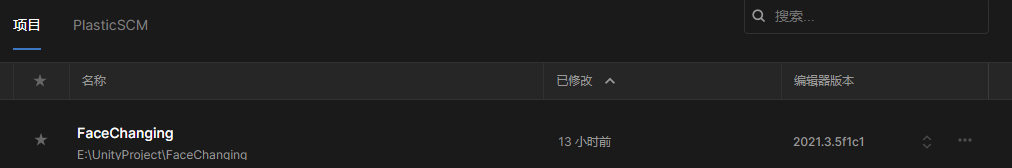
# Experiment Five: FaceChanging – AR Peking Opera Face Swapsummary

The purpose of this experiment was to develop a Peking Opera face-swapping application using Unity's Augmented Reality (AR) capabilities, which would apply a virtual Peking Opera mask or makeup to the user's face. Through experiments, I gained an in-depth understanding of Unity's AR capabilities and related face recognition and interaction technologies, and successfully implemented a simple Peking Opera face swapping application.

# Experimental Procedure

## Experiment preparation

软件：Unity+Xdreamer



Xdreamer

Xdreamer is somewhat similar to Unreal Engine's Blueprint functionality and is an extensible Chinese interactive editing software in Unity.

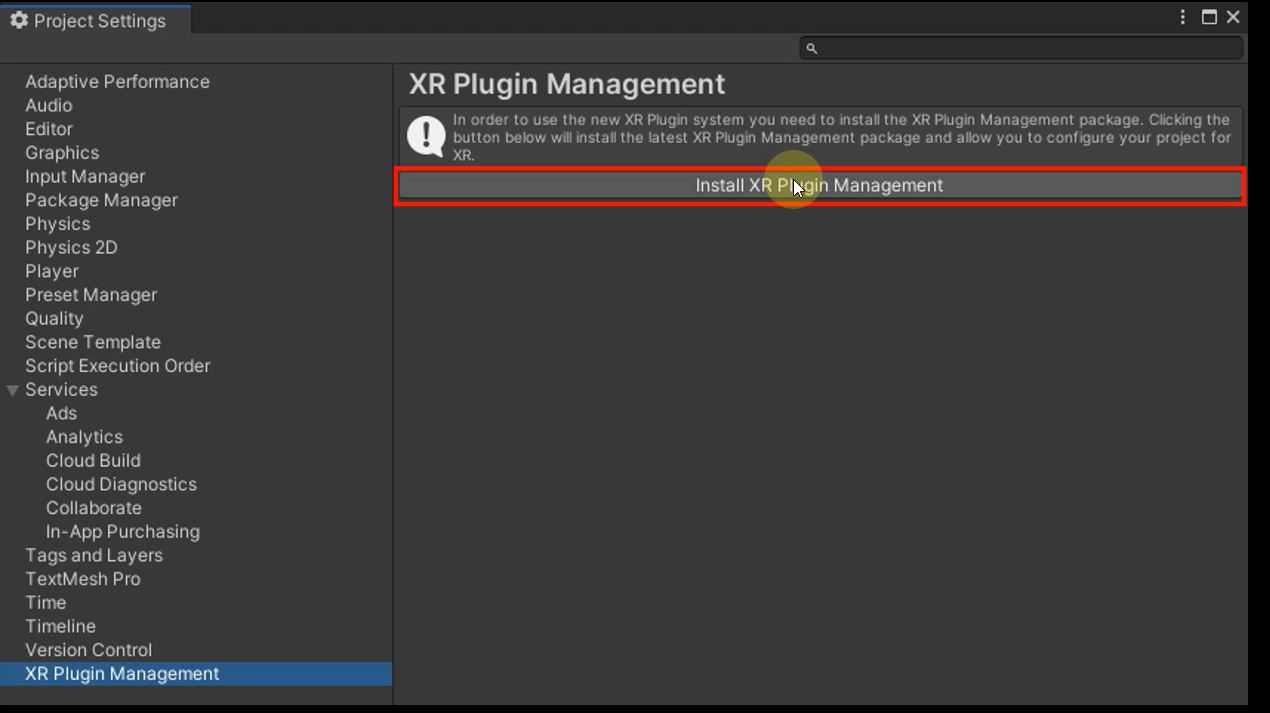


## 2 Experimental Procedure

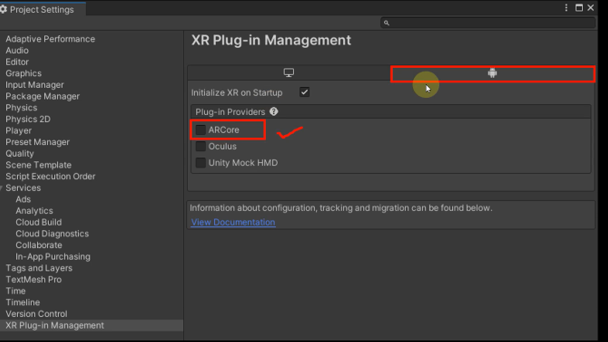
### 2.1 Create a new project and configure the experimental environment

First, create a project, name it FaceChanging, note that there can be no Chinese in the project path.

