play Range)

Freeze Properties

Property	Node
[✓] eCTRLSmile (Smile)	Genesis 3 M
[✓] eCTRLMouthSmileSimple (Mouth Smile Simple)	Genesis 3 M
[✓] eCTRLMouthSmile (Mouth Smile)	Genesis 3 M

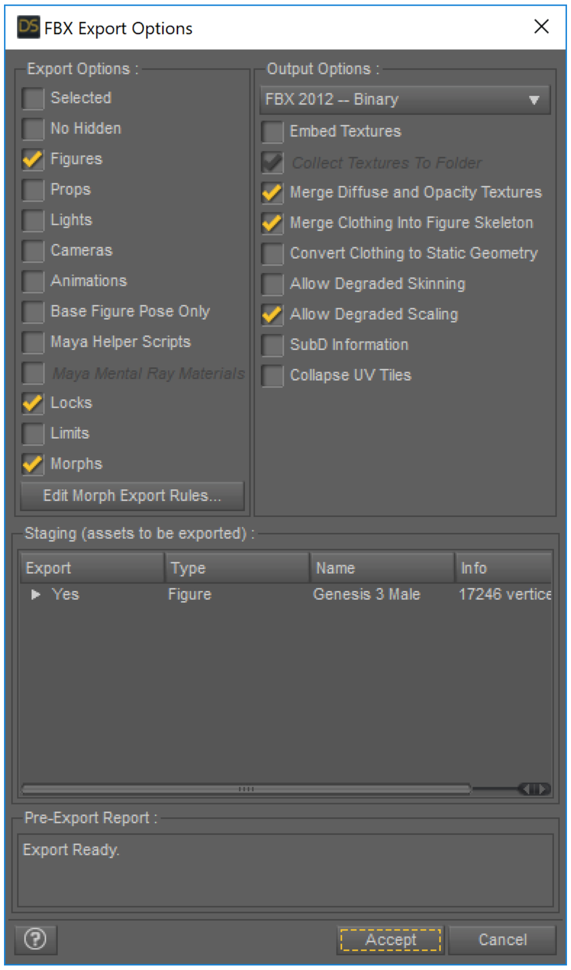
Additional Options
[✓] Restore Figure
[✓] Apply Control Property
[✓] Restore Figure Rigging

Next, search for our custom property (e.g. Camelot.Happy), right click on it and choose ERC Freeze and tick Key, then click Accept.

ERC Freeze is a feature of Daz Studio that packs many properties together to create a new property, or a blendshape property in unity, that is, it makes the chosen morphs values proportional to the new property's value. The max value of each property is the value it has at frame 30.

With this blendshape, the degree of expression (i.e how much happy one is) can be changed in Unity just as other property.

By this time, we have finished creating a new property representing a facial expression. Continue creating four more properties, add some clothes, hair and eye and we are ready to export to FBX file, which is compatible to Unity: File --> Export. After we enter file name, FBX Export Options shows up:


FBX Export Options

Export Options :
Selected
No Hidden
[✓] Figures
Props
Lights
Cameras
Animations
Base Figure Pose Only
Maya Helper Scripts
Maya Mental Ray Materials
[✓] Locks
Limits
[✓] Morphs
Edit Morph Export Rules...

Output Options :
FBX 2012 -- Binary
Embed Textures
[✓] Collect Textures To Folder
[✓] Merge Diffuse and Opacity Textures
[✓] Merge Clothing Into Figure Skeleton
Convert Clothing to Static Geometry
Allow Degraded Skinning
[✓] Allow Degraded Scaling
SubD Information
Collapse UV Tiles

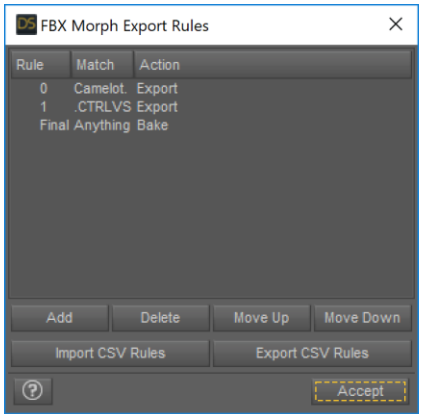
Staging (assets to be exported) :

Export	Type	Name	Info
Yes	Figure	Genesis 3 Male	17246 vertice

Pre-Export Report :
Export Ready.

