UE5で作ったVR作品を
Meta Quest実機で動かすまで

初心者向け11Stepガイド

「UE5でVR作品を作ったけれど、Questでどうやって動かせばいいの？」
そんな人向けに、この資料では
UE5で作ったVR作品をパッケージ化して、自分のMeta Questで起動するまで
を11Stepで解説します。
難しい機能やストア公開については扱いません。
まずは、
Quest本体で起動し、VR空間を見回せるようにすること
をゴールにしましょう!

この資料の対象環境

・PC：Windows 11
・Unreal Engine：UE5.7
・ヘッドセット：Meta Quest 3S
・使用するプラグイン：UE5標準のOpenXR

本資料では、VR Templateなどを使って作成した、VR操作が可能なプロジェクトを対象にしています。
PCを接続したまま使うVRプレビューではなく、Quest本体だけで動くアプリを作成します。

※PCのVRプレビューで動く作品でも、そのままQuest本体で同じように動くとは限りません。
使用している描画機能や作品の重さによっては、Quest向けの調整が必要です。最初は、小さなテスト用の作品で起動を確認しましょう。

全体の流れ

第1章 Meta側の準備

- Step 1 Meta開発者アカウントを準備
- Step 2 開発者チームを作成
- Step 3 アカウント認証・二段階認証
- Step 4 アプリを作成〈任意〉

第2章 QuestとPCを開発環境にする

- Step 5 Questを開発者モードにする
- Step 6 Android開発環境を準備する

第3章 UE5をQuest向けに設定する

- Step 7 OpenXR Pluginを確認
- Step 8 Android向け設定

第4章 Questで実際に動かす

- Step 9 パッケージ化
- Step 10 APKをインストール
- Step 11 Questで起動

第1章　Meta側の準備

Step 1　Meta開発者アカウントを準備

まず、普段Questで使っているMetaアカウントを、開発用として利用できる状態にします。
Meta Horizon Developer Dashboardへアクセスして、Metaアカウントでログインします。
開発者登録がまだの場合は、画面の案内に沿って登録を進めます。