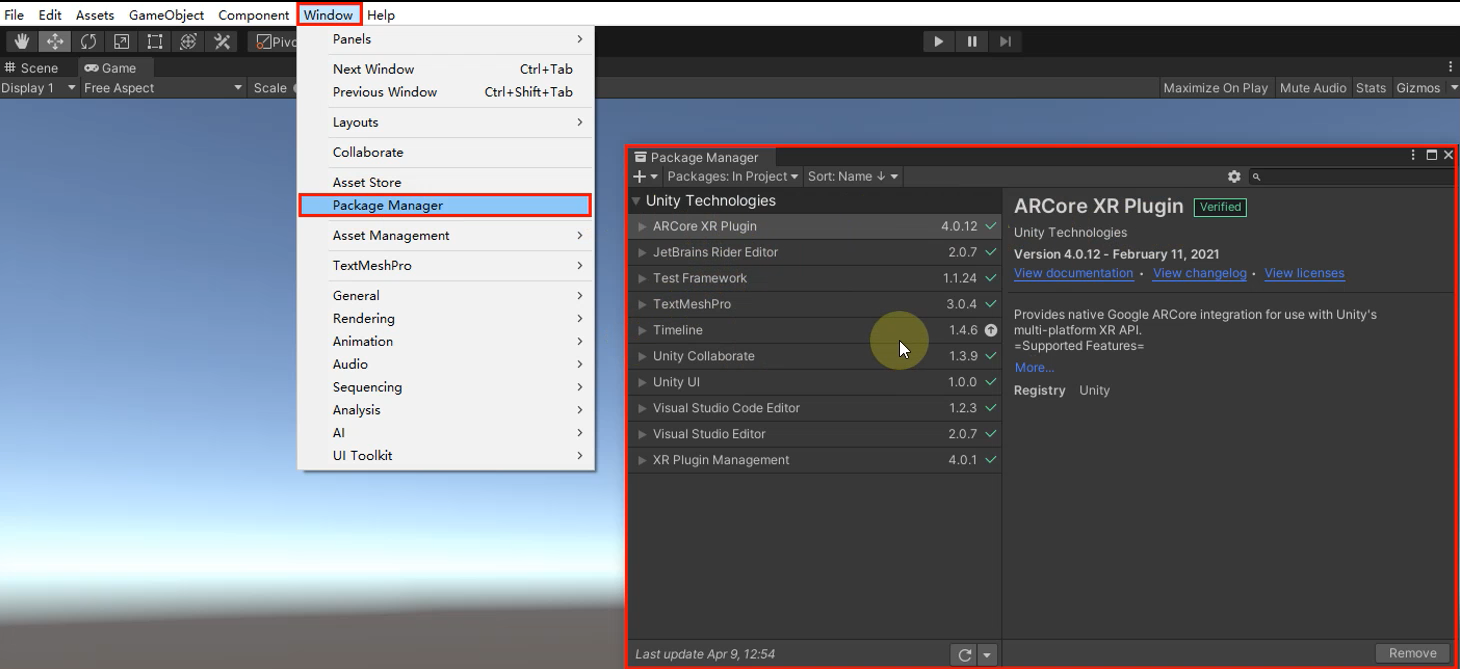
Install the XR plugin



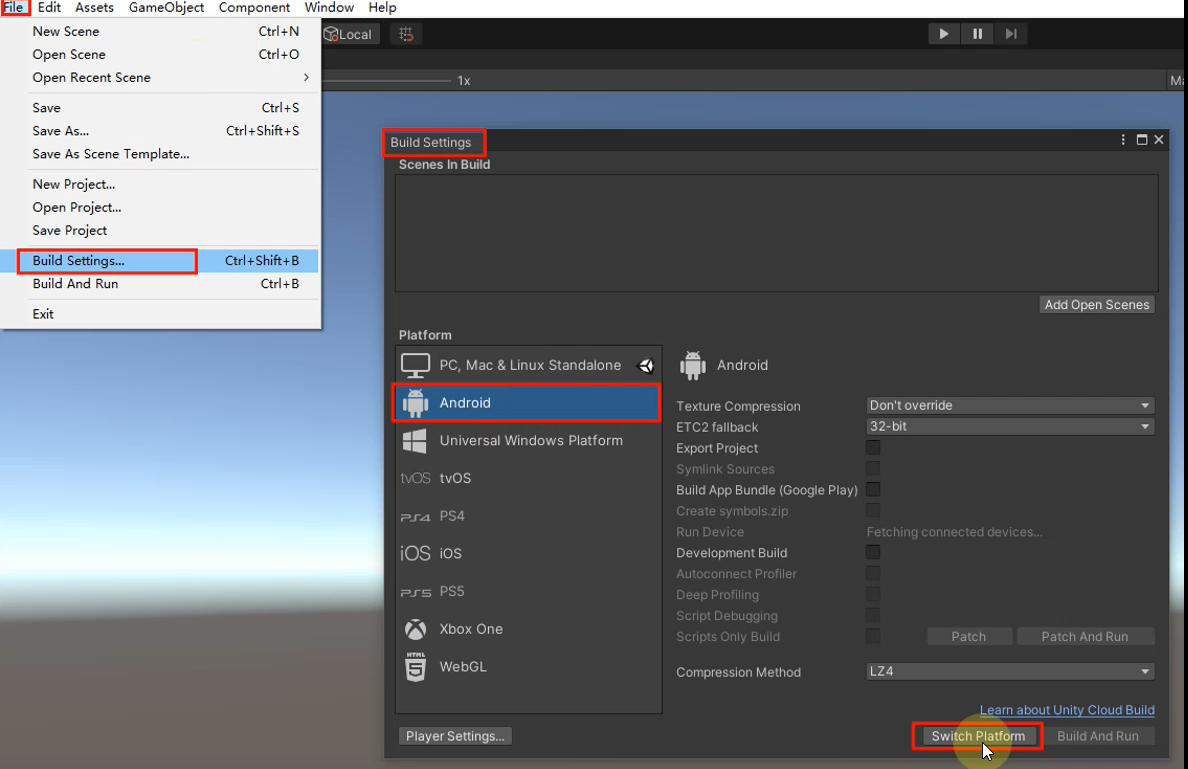
Enable ARcore in the Android options



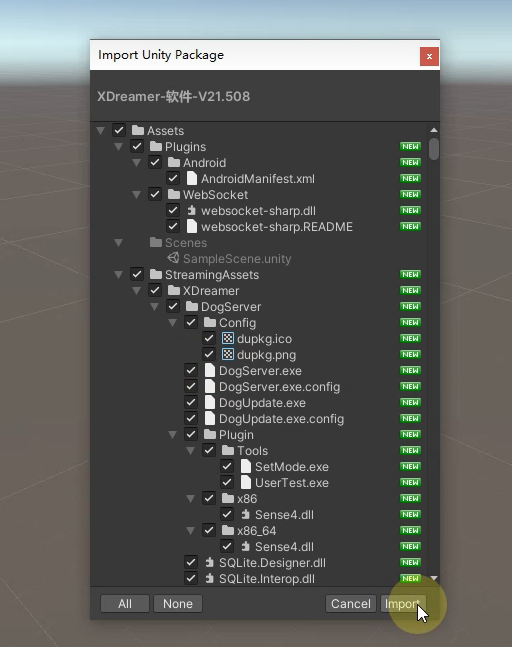
Load the ARFoundation plugin in the package manager



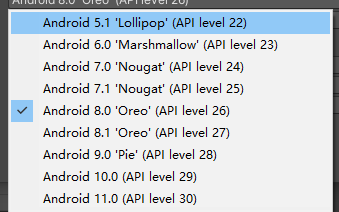