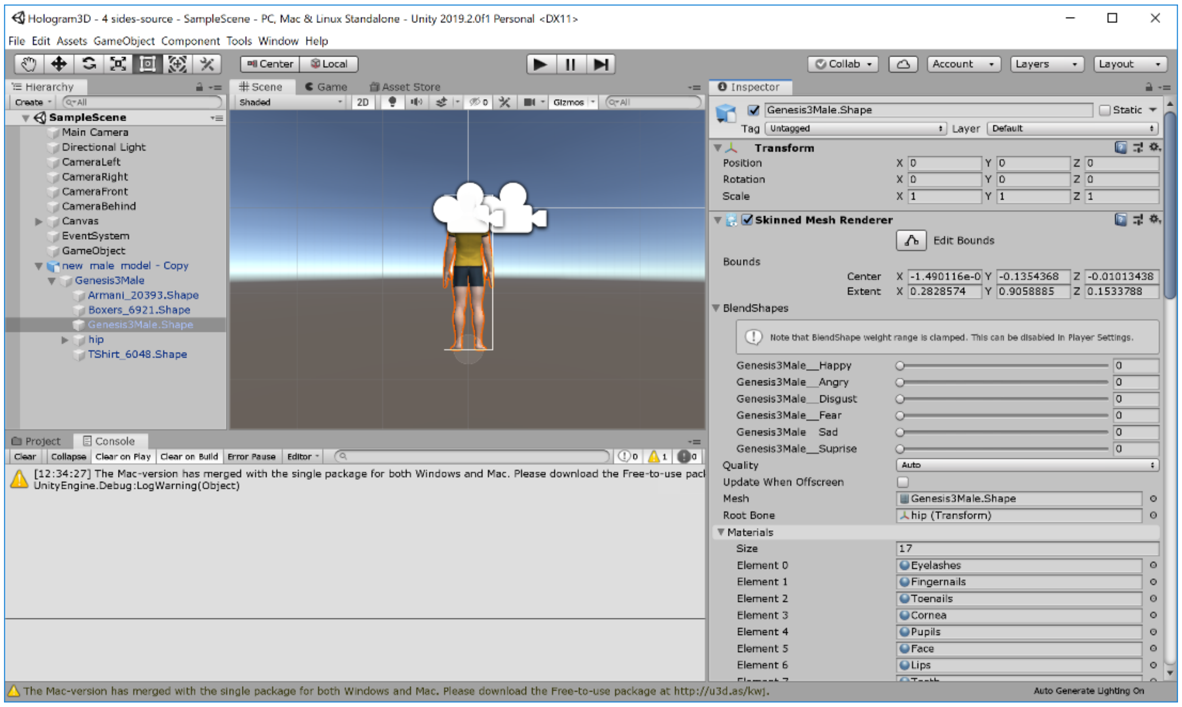
Tick on Morphs and click Edit Morph Export Rules:

Add new export rule :


FBX Morph Export Rules

Rule	Match	Action
0	Camelot.	Export
1	.CTRLVS	Export
Final	Anything	Bake

Click Accept to close FBX Morph Export Rules and Accept again to export to FBX file. The given FBX file, together with its texture folder, is ready for importing to Unity.



Hologram3D - 4 sides-source - SampleScene - PC, Mac & Linux Standalone - Unity 2019.2.0f1 Personal <DX11>

```
SampleScene
  Main Camera
  Directional Light
  CameraLeft
  CameraRight
  CameraFront
  CameraBehind
  Canvas
  EventSystem
  GameObject
  new male model - Copy
    Genesis3Male
      Armani_20393.Shape
      Boxers_6921.Shape
      Genesis3Male.Shape
      hip
      TShirt_6048.Shape
```

```
[12:34:27] The Mac-version has merged with the single package for both Windows and Mac. Please download the Free-to-use pacl
UnityEngine.Debug:LogWarning(Object)
```

Genesis3Male.Shape			
Tag	Untagged	Layer	Default

Transform

	X	Y	Z
Position	0	0	0
Rotation	0	0	0
Scale	1	1	1

Skinned Mesh Renderer

Bounds

	X	Y	Z
Center	-1.490116e-0	-0.1354368	-0.01013438
Extent	0.2828574	0.9058885	0.1533788

BlendShapes

Note that BlendShape weight range is clamped. This can be disabled in Player Settings.