ここで確認すること

☐ Metaアカウントでログインできる
☐ Developer Dashboardを開ける

POINT
Questで使用しているMetaアカウントと、開発で使用するアカウントを揃えておくと管理しやすくなります。

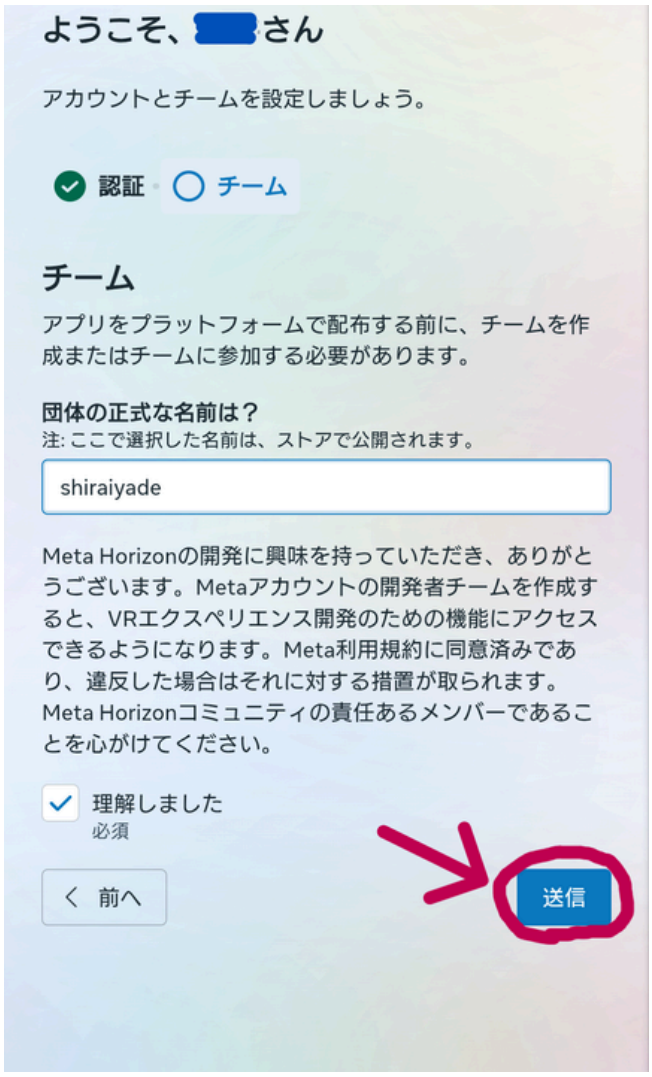
Step２ 開発者チームを作成

次に、Developer Dashboardから開発者チームを作成します。

ここでいう「チーム」は、Meta上でアプリや開発メンバーを管理するための単位です。自分1人だけのチームでも大丈夫です。

ここで確認すること

☐ Developer Teamを作成した
☐ 自分のMetaアカウントがチームに参加している



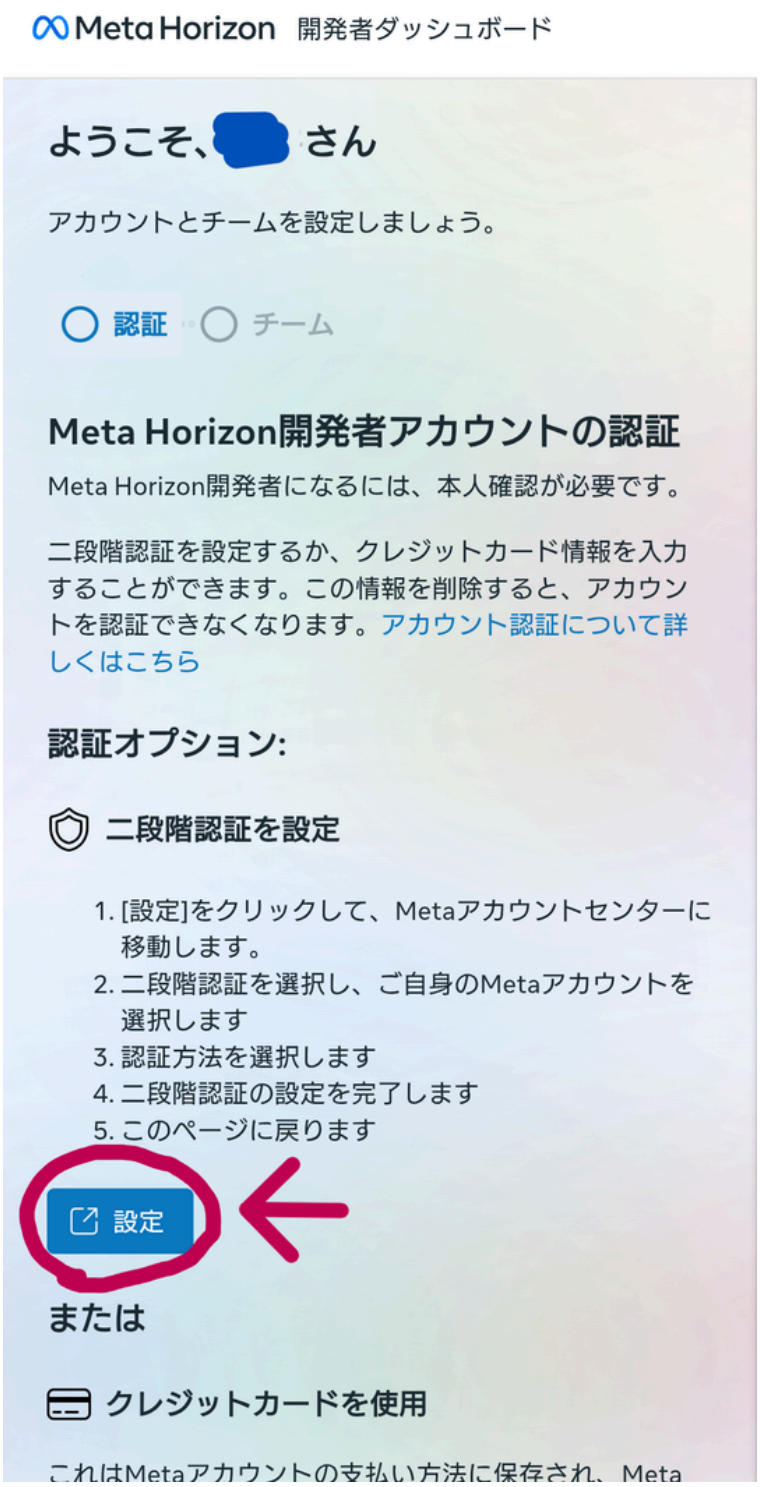


Step 3 アカウント認証・二段階認証

開発者モードを利用するために、Metaの開発者アカウントを認証します。Developer Dashboardの認証画面を開き、表示される案内に沿って進めてください。
電話番号によるSMS認証など、画面に表示される方法で手続きを完了します。