Convert the project to the Android platform

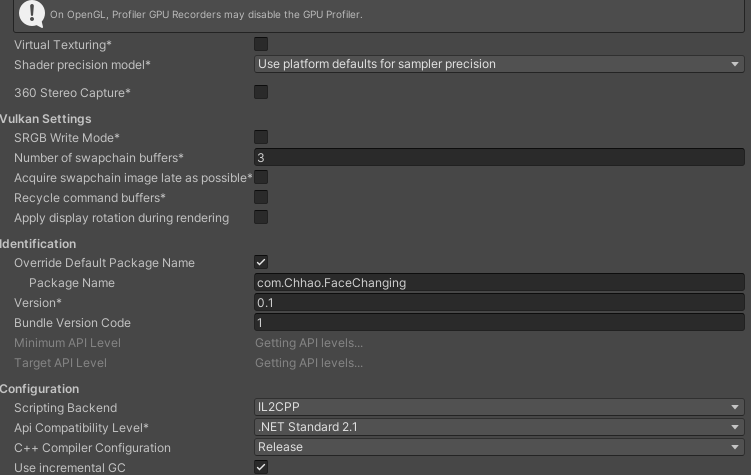


Import the Xdreamer package



Finally, let's set the basic information of the project, and note that the API level of Android should be set to 7.0 or above.

