



BÁO CÁO

MÔN CÁC CÔNG NGHỆ MỚI TRONG
PHÁT TRIỂN PHẦN MỀM

ARFoundation

Phạm Minh Hoàng - 1712460



MỤC LỤC

I.	Giới thiệu ARFoundation:	3
II.	Sử dụng Unity với XCode:	3
III.	Các class, interface phổ biến mà framework hỗ trợ.....	5
IV.	Các component phổ biến mà framework hỗ trợ	6



I. Giới thiệu ARFoundation:

ARFoundation là một cross-platform framework cho phép chúng ta xây dựng các nền tảng AR, có thể build được trên thiết bị Android hay iOS. AR Foundation không implement bất kỳ tính năng AR nào nhưng nó define API multi-platform cho phép chúng ta làm việc với chức năng chung cho nhiều nền tảng. Và để sử dụng AR Foundation trên loại thiết bị mong muốn, chúng ta cũng cần các packet riêng biệt cho các nền tảng tương ứng như:

- ARCore XR Plugin trên Android
- ARKit XR Plugin trên iOS
- Magic Leap XR Plugin trên Magic Leap
- Windows XR Plugin trên HoloLens

AR Foundation là một tập hợp các MonoBehaviour, APIs để xử lý các thiết bị hỗ trợ các ý tưởng như: Theo dõi thiết bị, phát hiện mặt phẳng, ước tính ánh sáng, theo dõi khuôn mặt, theo dõi hình ảnh 2D, theo dõi đối tượng 3D, chia lưới, ...

Hệ thống con (Subsystems)

AR Foundation được xây dựng trên các hệ thống con (subsystem). Các hệ thống con liên quan đến AR được define trong packet AR Subsystems và sử dụng namespace UnityEngine.XR.ARSubsystems. Đôi khi chúng ta sẽ cần phải tương tác với các loại trong packet Hệ thống con AR. Mỗi hệ thống con xử lý chức năng cụ thể.

Nhà cung cấp (providers)

Nhà cung cấp là một triển khai cụ thể của một hệ thống con. Ví dụ: ARCore XR Plugin packet chứa implementation ARCore cho nhiều hệ thống con AR.

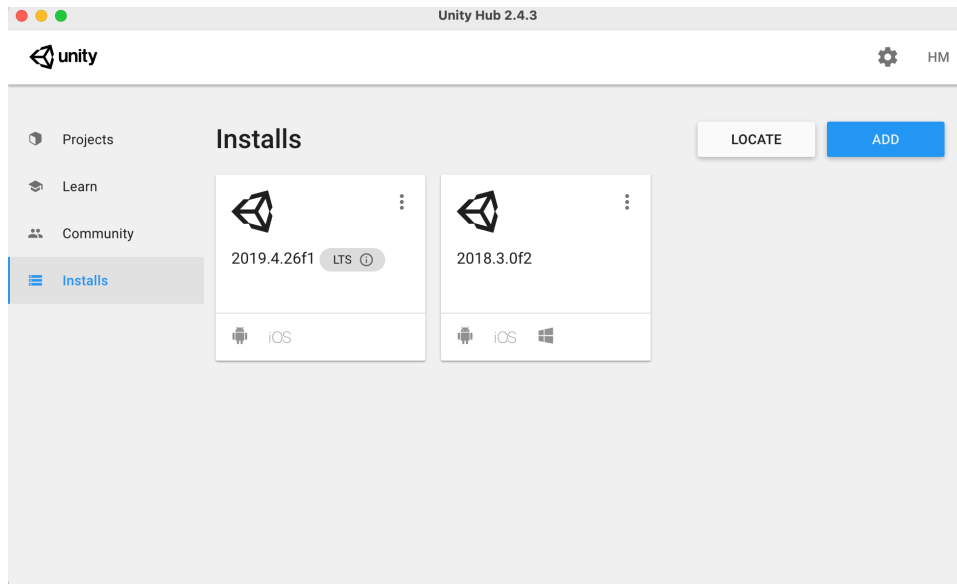
Vì các nhà cung cấp khác nhau có hỗ trợ khác nhau cho các tính năng cụ thể, mỗi hệ thống con cũng có một bộ mô tả cho biết nó hỗ trợ các tính năng hệ thống con cụ thể nào. Ví dụ, các properties XRPlaneSubsystemDescriptor chứa cho biết liệu nó có hỗ trợ phát hiện mặt phẳng ngang hay dọc hay không.

