

Politechnika Świętokrzyska
w Kielcach
Wydział Elektrotechniki, Automatyki i Informatyki

Rzeczywistość Wirtualna i Rozszerzona

Instrukcja laboratoryjna 1

Ustawienie sceny w silniku Unity

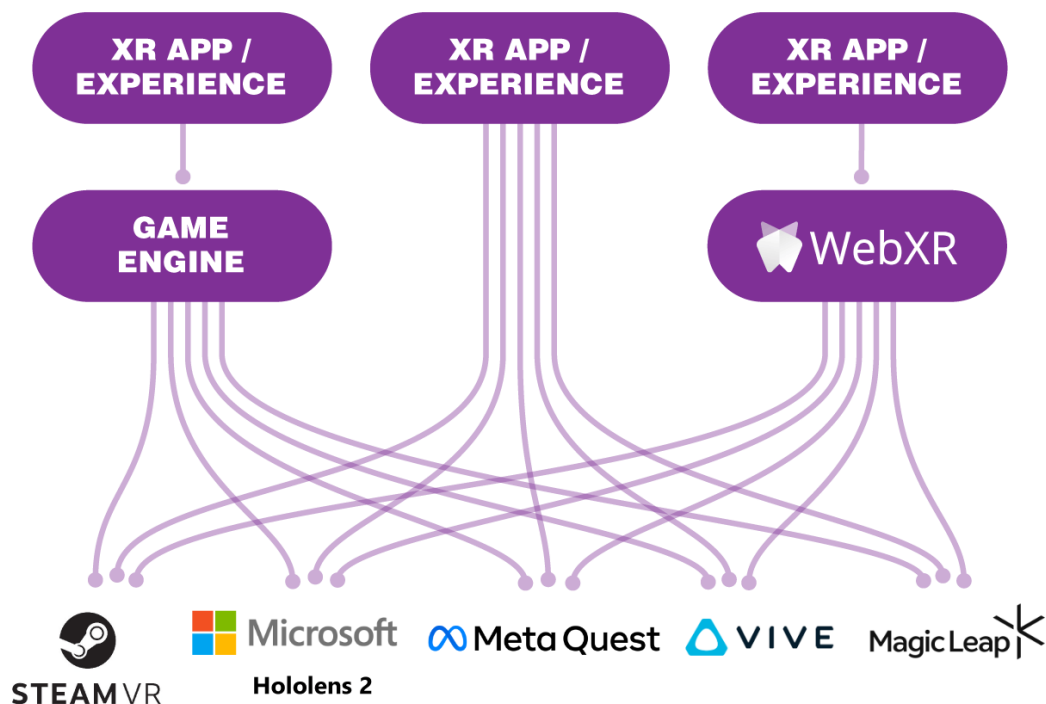
Przygotował:
mgr inż. Michał Wójcik

Kielce, 2025

1. OpenXR

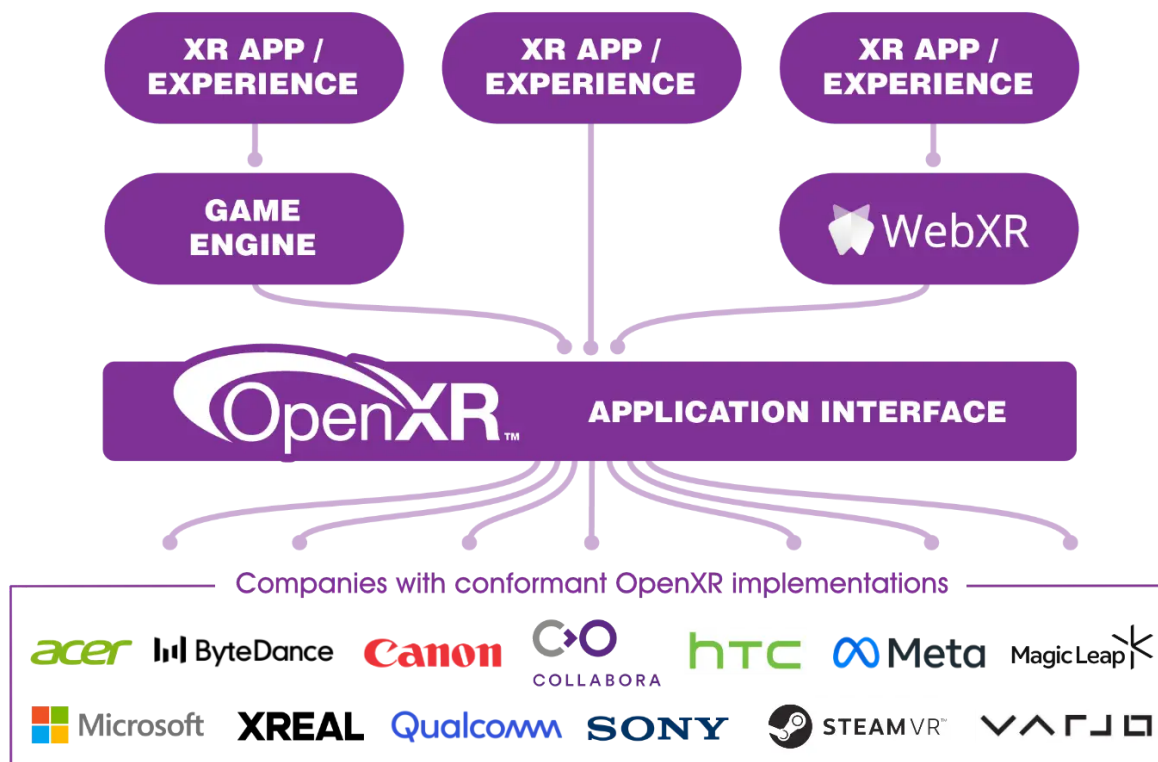
OpenXR to wolny od opłat, otwarty standard, który zapewnia wspólny zestaw interfejsów API do tworzenia aplikacji XR działających na szerokiej gamie urządzeń AR i VR. Dzięki temu skraca się czas i koszty potrzebne deweloperom na dostosowanie rozwiązań do poszczególnych platform XR, a jednocześnie tworzy się większy rynek łatwo wspieranych aplikacji dla producentów urządzeń wykorzystujących standardy OpenXR.