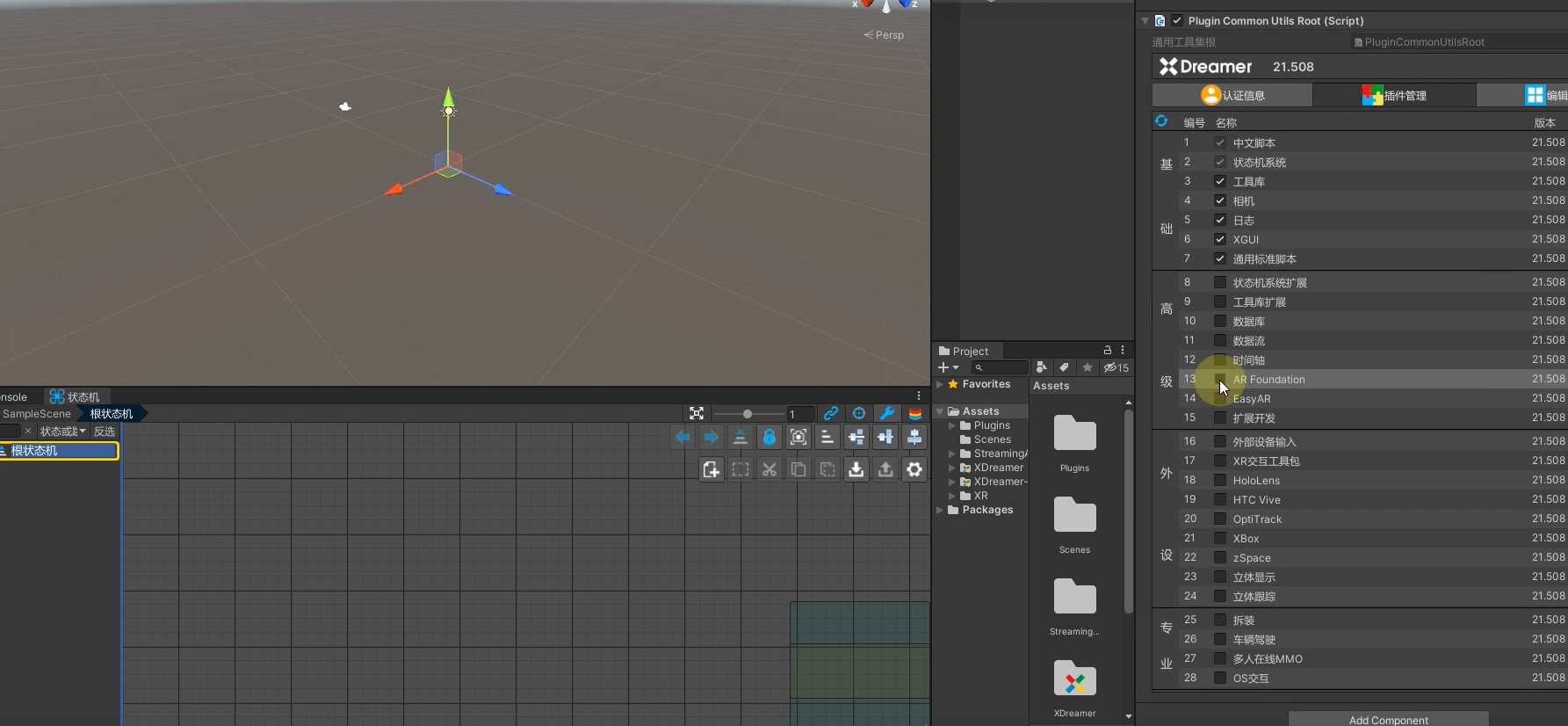




Setting up the lab environment at the very beginning of the project is not only quick to configure. And it can save a lot of trouble in the later stage.

### 2.2 The realization of the core function of face changing

First, create a state machine and check ArFoundation



Open the Tools Library and the State Library



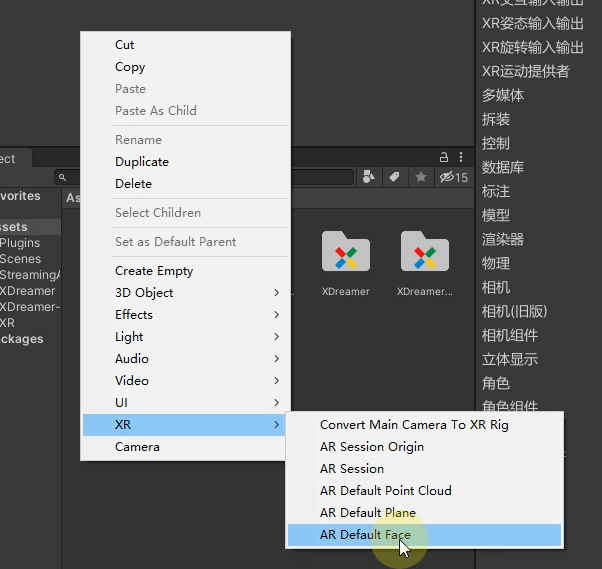


Select ArFoundation's face tracker and you can see that there is an automatic extra column on the right side

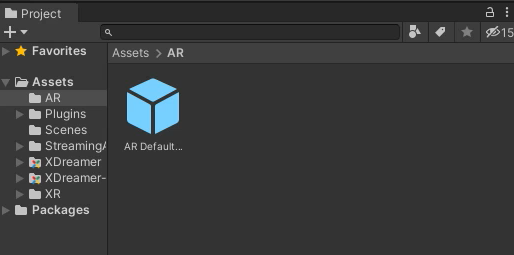




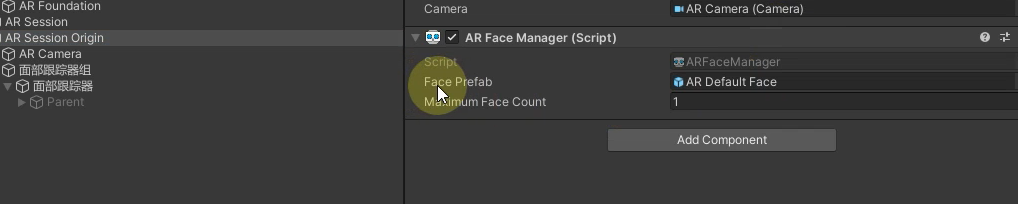
After that, a model of the face mesh is created



Turn this model into a prefab



Drag the prefab to the appropriate position.