ここで確認すること

☐ Developer Accountの認証が完了している
☐ 必要に応じて二段階認証を設定した



Step 4 アプリを作成〈任意〉

このStepは、今後の配布やMetaのプラットフォーム機能を利用するための準備です。自分のQuestへアプリを直接インストールして確認する今回の手順では、App IDの入力は扱いません。

Developer Dashboardから、新しいアプリを作成します。
「Create a New App」からアプリ名を入力し、Quest向けであれば対象プラットフォームとしてMeta Horizon Storeを選択します。
作成すると、そのアプリ専用の
App ID
が発行されます。
App IDは、将来Metaのプラットフォーム機能を使用するときなどに利用します。





POINT

自分のQuestへAPKをインストールしてテストするだけなら、App IDを使わないケースもあります。
ただし、今後の配布やMeta機能の利用を考えて、この段階でアプリを作成しておくと分かりやすいです。

第2章 QuestとPCを開発環境にする

Step 5　Questを開発者モードにする

次に、Meta Questを
Developer Mode（開発者モード）
にします。
スマートフォンのMeta Horizonアプリを開き、使用するQuestを選択します。
そこからDeveloper ModeをONにします。
開発者モードをONにすることで、Meta Horizon Store以外から自分で作成したアプリをQuestへインストールできるようになります。

PCとQuestをUSB接続

開発者モードをONにしたら、QuestとPCをUSB-Cケーブルで接続します。
このとき、**充電専用ではなくデータ通信に対応したUSBケーブル**を使用してください。
Quest側に
USBデバッグを許可しますか？
という確認が表示されたら許可します。
自分のPCであれば、
このコンピューターから常に許可
を選択しておくと便利です。

ここで確認すること

□ Developer ModeがON
□ QuestとPCをUSB接続した
□ USB Debuggingを許可した

Step 6 Android開発環境を準備する

Quest本体で動くアプリを作るために、**PCにAndroid用の開発ツールを準備します**。
UE5.7には、必要なツールのインストールを手伝ってくれる機能があります。次の順番で進めましょう。

① UE5からインストールを開始する

UE5でプロジェクトを開き、上部のメニューから、
プラットフォーム（Platforms）→ SDK Management → Android → Install SDK
を選択します。
画面の案内に沿って、Android Studioのインストールを進めます。

② Android Studioをインストールする

インストールする項目と保存先は、基本的に**最初から選ばれている設定のまま**進めます。
特に、保存先は変更しないようにしてください。UE5が必要なファイルを見つけられなくなる場合があります。
インストールが終わったら、Android Studioを起動します。初回設定が表示された場合は、案内に沿って進めてください。

③ 必要な追加ツールを入れる

Android Studioの開始画面で、
More Actions → SDK Manager
を開きます。
次に、**SDK Tools**タブを選択します。
一覧の中にある、
Android SDK Command-Line Tools (latest)
にチェックを入れ、**Apply**を押します。
確認画面や利用規約が表示されたら、内容を確認して進めます。
※このツールは、UE5が必要なSDK・NDKなどをダウンロードするために使います。入れ忘れないようにしましょう。

④ Android Studioを閉じる

追加ツールのインストールが終わったら、OKを押して設定画面を閉じ、Android Studioも終了します。
UE5のセットアップ画面に戻り、残りのツールのインストールが完了するまで待ちます。
完了時に、成功したことを示すメッセージが表示されているか確認してください。

⑤ PCを再起動する

セットアップが完了したらUE5を閉じ、PCを再起動します。

⑥ UE5で確認する

再起動後、もう一度UE5を開き、
プラットフォーム（Platforms）→ SDK Management → Android
を確認します。
必要なバージョンとインストール済みのバージョンが一致し、インストールや修復を求められていなければ、準備完了です。

ここで確認すること

☐ PCにAndroid Studioをインストールした
☐ Android SDK Command-Line Toolsをインストールした
☐ UE5側のセットアップが成功した
☐ PCを再起動した
☐ UE5でAndroid SDKのインストールや修復を求められていない