Przed opracowaniem OpenXR, deweloperzy musieli pisać kod oddzielnie dla każdego obsługiwanego urządzenia na rynku. Dzięki OpenXR deweloperzy mają teraz dostęp do jednego, wydajnego, wieloplatformowego interfejsu API, który umożliwia im stworzenie rozwiązania raz, a następnie łatwe przeniesienie i optymalizację go. To pozwala dotrzeć do większej liczby użytkowników, nie rezygnując z innowacyjnych funkcji konkretnych platform poprzez rozszerzenia i warstwy API OpenXR.



Before OpenXR: Applications and engines needed separate proprietary code for each device on the market.

Źródło: https://www.khronos.org/api/index_2017/openxr



OpenXR provides a single cross-platform, high-performance API between applications and all conformant devices.

Źródło: https://www.khronos.org/api/index_2017/openxr

OpenXR standaryzuje używanie wielu różnych urządzeń XR na wielu platformach, w tym:

- Wyświetlacze nagłowne
- Kontrolery
- Stacje bazowe
- Urządzenia śledzące: ciało, dłonie, obiekty, oczy itd.
- Urządzenia haptyczne
- Silniki gier
- Infrastrukturę chmurową/5G
- I wiele innych

2. Unity

Unity to zaawansowany silnik do tworzenia gier i aplikacji interaktywnych, który wspiera zarówno projekty 2D, jak i 3D. Oferuje bogate środowisko do programowania w języku C#, co pozwala na tworzenie skryptów sterujących logiką gry oraz interakcją z obiektami.

Silnik korzysta z komponentowego modelu architektury, co umożliwia elastyczne dodawanie i modyfikowanie zachowań obiektów w scenie. Unity zawiera wbudowany silnik fizyczny, który obsługuje kolizje, symulacje ruchu oraz zachowania ciał stałych, co pozwala na realistyczne odwzorowanie dynamiki obiektów.

Unity jest kompatybilne z wieloma platformami, w tym Windows, macOS, Linux, Android, iOS, a także VR/AR, co ułatwia rozwój aplikacji wieloplatformowych. Dzięki Asset Store deweloperzy mają dostęp do szerokiej gamy zasobów, wtyczek i narzędzi, co przyspiesza proces tworzenia.

W trakcie zajęć laboratoryjnych wykorzystywana będzie wersja **6000.2.6f2** silnika.

2.1. XR Interaction Toolkit

Pakiet XR Interaction Toolkit to zestaw narzędzi oraz system interakcji na wysokim poziomie, oparty na komponentach, przeznaczony do tworzenia doświadczeń VR i AR. Zapewnia solidną podstawę, która umożliwia interakcje 3D i interfejsu użytkownika na podstawie zdarzeń wejściowych silnika. Rdzeniem tego systemu jest zestaw podstawowych komponentów: "Interactor" i "Interactable" oraz Menedżer Interakcji, który łączy te dwa typy komponentów. Pakiet zawiera również komponenty, które można wykorzystać do lokomocji i wizualizacji.

Pakiet zawiera zestaw komponentów, umożliwiających między innymi:

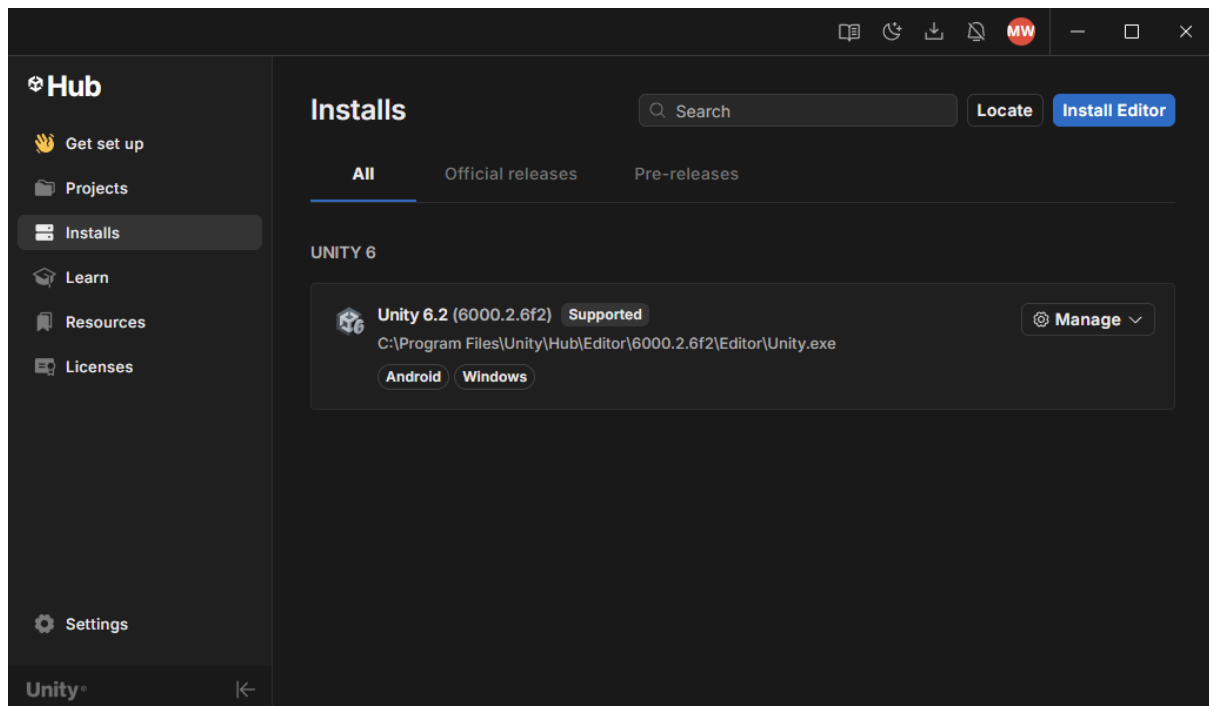
- Obsługę wielu platform: Meta Quest, OpenXR, visionOS itd.
- Podstawowe interakcje z obiektami: unoszenie, wybór i chwytanie
- Haptyczne wibracje dla kontrolerów XR
- Wizualne informacje (podświetlenie, renderowanie linii) wskazujące możliwe i aktywne interakcje
- Podstawowe interakcje z płótnem UI
- Narzędzia do interakcji z XR Origin, szkieletem kamery VR do obsługi doświadczeń stacjonarnych i w skali pomieszczenia

XR Interaction Toolkit udostępnia też symulator urządzenia XR: komponent, który umożliwia pracę nad aplikacją XR bez posiadania docelowego sprzętu, lub po prostu testowanie jej bez potrzeby zakładania urządzenia na głowę. Symulator nie manipuluje bezpośrednio obiektami wchodzącymi w skład szkieletu XR Origin, zamiast tego symuluje urządzenie wejściowe, takie jak headset lub kontrolery.