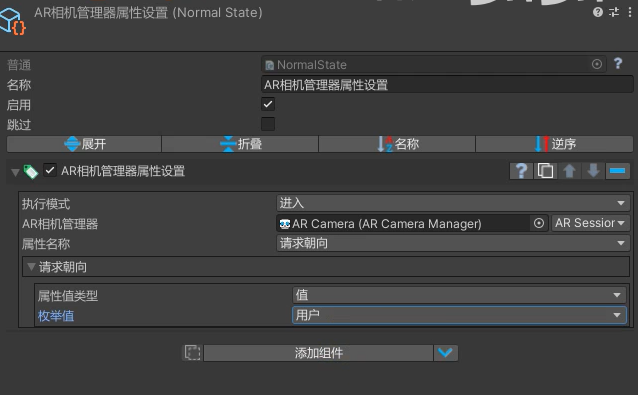


State machine logic-1

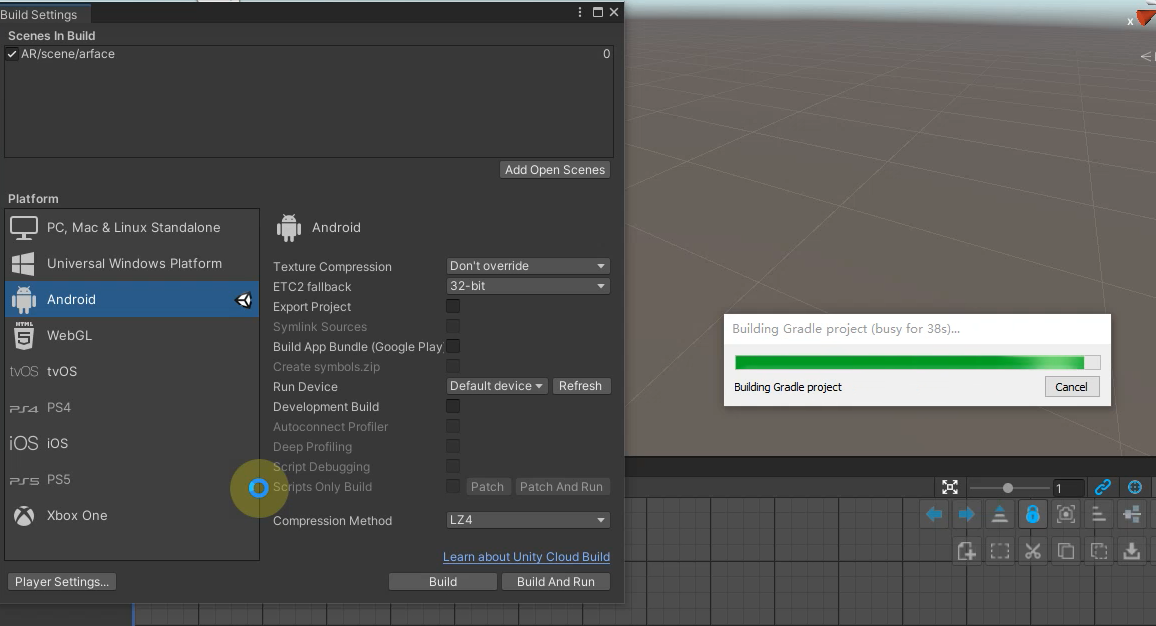


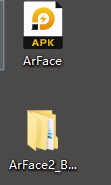
With the state machine, the user automatically turns on the front-facing camera when using the software. Add "AR Camera Manager Property Settings" to the state machine





After that, the first packaging test of the project is carried out.

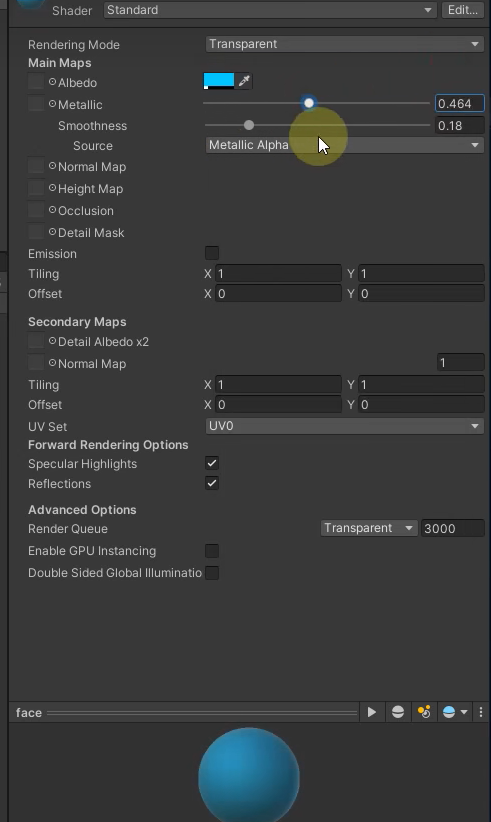




Check the running result.