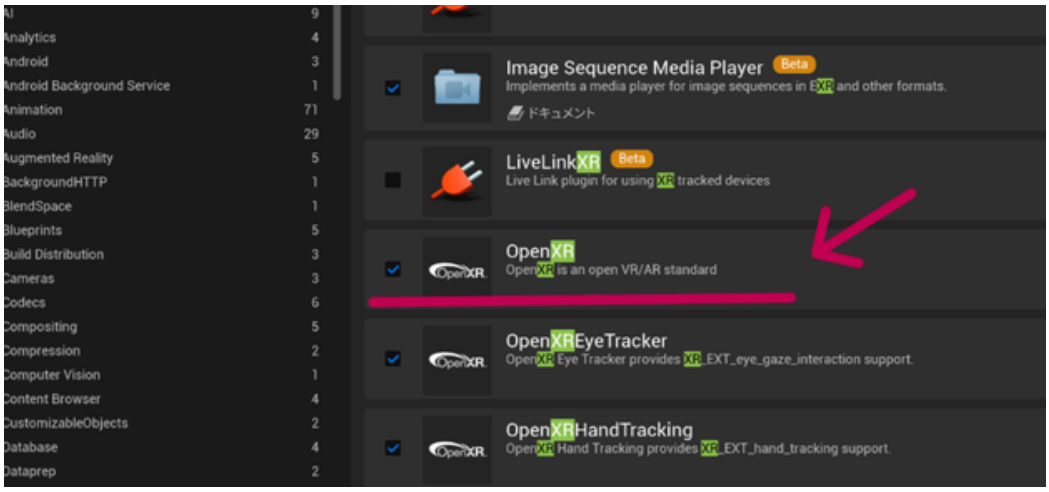
第3章 UE5をQuest向けに設定する

Step 7 OpenXR Pluginを確認

UE5のプロジェクトを開きます。
上部メニューから、
編集（Edit）→ プラグイン（Plugins）を開き、
検索欄に、**OpenXR**と入力します。
一覧に表示される 「OpenXR」 の左側にチェックが入っていることを確認します。