Mỗi nhà cung cấp riêng lẻ xác định cách triển khai từng hệ thống con. Nói chung, họ đóng gói SDK gốc của nền tảng đó (ví dụ: ARKit trên iOS và ARCore trên Android).

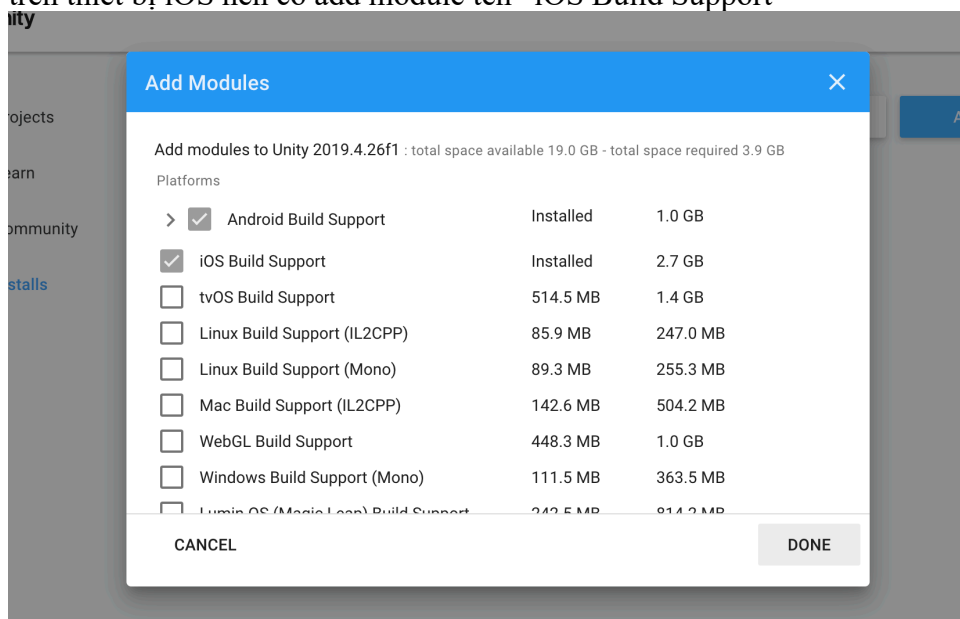
II. Sử dụng Unity với XCode:

Để sử dụng Unity xây dựng một app, cụ thể ở đây là app dành cho iOS ta làm như sau:

- 1) Download Unity Hub về máy
- 2) Mở Unity Hub lên, cài đặt bản Unity mà chúng ta mong muốn

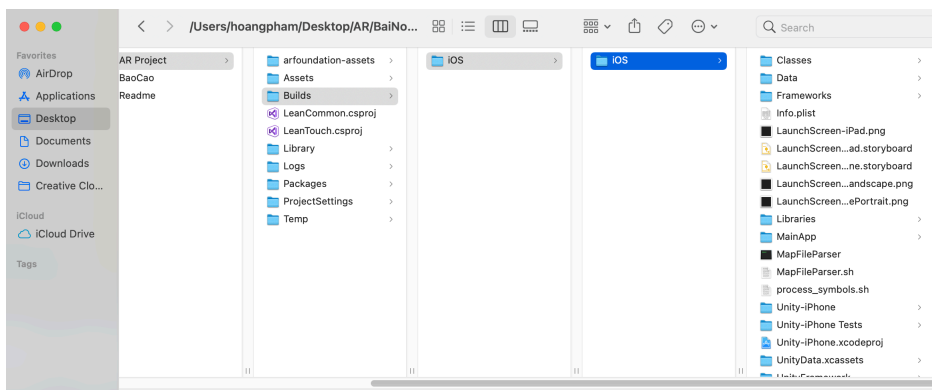


- 3) Add modules mà chúng ta mong muốn lên phiên bản Unity, ở đây ta mong muốn build trên thiết bị iOS nên có add module tên “iOS Build Support”