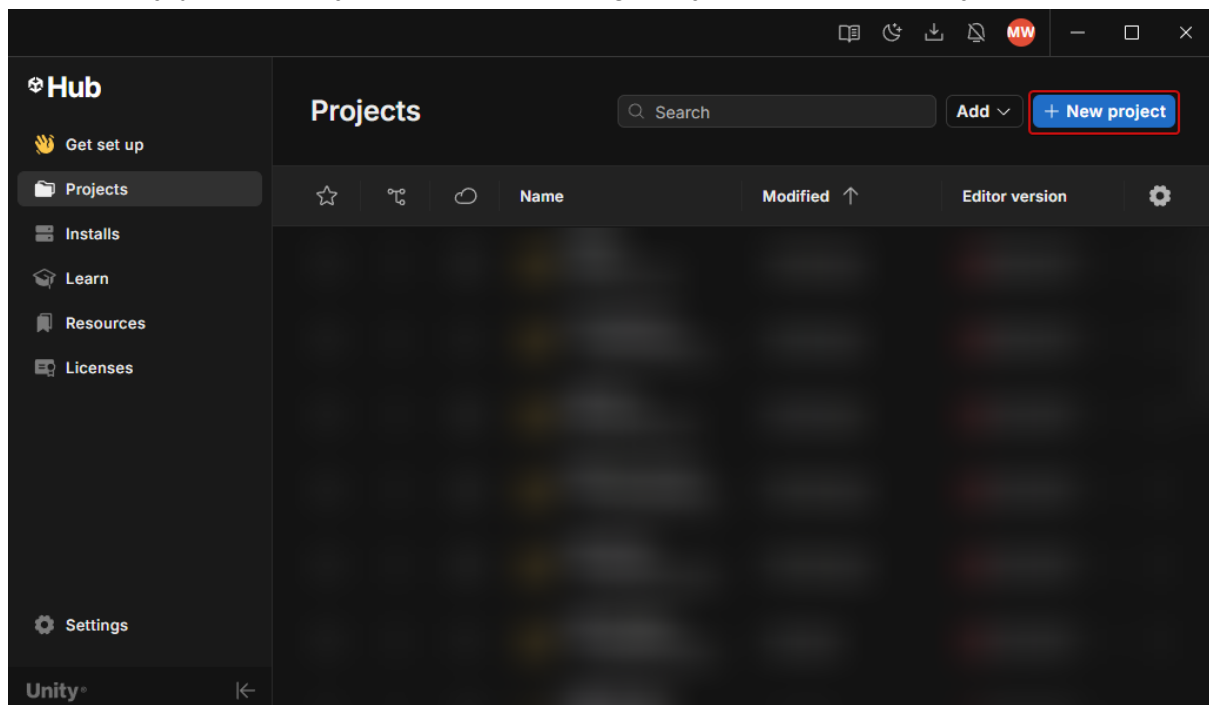
W trakcie zajęć laboratoryjnych pakiet używany będzie w wersji **3.2.1**.

2.2. Utworzenie projektu

Aby utworzyć nowy projekt Unity, należy skorzystać z programu **Unity Hub**. Na początek upewnij się, że zainstalowana jest odpowiednia wersja Unity: **6000.2.6f2**.



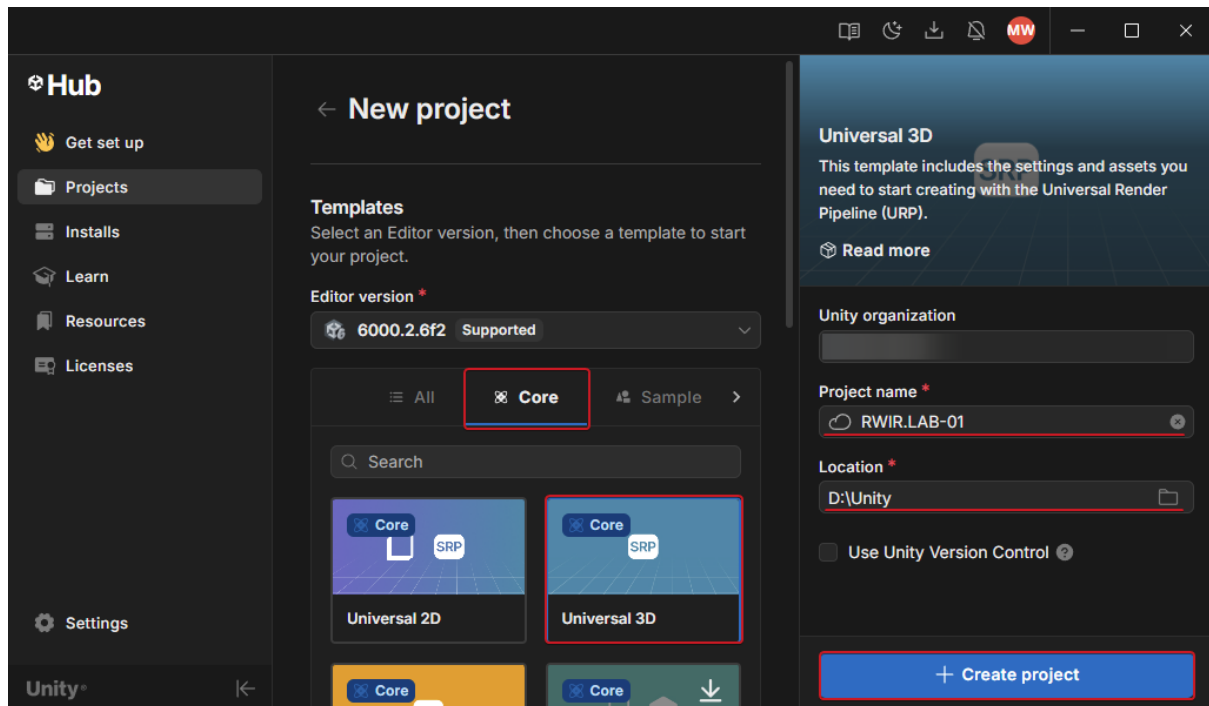
Kolejnym krokiem jest utworzenie nowego projektu w zakładce Projects.