Genesis3Male__Happy	0
Genesis3Male__Angry	0
Genesis3Male__Disgust	0
Genesis3Male__Fear	0
Genesis3Male Sad	0
Genesis3Male__Suprise	0
Quality	Auto
Update When Offscreen	
Mesh	Genesis3Male.Shape
Root Bone	hip (Transform)

Materials

Size	17
Element 0	Eyelashes
Element 1	Fingernails
Element 2	Toenails
Element 3	Cornea
Element 4	Pupils
Element 5	Face
Element 6	Lips

The Mac-version has merged with the single package for both Windows and Mac. Please download the Free-to-use package at http://u3d.as/kwj.

Auto Generate Lighting On

Hologram technique

Hologram is the combination of 2D image of a 3D object from three or four different viewpoints. By using a hologram lens, or hologram pyramid, those images will combine together to create visual feeling of 3D object.

In Unity, to get an image from a viewpoint, we use a camera object placed at that viewpoint. However, the key point is displaying three or four such image on the screen at special positions so that they can combine neatly. This can be done using a feature of Unity called Render Texture.

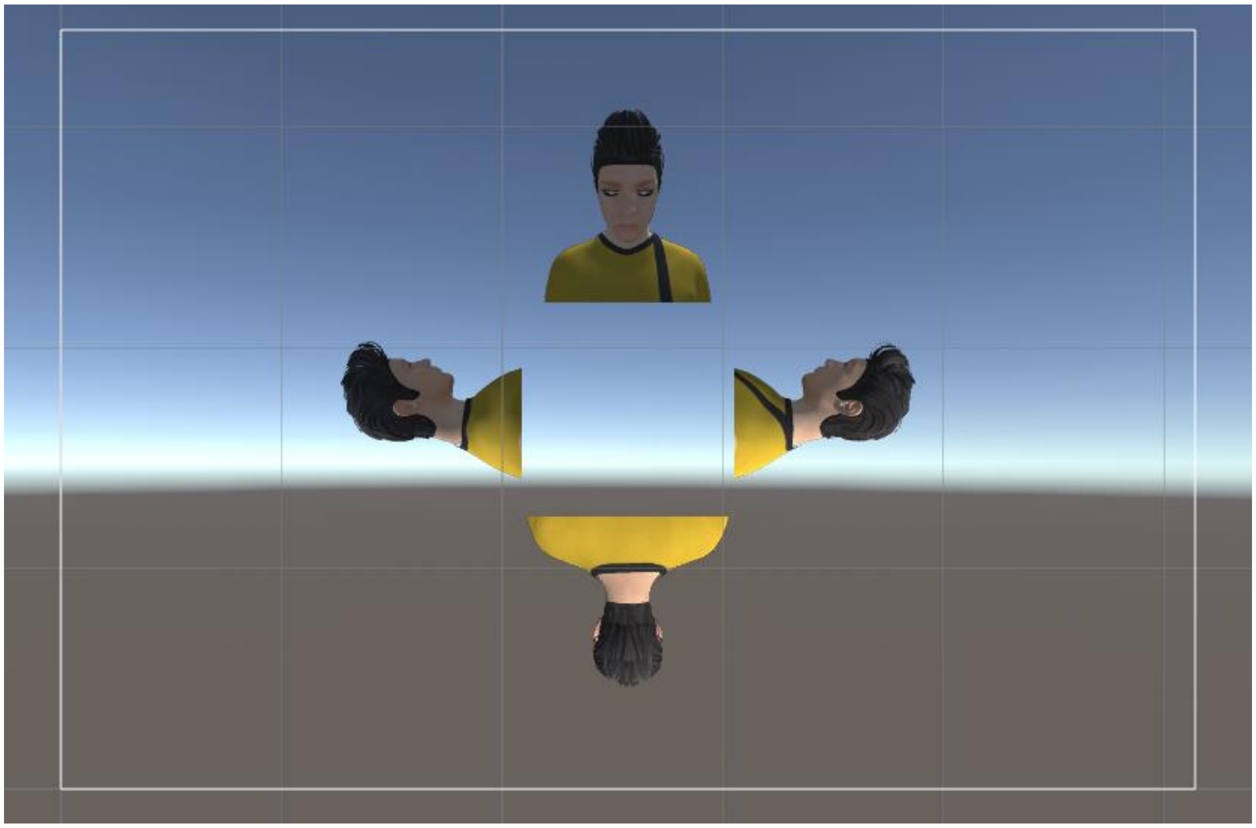
Render Texture, as its name suggest, is a texture that can applied to surfaces, sprites or UI elements. However, it is not pre-designed by designer, but rendered in real-time by camera. A