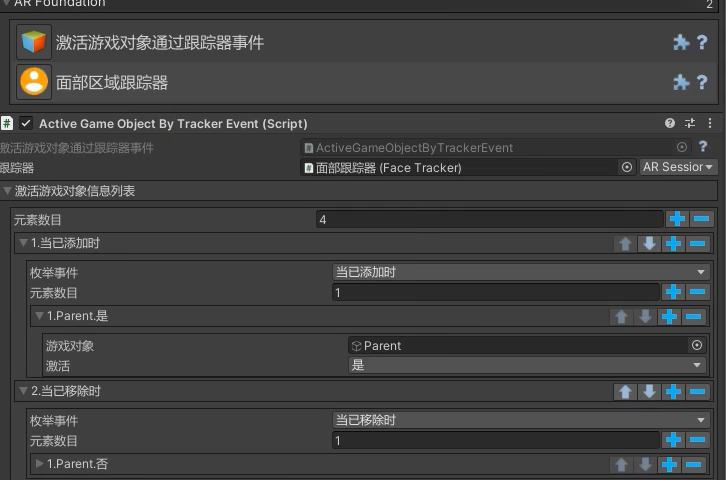
  

The result is no problem, but the model of the face is a little too green, adjust the material of the face model to a white transparent material.



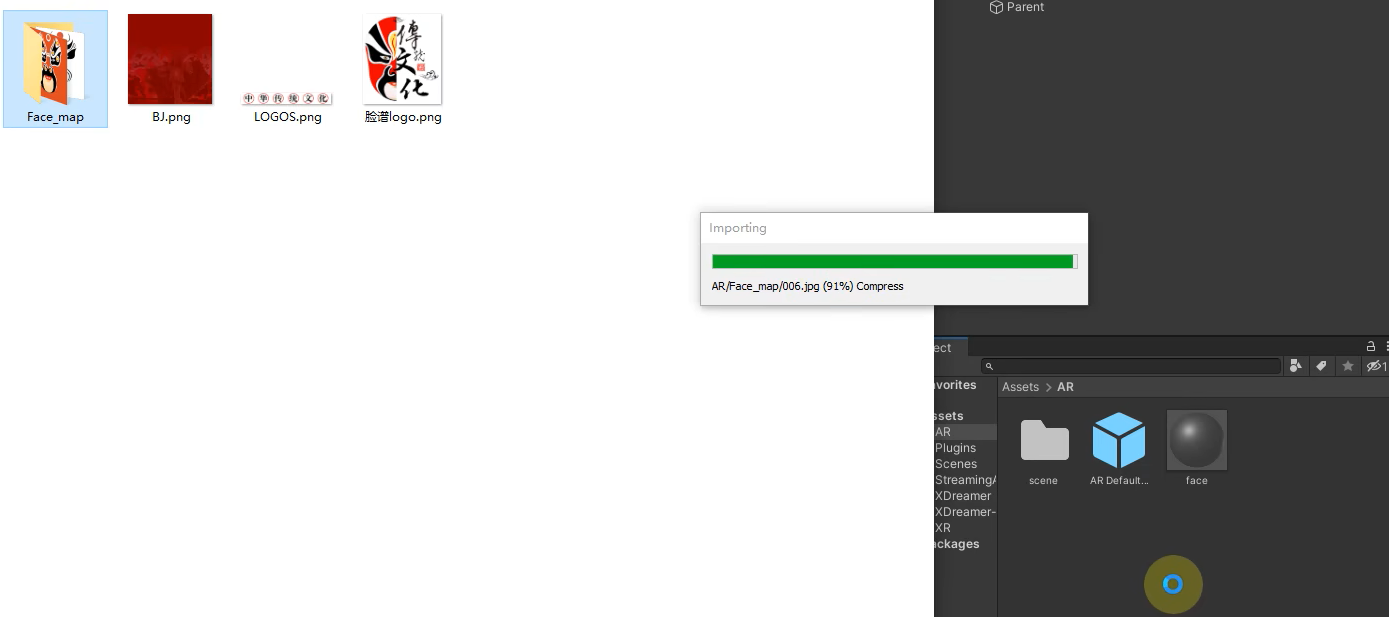
At the same time, the axis of the face is hidden.

The core logic of face change is achieved with the facial area tracker

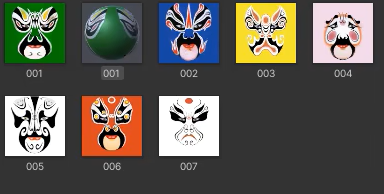


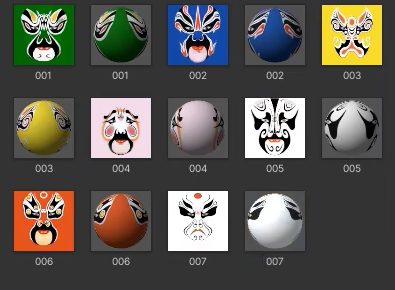
Modify the face model material using transforms that are obscured and unoccluded.