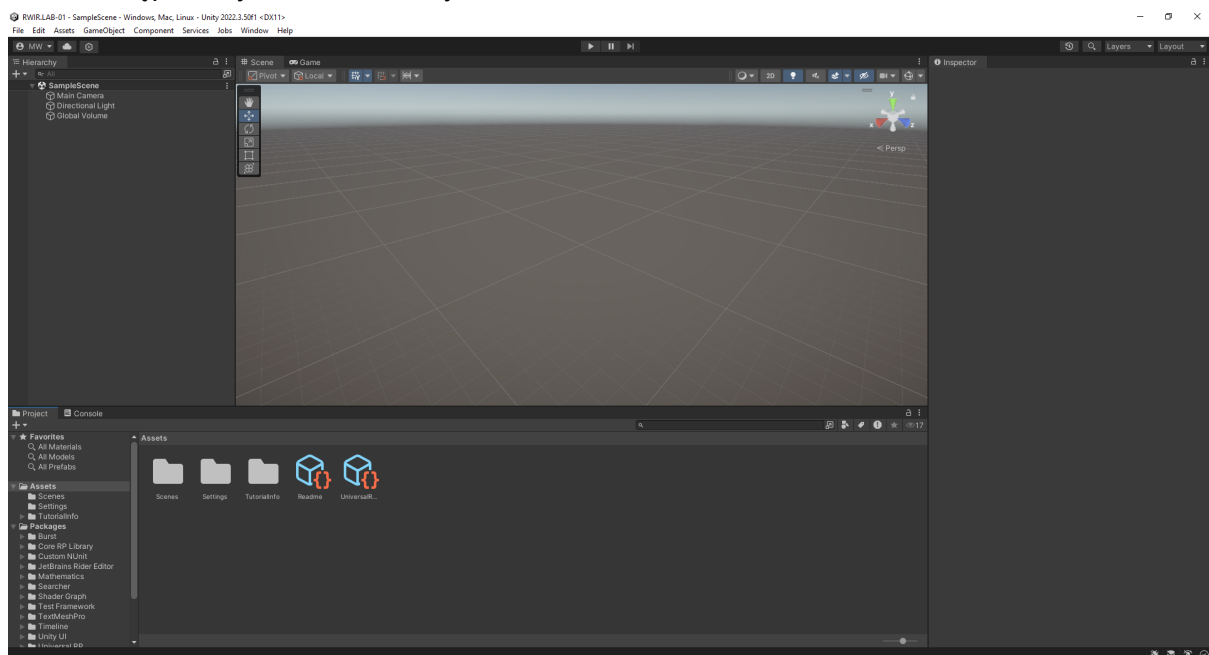
Unity umożliwia przygotowanie projektu VR na dwa sposoby: gotowy szablon projektu dla aplikacji VR, AR oraz MR (Mixed Reality) oraz ręczna instalacja potrzebnych pakietów. Podczas gdy skorzystanie z gotowego szablonu jest szybsze, może okazać się, że pobieramy niepotrzebne zasoby. Przygotowanie projektu od

podstaw pozwala na większą swobodę w doborze pakietów, odrzucenie zbędnych zasobów, a zarazem zmniejszenie skomplikowania projektu na start. Na potrzeby laboratorium oraz w celu zrozumienia zależności pomiędzy danymi komponentami, poniżej zostaje przedstawiona druga opcja.

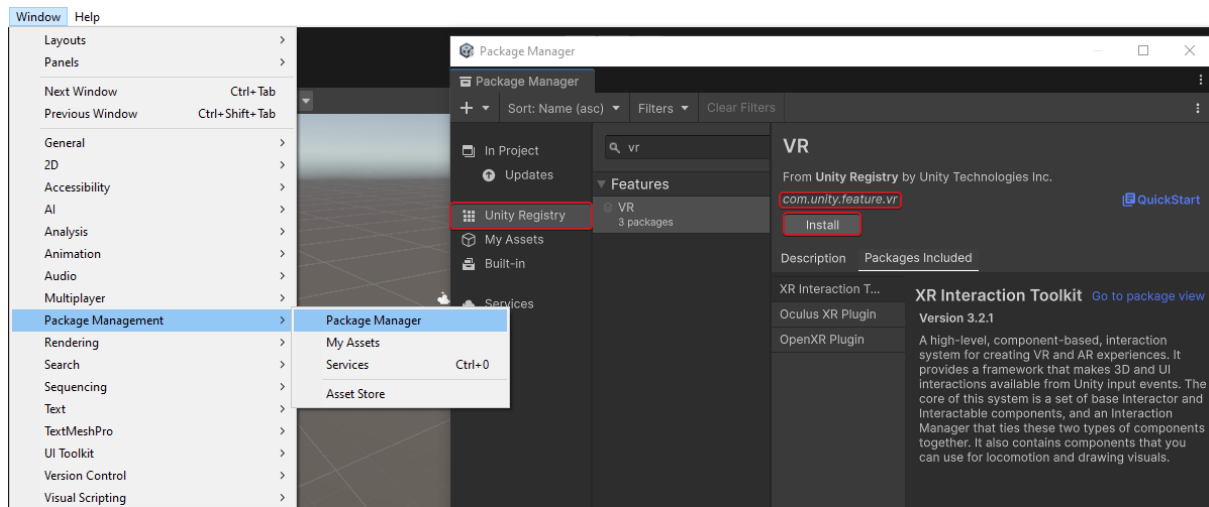
W oknie nowego projektu, wybierz szablon “Universal 3D”, nadaj nazwę projektowi, wybierz folder docelowy i naciśnij “Create project”.