render texture is an image of what a camera is showing.

We will describe how to create 4-side hologram.

Create 4 camera, apart from the main camera, positioned at four position: left, right, front and back of the 3D object, all of which share the same distance to the object so that there is no difference in image size, because the farther from the 3D model, the smaller the image is. Set all camera background, including the main camera's, to black.

After that, set each camera to a render texture. To display 4 render texture on the screen, we use 4 raw images (UI element) on Canvas. Position the raw images such that their distance to center of the screen is uniform, and the rotation of them is as follows:



Finally, to hide the actual 3D model from view, set the main camera's culling mask to UI only.

For better hologram experience, we can animate the model by adding a script to rotate it over time (and also make sure the direction of the rotation is correct).

To delete a part of the image from one view (because there might be another view overlap this one), we can insert a black image with the desired shape (which we want to hide) on top of the texture using for that view. However, since we are building our application for 4-side hologram pyramid, we don't have to use this technique yet.

Expression Recognition

We adopt the trained model of Akash from https://github.com/akash720/Facial-expression-recognition

First, we use OpenCV and haar_cascade to detect faces in an image. After that, the face is cropped out of the image, resized and plugged into the pre-trained model. The process is repeated for every frame of the input video. For any frame with multiple faces detected, we decide to skip that frame.

This pretrained model has an estimated accuracy of 60% on fer2013 database in Kaggle Facial Expression Recognition challenge, the highest result of which is 71.161%.

How to run (test)

In the Github source, there are 2 folders. Hologram3D - 4 side is the unity client and Facial-expression-recognition-master contains python server.

First, the server need to be start, open file *facial-expression-recognition-master/ /facial-expression-recognition-master/trained models/run.py* ,install necessary packages (keras, numpy, opencv, etc) and change path to test video.

After that, run the file run.py, waits for a moment until the 'listening' message appear on the console. At this time, the server is successfully started and result of facial expression in the video is ready to be streamed to client.

Then run Hologram 3D or Hologram 3D -4 side in Unity, the server will try to connect to server and display expression on 3D avatar model

Demo video:

https://drive.google.com/file/d/1vks0Vk4MsuNIOAZxZqVFIAgiPm-U1L6D/view?fbclid=IwAR0Sswp2cr4q1cU76fEjbfD3GU4RdqYU7wR3EGcj49aq7x17dNMB12kmHXI

References:

Hologram technique: Mr. Trần Ngọc Đạt Thành’s training (tndthanh@selab.hcmus.edu.vn)

Input video for demo and testing:

https://www.youtube.com/watch?v=2NtMobMvKrc

And self-recorded videos

Pretrained model for facial expression recognition:

https://github.com/akash720/Facial-expression-recognition

Kaggle Facial Expression Recognition challenge

https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data

3D model by Daz Studio:

https://www.youtube.com/watch?v=jIdNZsEJK-U

TCP socket guide - python:
https://docs.python.org/2/library/socket.html

C# socket guide:
https://docs.microsoft.com/en-us/dotnet/api/system.net.sockets.tcpclient?view=netframework-4.8

TCP - socket client basics:
https://www.geeksforgeeks.org/tcp-server-client-implementation-in-c/

MVC model - image
https://upload.wikimedia.org/wikipedia/commons/thumb/a/a0/MVC-Process.svg/1200px-MVC-Process.svg.png

Supporting tools:

Unity:	https://unity.com
Daz Studio:	https://www.daz3d.com