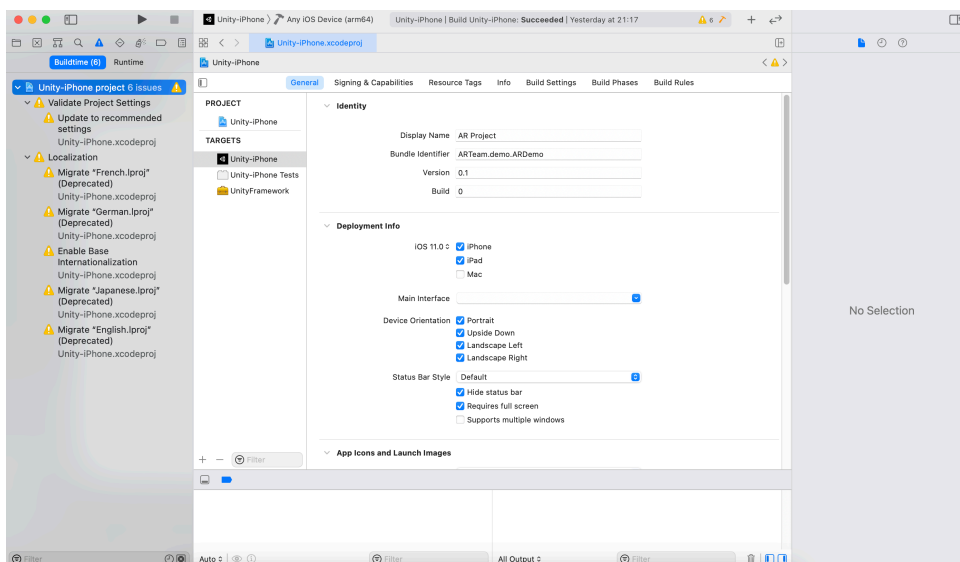


- 4) Tạo một project, thêm bớt xóa sửa các chức năng mà chúng ta mong muốn (Lưu ý: ta phải cài các packets cho project như ARFoundation, ARKit XR Plugin trong phần packet manager, cũng như phải config một vài thứ trong player setting như bundle identifier, requires ARKit support, target minimum iOS version,...)

- 5) Sau có được các chức năng ta mong muốn, chúng ta chọn Build and Run, chọn thư mục build, đây là thành quả sau khi Unity build xong (trong đó có file .xcodeproj)



- 6) Sau đó project của chúng ta sẽ được tự động bật lên một project XCode, chúng ta kết nối thiết bị phù hợp sau đó nhấn Build để XCode build, cài đặt app trên thiết bị rồi tự động chạy app mà chúng ta đã cài đặt.



III. Các class, interface phổ biến mà framework hỗ trợ

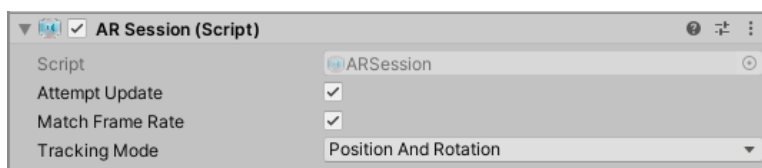
- ARController: Camera listener lấy tất cả lifecycle của event từ ARCamera.
- Device: Đại diện (trừu tượng) cho thiết bị (device) đang chạy
- IDisposable: chứa IUnityPlayer
- INotifyPropertyChanged
- ISessionRecorder: Trình ghi phiên (session) cho phép người dùng ghi lại các phiên để phát lại trong trình chỉnh sửa và trên thiết bị
- ITrackerManager: chứa class TrackerManager (Lớp này đóng vai trò như một singleton để truy xuất Trackers và StateManager)
- IUnityARFoundationFacade: Cung cấp quyền truy cập vào chức năng AR Foundation
- StateManager: Lớp này được sử dụng để quản lý trạng thái của tất cả các TrackableBehaviours, tạo chúng, liên kết chúng với Trackable, cập nhật tư thế (pose) của chúng, v.v.

