Meta Quest 3: podłączenie i wgranie aplikacji

Standalone VR/MR; konfiguracja przez prowadzącego

Odbiorca: prowadzący lub administrator. Student korzysta z już skonfigurowanego Quest 3. Sprzęt opcjonalny, bez wpływu na ocenę.

Dwie różne ścieżki: Quest standalone uruchamia APK bez komputera; **Meta Horizon Link** uruchamia PC VR na zgodnym Windows i przesyła obraz do gogli. Wyników wydajności obu trybów nie porównuj jak tego samego pomiaru.

1. Przygotuj urządzenie i dostęp deweloperski

1. Skonfiguruj Quest 3 i sparuj je z aplikacją *Meta Horizon* przy użyciu uprawnionego konta Meta prowadzącego lub instytucji. Do włączenia trybu programisty wymagane jest uprawnione konto zespołu deweloperskiego Meta. Nie przekazuj konta studentom.
2. W **Meta Horizon** > **urządzenie** > **Headset Settings** > **Developer Mode** włącz tryb deweloperski. Po aktualizacji Meta nazwy ekranów mogą się zmienić.
3. Podłącz gogle do komputera przez **USB-C z transmisją danych**, załóż gogle i potwierdź komunikat **Allow USB debugging** na urządzeniu. Na Windows w razie braku sterownika sprawdź oficjalny *Oculus ADB Driver*; na Ubuntu sprawdź uprawnienia udev.
4. Na komputerze z Android SDK uruchom `adb devices -l`. Prawidłowy stan to `device`, nie `unauthorized` ani `no permissions`.

2. Zbuduj Quest APK w Unity

1. W Unity Hub dla zatwierdzonego edytora dodaj **Android Build Support**, **Android SDK/NDK Tools** i **OpenJDK**.
2. Otwórz projekt VR ze sceną obsługującą XRI. W **File** > **Build Profiles** wybierz dostępny profil **Meta Quest**, a jeśli nie występuje w tej wersji, cel **Android**; wykonaj **Switch Platform**.
3. W **Project Settings** > **XR Plug-in Management** włącz **OpenXR** dla tego celu i funkcje **Meta Quest Support**. Wybierz właściwe profile kontrolerów. Nie mieszaj w tym samym profilu Quest z providerem ARCore do telefonu.
4. W **Player Settings** zweryfikuj **IL2CPP**, **ARM64**, nazwę pakietu, obsługiwany poziom Android API i komunikaty XR validation. Dodaj właściwą scenę do **Scene List**.
5. W **Run Device** wybierz wykryty Quest 3 i użyj **Build And Run**. Zapisz wynikowy APK, identyfikator wersji i sumę SHA-256. Testy wydajności wykonuj na odpowiednim buildzie, nie na wersji Debug/Development.

3. Zainstaluj gotowy APK bez Unity

Gdy plik APK został już zbudowany i podpisany, możesz wykonać:

```
adb devices -l
adb install -r WiRR-Quest.apk
```

Nazwa pliku jest przykładowa. Flaga `-r` aktualizuje aplikację przy zgodnym podpisie i identyfikatorze pakietu. Przed reinstalacją wymagającą usunięcia aplikacji zabezpiecz lokalne wyniki. W goglach uruchom aplikację z **Library** > **Unknown Sources** (nazwa menu zależy od wersji systemu).

4. Odbiór demonstracji przed zajęciami

- ☐ `adb devices -l` pokazuje urządzenie jako `device`.
- ☐ APK uruchamia się **na Quest bez Meta Horizon Link**; działa obraz stereo, skala, kontrolery, reset i STOP.
- ☐ Przygotowano wyraźną granicę przestrzeni, asekurację, higienę i możliwość przerwania ekspozycji.
- ☐ Student bez gogli może wykonać wszystkie obowiązkowe zadania na PC Ubuntu/Windows.

Jeśli coś nie działa: najpierw USB i Developer Mode, potem autoryzacja ADB, następnie profil Android/OpenXR i sceny, na końcu `adb logcat`. Brak obrazu na ekranie Unity nie jest dowodem, że aplikacja na Quest działa. Nie wykonuj resetu fabrycznego w ramach zajęć.

Oficjalne materiały: https://developers.meta.com/vr/documentation/android-apps/enable-developer-mode/; https://developers.meta.com/vr/documentation/unity/unity-build/.

5. Zapis testu sprzętowego

- Zapisz datę, wersję Unity, wersję systemu Quest, wersję APK, SHA-256 i wynik `adb devices -l`; nie udostępniaj numerów seryjnych ani identyfikatorów kont.
- Oddziel test **standalone APK** od streamingu **PC VR**. Przy błędzie aplikacji zapisz pierwsze istotne komunikaty `adb logcat` bez danych prywatnych.
- Odbiór: uruchomienie bez komputera, prawidłowa interakcja, bezpieczne zakończenie sesji i niezależna ścieżka PC.