チェックが入っていない場合は、チェックを入れて有効にします。再起動を求められた場合は、UE5を再起動してください。

※本資料では、UE5標準のOpenXRを使って進めます。Meta XR Pluginの追加インストールは、この手順では必要ありません。



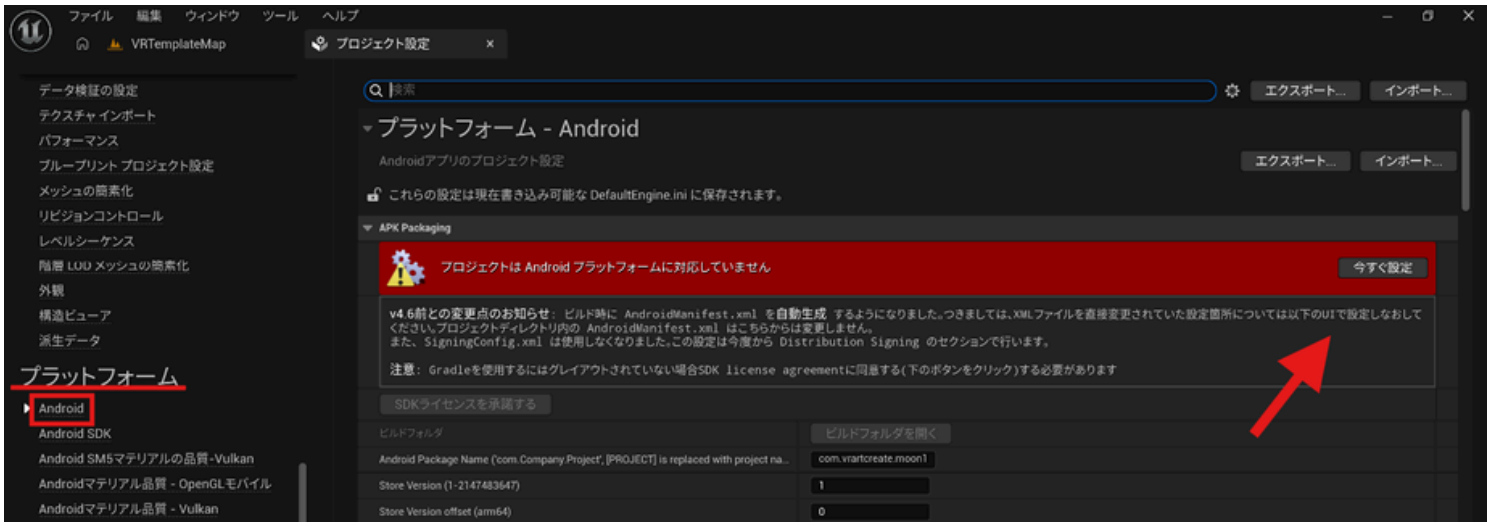
ここで確認すること

□ 「OpenXR」が表示される
□ 「OpenXR」にチェックが入っている
□ 再起動を求められた場合は、再起動が完了している

Step 8 Android向け設定

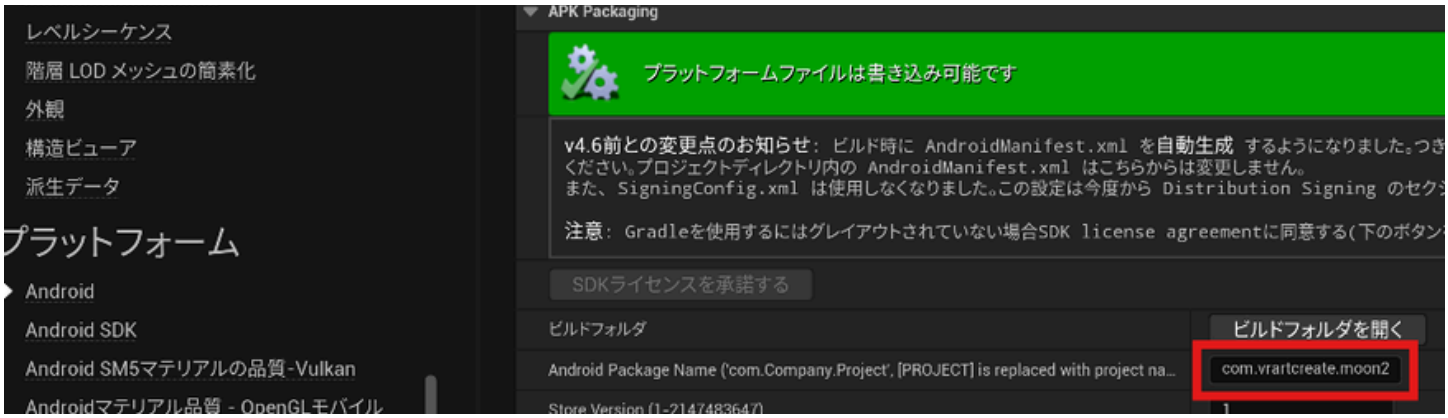
① Android向けの初期設定をする

「編集 → プロジェクト設定 → プラットフォーム → Android」を開きます。
上部に「プロジェクトはAndroidプラットフォームに対応していません」と赤く表示されている場合は、右側の「今すぐ設定」をクリックします。
これは、プロジェクトにAndroid向けの設定ファイルを準備する操作です。クリック後、赤い案内が消えることを確認してください。
すでに設定済みで、この案内が表示されていない場合は、次へ進みます。