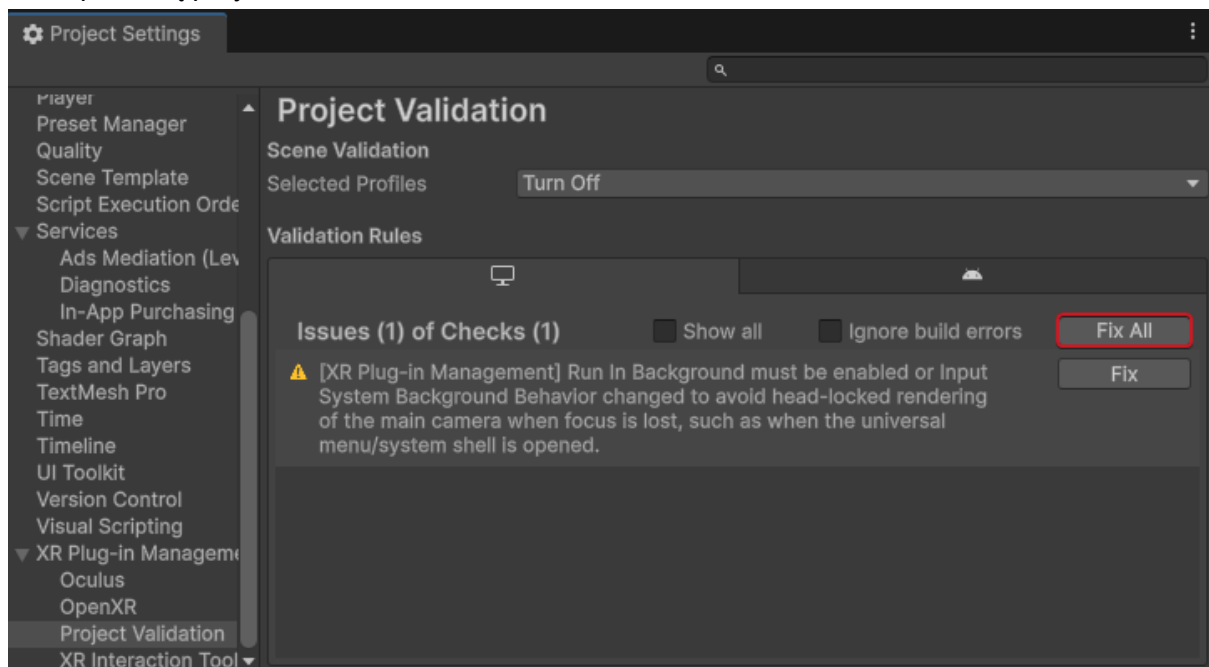
Program Unity Hub utworzy nowy projekt, zainstaluje odpowiednie zależności, a następnie wyświetli okno edytora.



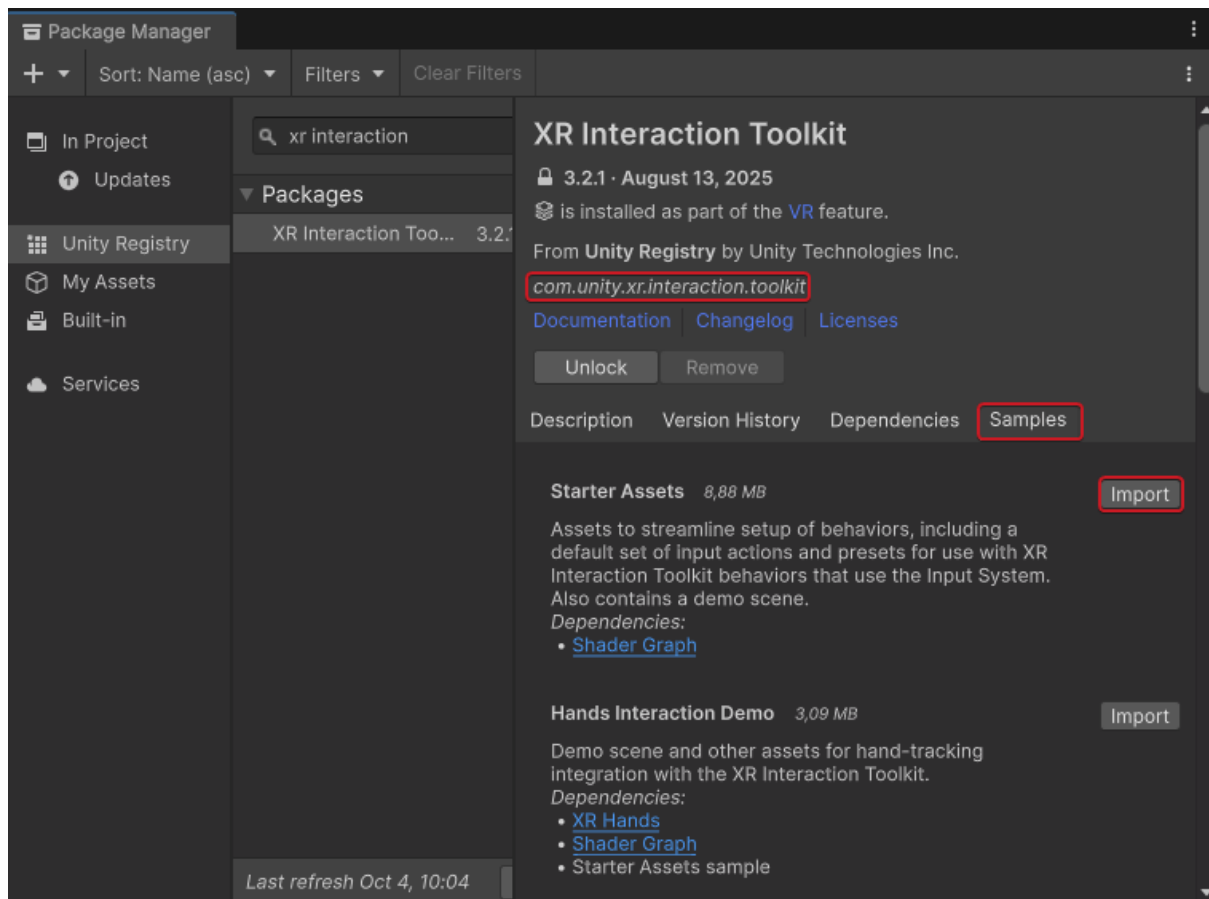
Teraz należy zainstalować XR Interaction Toolkit oraz potrzebne wtyczki. W oknie edytora wybierz **Window > Package Management > Package Manager**, a w wyświetlonym oknie menedżera znajdź pakiet **com.unity.feature.vr** w rejestrze (Unity Registry). Wybierz pakiet z listy i zainstaluj go.



