Telefon Android: instalacja i uruchomienie AR

AR Foundation, ARCore, opcjonalny Depth API i APK

Dla prowadzącego oraz chętnych użytkowników własnych telefonów. Wymagania Lab 03–04 w pełni wykonasz także na PC.

1. Sprawdź, czy telefon obsługuje wymagane funkcje

1. Zweryfikuj model na oficjalnej liście https://developers.google.com/ar/devices. **Android nie oznacza automatycznie ARCore**. Sprawdź też *Google Play Services for AR* oraz wymagane uprawnienie kamery.
2. **Depth API jest funkcją osobną.** Brak Depth API może uniemożliwiać okluzję opartą o realną głębię, ale nie musi blokować zwykłego rozmieszczania modeli AR. Aplikacja ma rozpoznawać brak funkcji i pokazać stosowny komunikat.

2. Przygotuj odrębny APK w Unity (prowadzący)

1. W Unity Hub doinstaluj **Android Build Support, SDK/NDK, OpenJDK**. Otwórz projekt z przygotowaną sceną AR, a w **File > Build Profiles > Android** wykonaj **Switch Platform**.
2. W **XR Plug-in Management** aktywuj dla Androida **Google ARCore XR Plug-in**, ze zgodną wersją **AR Foundation**. Skonfiguruj *AR Session*, *XR Origin*, kamerę AR, detekcję płaszczyzn i raycast. Okluzję włącz tylko w obsługiwanym scenariuszu.
3. Zweryfikuj *Minimum/Target API Level*, architekturę **ARM64**, uprawnienie kamery, nazwę pakietu, sceny w **Scene List** oraz kontrolę **Project Validation** odpowiadającą zatwierdzonym wersjom pakietów. Nie kopiuj ustawień profilu Meta Quest do APK telefonu.
4. Zbuduj i podpisz **osobny** APK Android AR. Wydaj wersję i SHA-256 z zaufanego kanału dystrybucji, bez zapisywania kluczy podpisujących w repozytorium.

3. Instalacja na telefonie

Wariant zwykły, dobrowolny (student): pobierz zatwierdzony, podpisany APK z oficjalnego źródła kursu, zapoznaj się z komunikatem Androida dotyczącym instalacji z tego źródła, zaakceptuj ją wyłącznie świadomie, uruchom aplikację i przyznaj dostęp do kamery tylko dla jej funkcji AR. **Nie wyłączaj Play Protect i nie obchodź zabezpieczeń urządzenia.** Jeśli instalacja jest niedozwolona lub niechciana, korzystaj z gotowego laboratorium desktop.

Wariant testowy (telefon prowadzącego): włącz opcje programistyczne i debugowanie USB, podłącz przewód DATA, zatwierdź klucz RSA i wykonaj:

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

Możesz też wybrać telefon w **Run Device** i użyć **Build And Run**. **Student nie musi** instalować Android Studio ani włączać debugowania USB na swoim telefonie.

4. Test uruchomienia i pomiaru

- ☐ Aplikacja prawidłowo oznacza `REAL_AR` tylko przy działającej rzeczywistej sesji ARCore.
- ☐ Model o znanej skali można postawić na wykrytej powierzchni i obejrzeć z innego punktu.
- ☐ Okluzja dostępna tylko po pozytywnej kontroli Depth API; przy braku wsparcia stan `DEPTH_UNSUPPORTED`, bez fikcyjnych pomiarów.
- ☐ Gdy brak ARCore/kamery/sieci, student kończy Lab 03–04 w oznaczonym `SIM_AR` na PC, bez obniżenia oceny.

Ochrona danych: aplikacja nie powinna przesyłać nagrań kamery, geolokalizacji ani identyfikatora telefonu. Wyniki pomiarów z prawdziwego AR i symulacji przechowuj jako dwa różne źródła danych. Sam obraz kamery bez wzorca pomiarowego nie stanowi pomiaru bezwzględnej dokładności kotwicy.

Źródła: https://developers.google.com/ar/devices; https://developers.google.com/ar/develop/unity-arf/enable-arcore; https://developers.google.com/ar/develop/depth.

5. Rejestrowanie zgodności urządzenia

- Zapisz model urządzenia, wersję Androida, ARCore i APK oraz wynik kontroli Depth API. Nie publikuj numeru seryjnego, identyfikatora reklamowego ani obrazu z kamery.
- Przeprowadź próbę wykrycia płaszczyzny i umieszczenia obiektu. Osobno odnotuj, czy realna głębia jest dostępna, bez ekstrapolowania wyniku symulacji.
- Odbiór: realne AR oznaczone jako `REAL_AR`, fallback PC jako `SIM_AR`; brak ARCore/Depth nie pogarsza oceny.