Import footage



Use seven Facebook images to make a shade ball





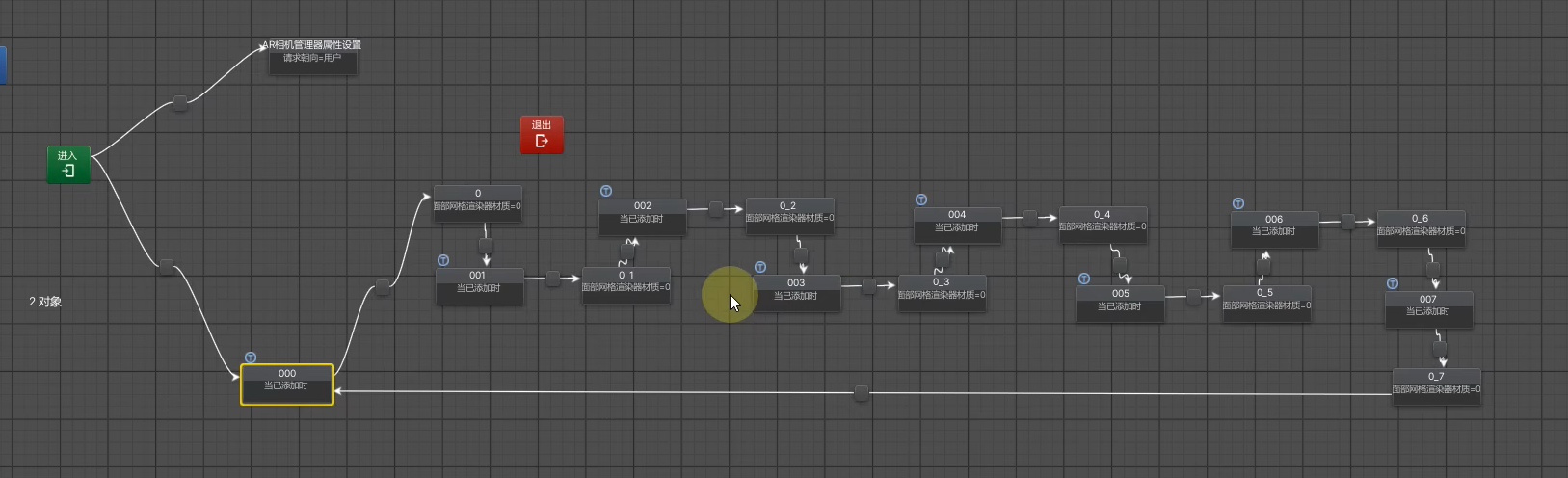
State machine logic-2



Use the Face Detector to detect face swapping and the Face Tracker property settings to swap materials



Realize the function of face swapping

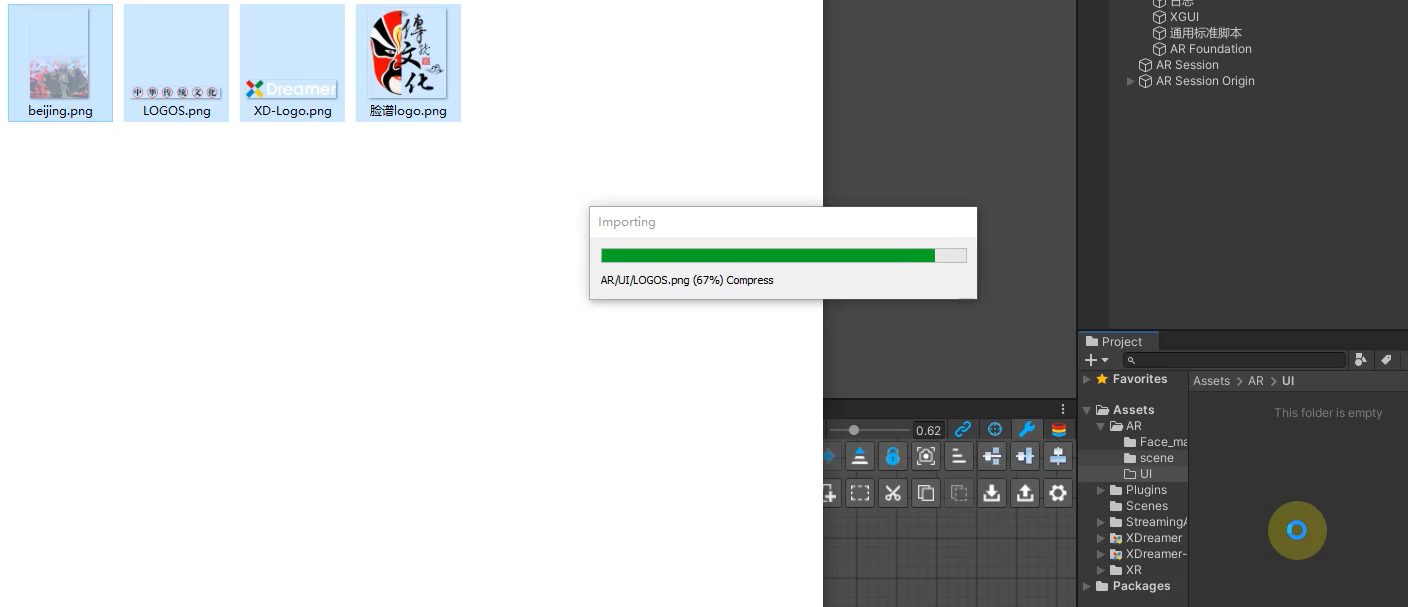


After that, a second packaging test was carried out, and the results were as follows.