- Trackable: Lớp cơ sở của các loại trackable, có chứa class ObjectTarget (gồm CylinderTarget, ModelTarget, ImageTarget,...)
- UnityEvent: Chứa các class con như (ContentPlacedEvent, HitTestEvent, SessionRecorderBehaviour.StartRecordingEvent)
- ...

IV. Các component phổ biến mà framework hỗ trợ

ARSession

ARSession kiểm soát vòng đời của một phiên AR. Phiên AR kiểm soát vòng đời của trải nghiệm AR bằng cách bật hoặc tắt AR trên nền tảng đích. Nó liên tục nhận các bản cập nhật (updates) từ hệ thống con (subsystem) để kiểm tra xem phiên (session) có còn (alive) hay không.



Khi chúng ta tắt ARSession, hệ thống không còn theo dõi các tính năng trong môi trường của nó, nhưng nếu sau đó chúng ta bật nó, hệ thống sẽ cố gắng khôi phục và duy trì các tính năng đã phát hiện trước đó.

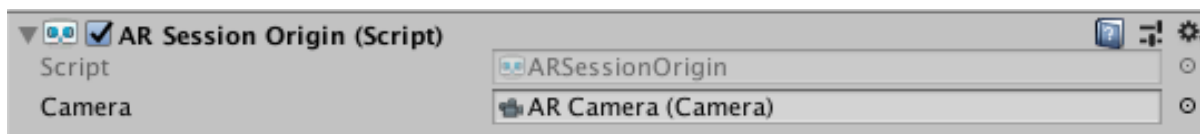
Trạng thái phiên (session state):

Để xác định trạng thái hiện tại của phiên (ví dụ: thiết bị có được hỗ trợ không, phần mềm AR có đang được cài đặt hay không và phiên có hoạt động hay không), ta hãy sử dụng ARSession.state. Chúng ta cũng có thể đăng ký một sự kiện khi trạng thái phiên thay đổi: ARSession.stateChanged. Một vài state điển hình như: None: Hệ thống AR chưa được khởi chạy và chưa xác định được tính khả dụng, Unsupported: Thiết bị hiện tại không hỗ trợ AR, Ready: AR được hỗ trợ và sẵn sàng, ...

AR Session Origin

ARSessionOrigin giữ các đối tượng ảo của bạn ở đúng vị trí trong môi trường AR.

Nó tiếp tục nhận các bản cập nhật từ hệ thống con và sửa đổi vị trí của các đối tượng ảo để luôn phù hợp với môi trường thực. Lưu ý rằng chúng ta cần đặt các đối tượng AR của mình dưới ARSessionOrigin trong Hệ thống phân cấp (Hierarchy). Nếu chúng ta không làm vậy thì vị trí của các đối tượng sẽ không cập nhật.

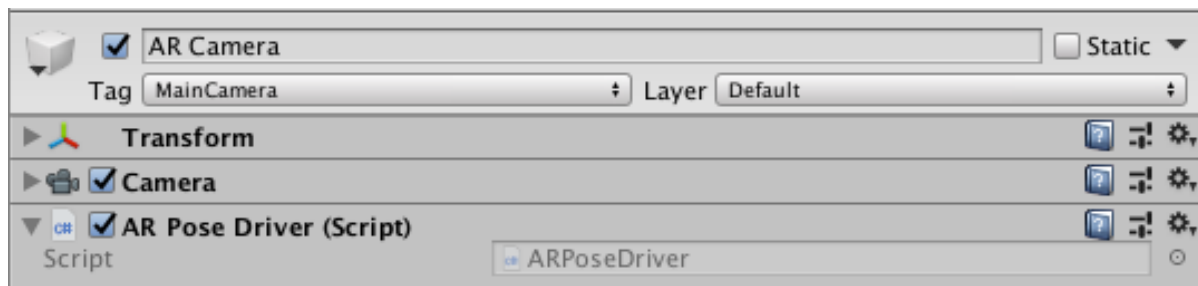


Mục đích của mục đích ARSessionOrigin là chuyển đổi các đối tượng địa lý có thể theo dõi, chẳng hạn như bề mặt phẳng và điểm đặc trưng, thành vị trí, hướng và tỷ lệ cuối cùng của chúng trong Unity Scene. Vì thiết bị AR cung cấp dữ liệu của chúng là không gian chưa được chia tỷ lệ so với đầu phiên AR, nên thiết bị ARSessionOrigin thực hiện chuyển đổi thích hợp thành không gian Unity.