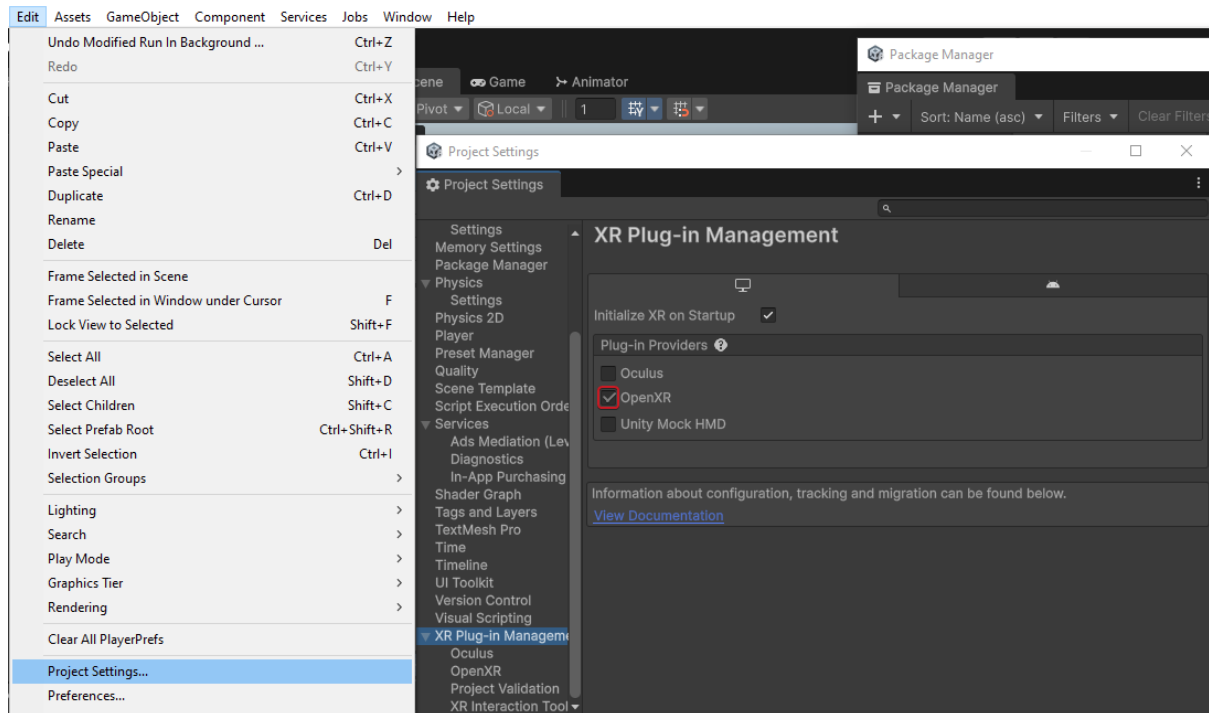
Po instalacji pakietu Unity może zapytać o automatyczną naprawę niektórych ustawień. Można na razie zignorować tę wiadomość lub zgodzić się na tę zmianę za pomocą przycisku "Fix All".



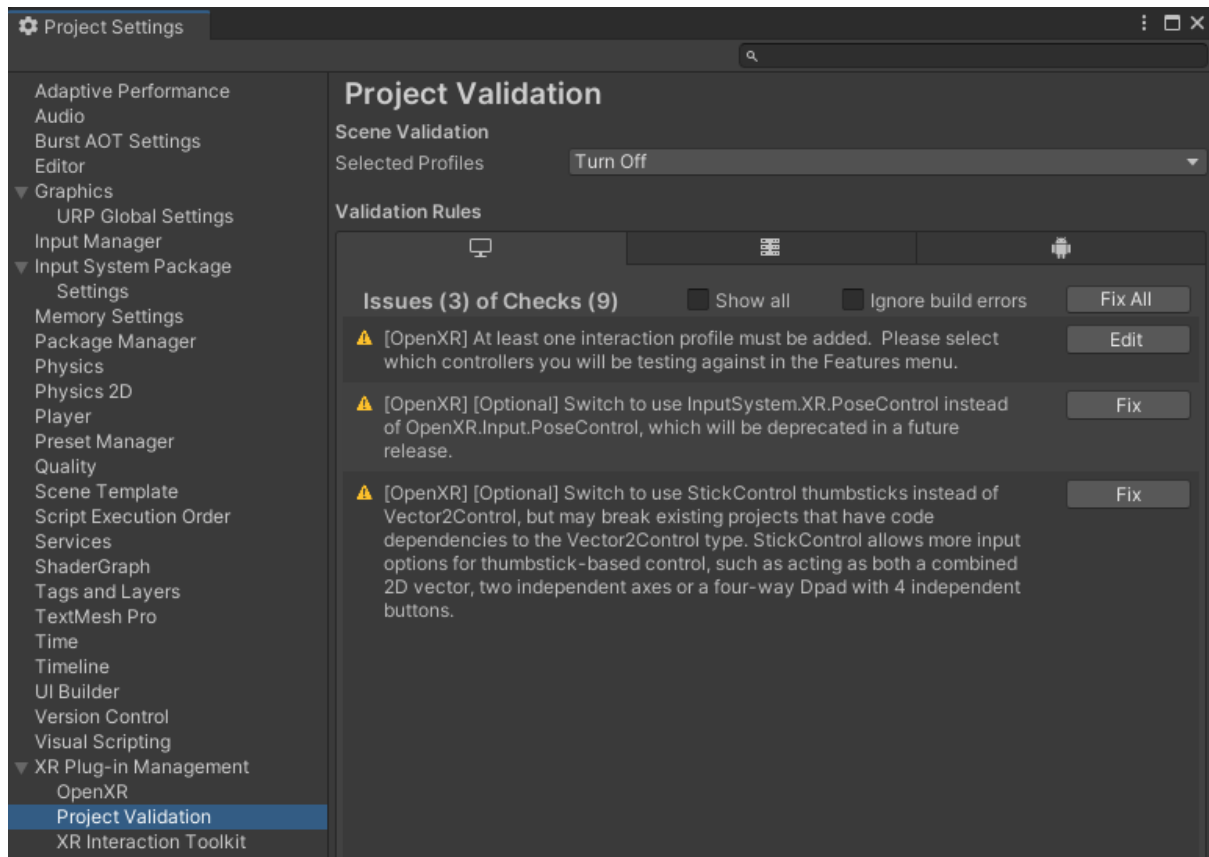
Po instalacji pakietu należy jeszcze raz odnaleźć go w menedżerze pakietów (**com.unity.xr.interaction.toolkit**) i zaimportować **Starter Assets** - jest to zestaw zasobów ułatwiających ustawienie projektu.