### 2.3 Make a UI interface, package and release the APP

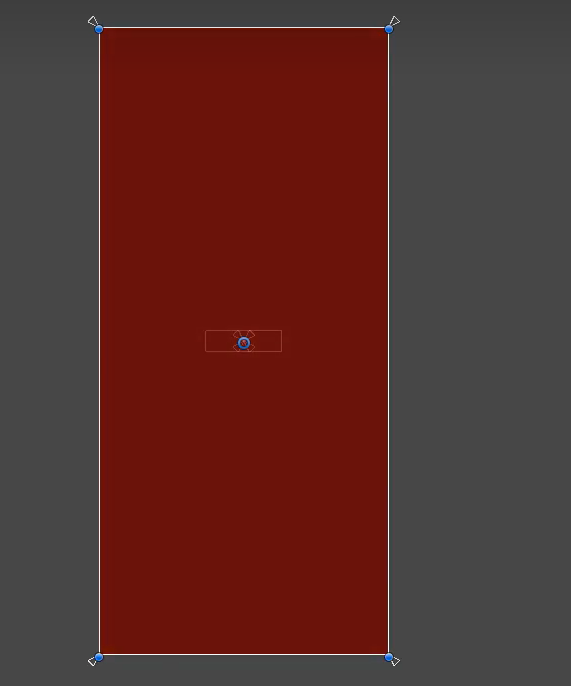
Import UI assets



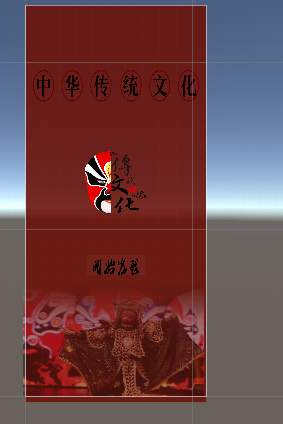
The Create button also modifies the canvas resolution



Set the background image

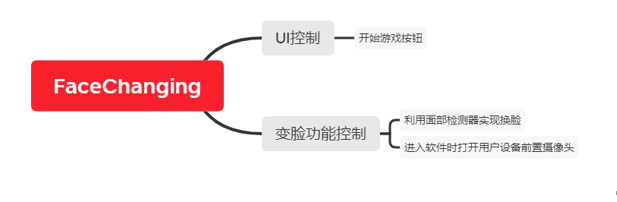


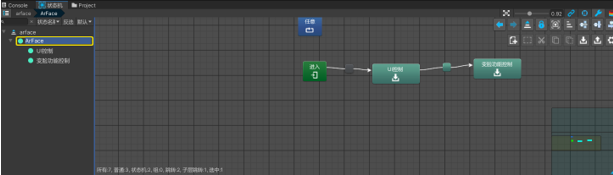
Add a basemap and logo



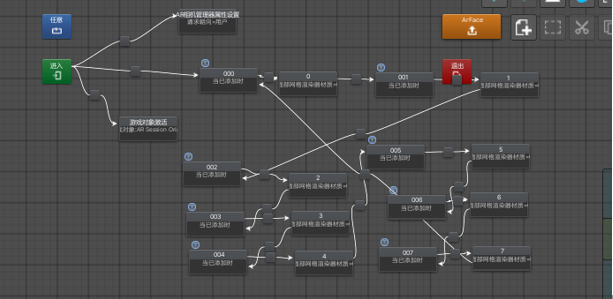
Set up the state machine of the UI

State machine-3





A switch has been added to the face change control, and the user will only enter the interface after clicking to start the game



Finally, the final version is packaged, and the experimental results are as follows:



# 3. Experimental experience

This experiment and this semester's course have brought me a very valuable learning experience.

I learned how to use AR features in Unity, including getting camera input, real-time image processing, and the application of virtual objects. This provided the foundation for me to understand and master augmented reality development.

At the same time, I improved the user experience by adding interactive features to the app, giving the option to choose different masks or makeup. This gave me a deeper understanding of designing and developing user-friendly AR applications.

During the experiment, I also encountered various problems and challenges, such as environment configuration, packaging errors, positioning of virtual objects, etc. By solving these problems, I learned the ability to debug code and find solutions.

Through this experiment, I not only learned about Unity's AR features and facial recognition technology, but also gained experience in solving problems and debugging code. This knowledge and experience will provide me with a valuable foundation and guidance for future AR development projects. At the same time, I also deepened my understanding and application of augmented reality technology through practical operations and experimental processes. I hope I will keep learning and practicing in the days to come, and continue to explore the field of VR and AR.