ARSessionOrigin cũng cho phép chúng ta chia tỷ lệ nội dung ảo và áp dụng độ lệch cho AR Camera. Nếu bạn đang điều chỉnh tỷ lệ hoặc bù trừ ARSessionOrigin, thì Camera AR của nó phải là con của ARSessionOrigin. Vì Máy ảnh AR được điều khiển theo phiên, thiết lập này cho phép Máy ảnh AR và các bảng theo dõi được phát hiện di chuyển cùng nhau.

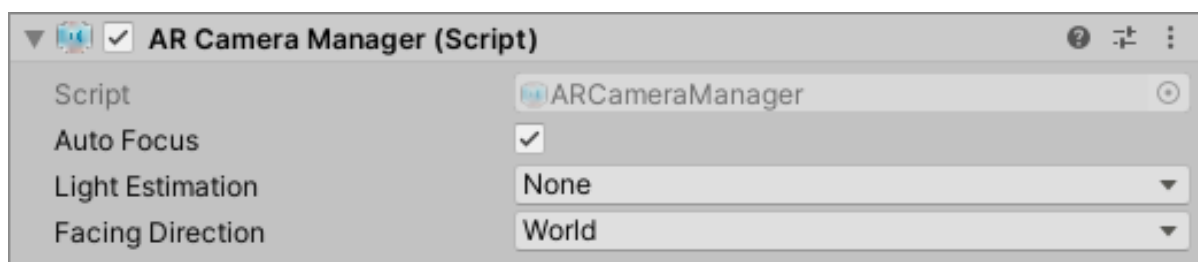
AR Pose Driver

ARPoseDriver điều khiển vị trí cục bộ và hướng của GameObject chính theo thông tin theo dõi của thiết bị. Trường hợp sử dụng phổ biến nhất là gắn ARPoseDriver vào Máy ảnh AR để điều khiển vị trí và hướng của máy ảnh trong một cảnh AR.



AR Camera manager

ARCameraManager cho phép các tính năng cho Máy ảnh AR, bao gồm quản lý kết cấu máy ảnh của thiết bị và các thuộc tính đặt chế độ đo ánh sáng.



AR Camera background

Nếu chúng ta muốn lấy video từ máy ảnh của thiết bị hiển thị dưới dạng nền cảnh trong thời gian chạy, chúng ta cần thêm một component ARCameraBackground vào Máy ảnh. Nếu không, nền trong thời gian chạy sẽ đến từ Camera.clearFlags. ARCameraBackground đăng ký các sự kiện Máy ảnh AR và hiển thị kết cấu Máy ảnh AR lên màn hình (nghĩa là, kết cấu nền



từ máy ảnh thiết bị phải được hiển thị cho mỗi khung hình). Việc này không bắt buộc, nhưng phổ biến đối với các ứng dụng AR.



Nếu chúng ta đã có một ARSessionOrigin, chúng ta chỉ cần thêm ARCameraBackground vào Máy ảnh đó. Nếu bạn có nhiều ARSessionOrigin (ví dụ: để hiển thị có chọn lọc các nội dung khác nhau ở các tỷ lệ khác nhau), chúng ta nên sử dụng các Máy ảnh riêng biệt cho từng cái ARSessionOrigin và một Máy ảnh AR riêng biệt cho ARCameraBackground.

(Truy cập hình ảnh máy ảnh trên CPU)

AR input manager

Thành phần này là bắt buộc để kích hoạt tính năng theo dõi (tracking). Nếu không có nó, Tracked Pose Driver không thể có được tư thế (pose) cho thiết bị.

Thành phần này có thể ở bất kỳ đâu trong Scenes, nhưng chúng ta không nên có nhiều hơn một cái.