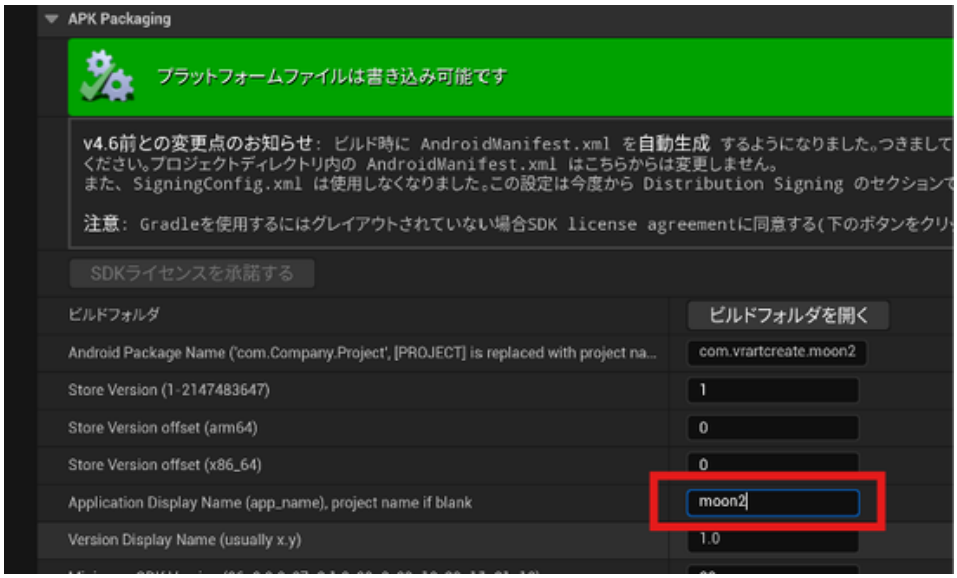
② アプリを識別する名前を設定する

「Android Package Name」に、アプリを識別するための名前を入力します。
入力例：com.vrartcreate.myvrproject
日本語やスペースを使わず、半角英字の小文字を中心に、画像のようにピリオドで区切って入力しましょう。ほかの作品を作る場合は、最後の部分を別の名前にします。
この名前は、Questのアプリ一覧に表示されるタイトルとは別のものです。



③ Questに表示するアプリ名を設定する

「Application Display Name」に、Quest上で表示したいアプリ名を入力します。
空欄の場合は、プロジェクト名が使われます。



④ Quest向けのビルド設定を確認する

同じAndroid設定画面を下へスクロールし、「ビルド」にある次の項目を確認します。

- **Support arm64：ON**
  QuestのCPUに対応したアプリを作成するための設定です。
- **Support Vulkan：ON**
  今回の手順では、Vulkanという描画方式を使用します。
- **Support Vulkan Desktop［Experimental］：OFF**
  今回は使用しません。

すでに画像と同じ状態であれば、そのまま進めます。

⑤ Meta Quest向けのパッケージ設定を確認する

「Advanced APK Packaging」にある、
Package for Meta Quest devices
にチェックが入っていることを確認します。

これは、Meta Quest向けのアプリとしてパッケージ化するための設定です。

また、「App Bundles」の **Generate bundle（AAB） はOFF**にします。本資料では、Questへ直接インストールするAPK形式で作成します。

> ※Google Play Servicesの赤い表示は、今回の手順では設定不要です。 Questへの直接インストールではGoogle Playの機能を使わないため、先ほどのAndroid初期設定の赤い表示とは区別してください。



⑥ Questで最初に表示するマップを選ぶ

「編集 → プロジェクト設定 → マップ＆モード」を開きます。
「ゲームのデフォルトマップ」で、**Questで起動したい作品のマップ**を選びます。

画像では「VRTemplateMap」を選択しています。自分の作品を別の名前のマップに保存している場合は、そのマップを選んでください。
※上段の「エディタのスタートアップマップ」と間違えないようにしましょう。設定するのは、下段の「ゲームのデフォルトマップ」です。



第4章 Questで実際に動かす

Step9 パッケージ化

いよいよUE5のVR作品をQuest用アプリにします。

①パッケージ化の前に、UE5のプロジェクトが保存されている場所を確認します。

日本語を含む場所やOneDrive内では、環境によってパッケージ化に失敗する場合があります。たとえば、次のようなOneDriveの外にある、半角英数字の場所を使用します。

保存場所の例：C:\VRApp\UEprojects\moon2

保存場所を変更する場合は、次の手順で進めてください。

1. UE5で作業を保存し、終了します。
2. プロジェクトのフォルダーを丸ごと、上記のような場所へコピーします。
3. コピー先のフォルダー内にある .uproject ファイルをダブルクリックして開きます。

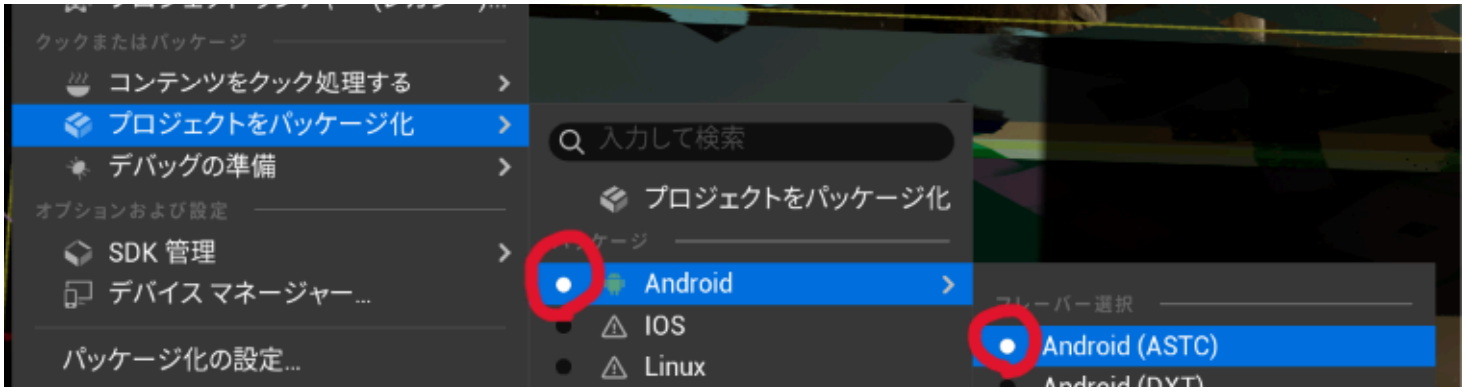
※ .uproject ファイルだけでなく、フォルダー全体をコピーしてください。元のフォルダーは、正常に動作することを確認するまで残しておきましょう。