Aby móc wykorzystywać urządzenie HTC Vive Pro oraz inne urządzenia wspierające backend OpenXR, przejdź do **Project Settings > XR Plug-in Management** i w sekcji "Plug-in Providers" zaznacz opcję przy **OpenXR**.

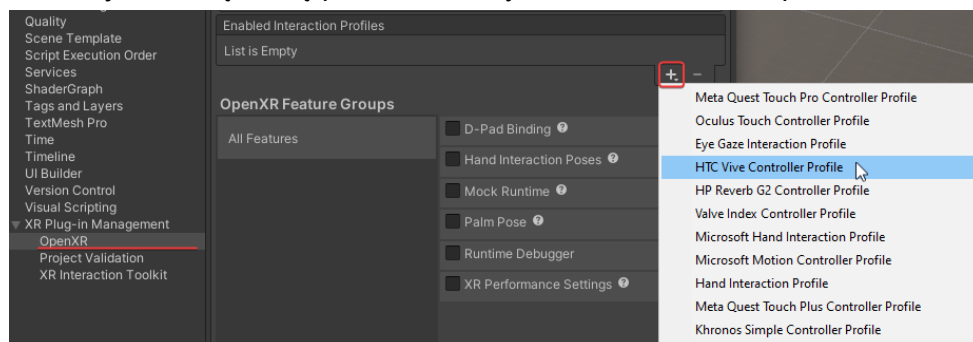


Po zainstalowaniu wtyczki należy ponownie przejść do sekcji rozwiązywania problemów: **Project Settings > XR Plug-in Management > Project Validation**.



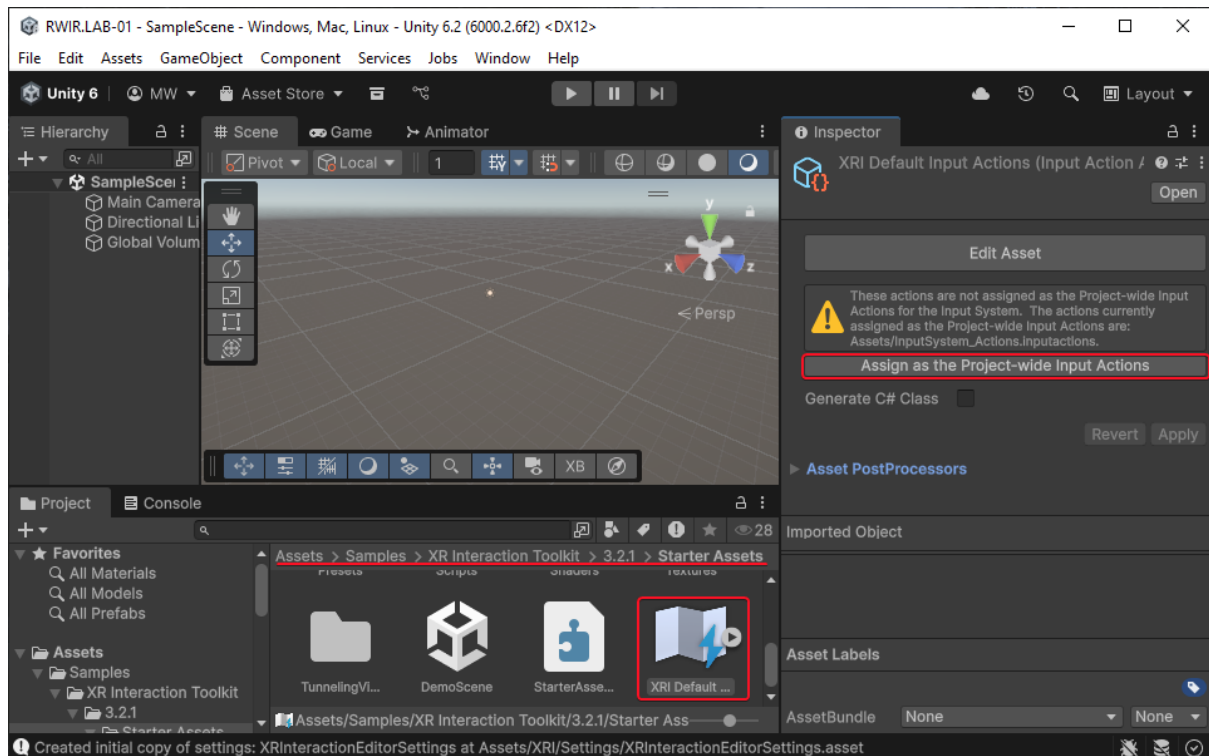
Pierwszy problem związany jest z dodaniem profilu interakcji, co zostanie przedstawione w dalszej części instrukcji. Pozostałe dwa problemy można naprawić automatycznie, klikając przycisk **Fix** przy nich.

Aby dodać nowy profil interakcji dla kontrolerów urządzenia HTC Vive Pro, przejdź do **Project Settings > XR Plug-In Management > OpenXR**, a następnie w sekcji **Enabled Interaction Profiles** dodaj profil **HTC Vive Controller Profile**. Możesz dodać tutaj dowolną liczbę profili, dla których chcesz dodać wsparcie.