すでに適切な場所に保存している場合は、変更不要です。

②QuestはAndroid端末なので、Android向けにパッケージ化します。

UE5のPlatforms → Androidからパッケージ化を実行します。
パッケージ化では、作品を動かす機器に合わせて選択します。

Windows PCで処理し、Questに映像を送って動かす場合：「Windows」
Quest本体で処理し、PCにつながずに動かす場合：「Android（ASTC）」
本資料では、Quest本体だけで作品を動かすため、「Android（ASTC）」を選択します。



保存先フォルダを指定すると、UE5が必要なファイルを作成します。
処理が正常に終了すると、APKなどのファイルが生成されます。

初回のパッケージ化には時間がかかります。
私の場合、40分程かかりました。コーヒーでも飲みながら気長に待ちましょう（＾＾）

ここで確認すること

☐ Android向けPackageが完了した
☐ APKが生成された
☐ Packaging Errorが出ていない

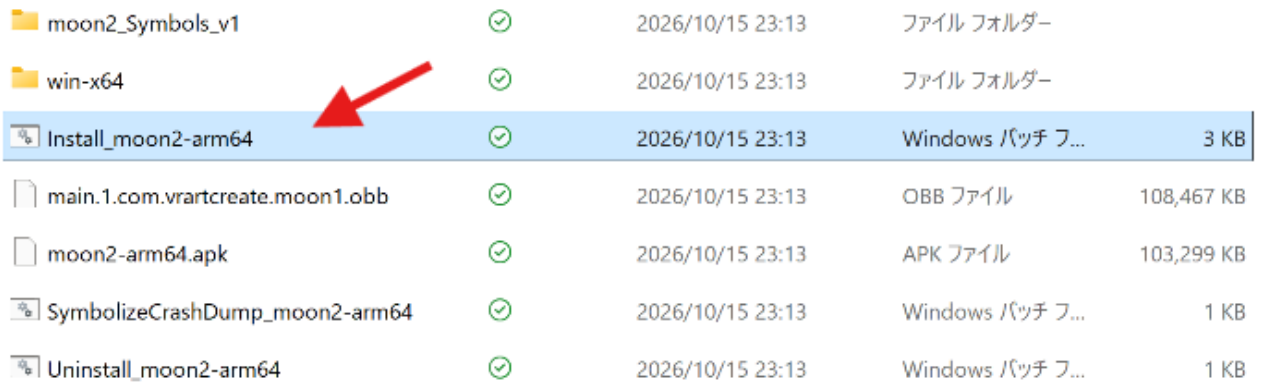
Step 10　APKをQuestへインストール

作成したAPKをQuestへインストールします。
方法はいくつかありますが、初心者の場合は、UE5がパッケージ化時に生成するインストール用ファイルを利用すると分かりやすいです。

パッケージ化で作成された保存先フォルダーを開きます。

QuestをPCへUSB接続し、画像の赤い矢印で示した「Install_」で始まるファイルをダブルクリックしてください。画像では「Install_moon2-arm64」です。

ファイル名はプロジェクトによって異なります。また、PCの設定によっては、末尾の「.bat」が表示されない場合があります。



インストール処理が成功すれば、APKがQuestへ転送されます。
[（Meta Quest Developer Hubを利用してAPKをインストールする方法もあります。）

このファイルだけを別の場所に移さず、同じフォルダーにあるほかのファイルも、そのまま残してください。

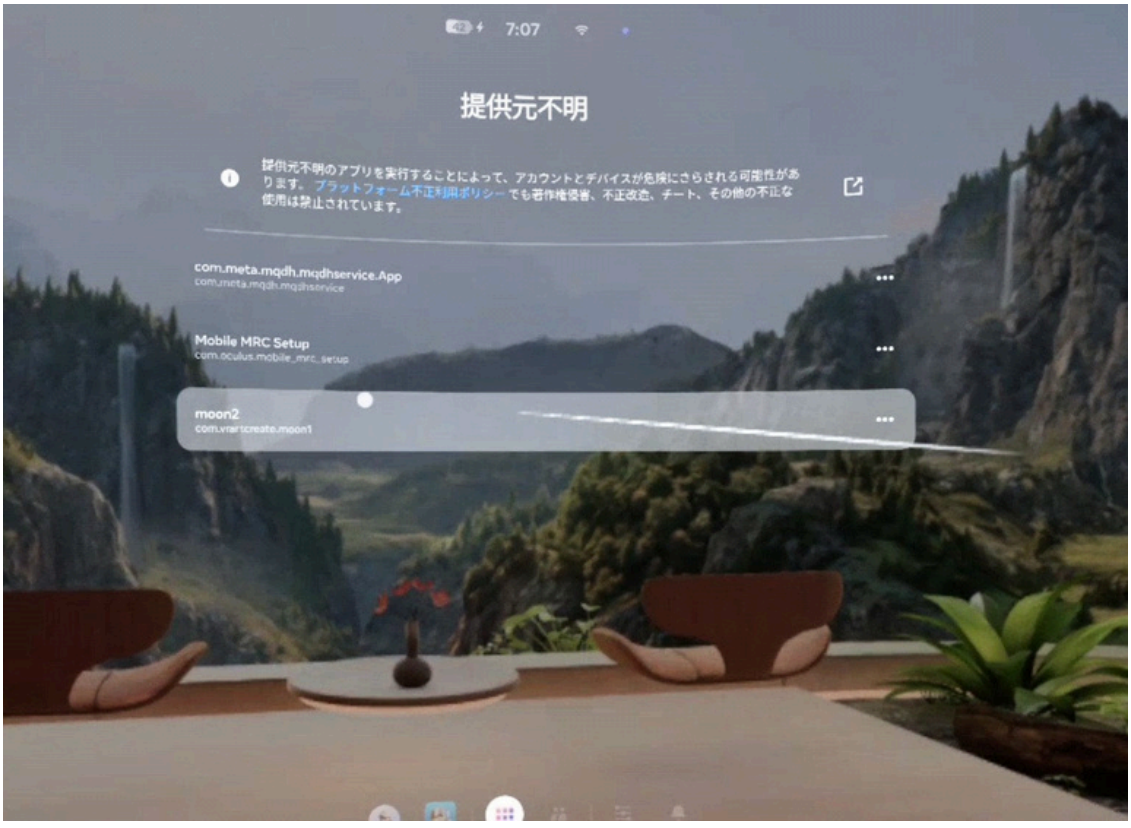
ここで確認すること

☐ QuestをUSB接続している
☐ APKのインストールが完了した
☐ エラーが表示されていない

Step11 Questで起動

最後にQuestを装着します。

インストールしたアプリは、Questのアプリライブラリ内の**「提供元不明（不明なソース）」に表示されます**。そこからアプリを選択して起動してください。表示名はQuestのシステムバージョンによって異なる場合があります。



UE5で作ったVR空間がQuest上で表示されれば成功です。

🎉 完成！

これで、
UE5 → Android Package → Questへインストール → 実機起動
まで完了です。
最初は設定項目が多く感じますが、一度環境を作ってしまえば、2回目以降はかなり楽になります。

作品を修正したら、

UE5で修正
↓
パッケージ化
↓
Questへインストール
↓
実機確認

という流れを繰り返していきます。

最終チェックリスト

うまく起動できない場合は、まず以下を確認してください。

Meta側

☐ Meta Developer Accountを準備した
☐ アカウント認証を完了した
☐ Developer Teamを作成した

Quest側

☐ Developer ModeがON
☐ USB Debuggingを許可した
☐ データ通信対応USB-Cケーブルを使用している

PC側

☐ Android SDKを準備した
☐ Android NDKを準備した
☐ JDKを準備した
☐ PCからQuestを認識できる

UE5側

☐ OpenXRが有効
☐ Android設定を完了した
☐ Package Nameを設定した
☐ Questをデバイスとして認識できる

実機テスト

☐ Android向けPackageに成功
☐ APKをQuestへインストールした
☐ QuestからVR作品を起動できた