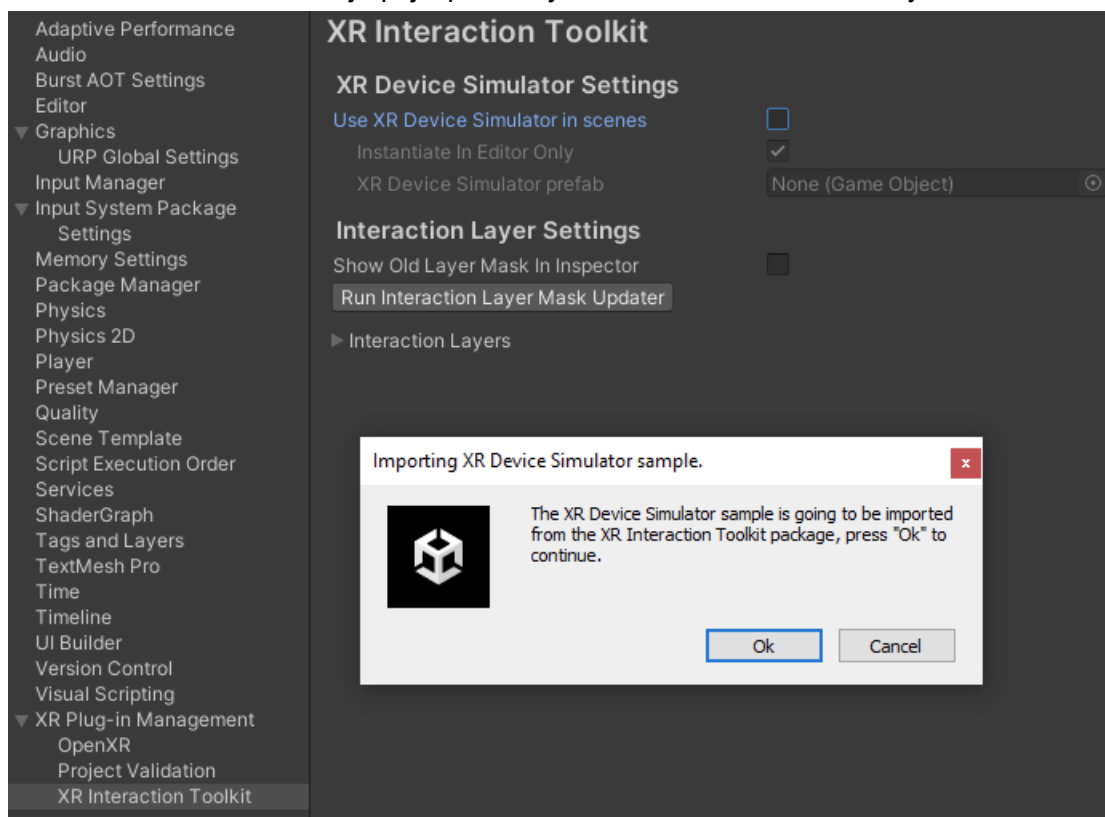


Kolejnym krokiem w ustawieniu kontrolerów będzie przejście do **Assets > Samples > XR Interaction Toolkit > 3.2.1 > Starter Assets**, wybranie pliku konfiguracyjnego **XRI Default Input Actions.inputactions** i ustawienie jako domyślny dla projektu. Plik ten zawiera ustawienia dotyczące **akcji wejścia** dla kontrolerów VR i przypisuje

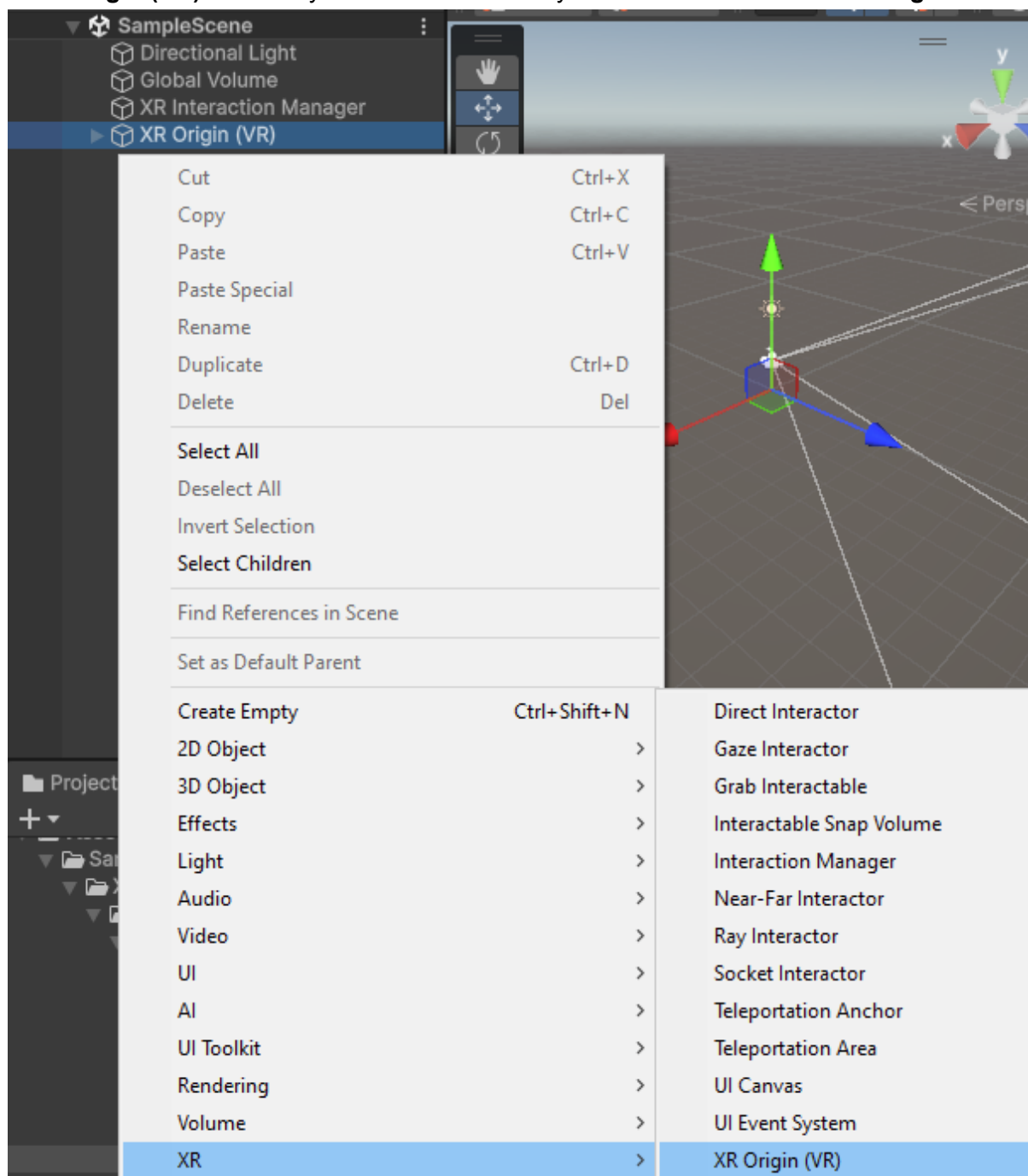
konkretne akcje danym przyciskom i czujnikom urządzenia. Temat systemu wejścia zostanie poruszony na przyszłych laboratoriach.



Jeśli chcesz skorzystać z symulatora, przejdź do **Project Settings > XR Plug-In Management > XR Interaction Toolkit** i zaznacz **Use XR Device Simulator in scenes**. Zaznaczenie tej opcji spowoduje zainstalowanie dodatkowych zasobów.

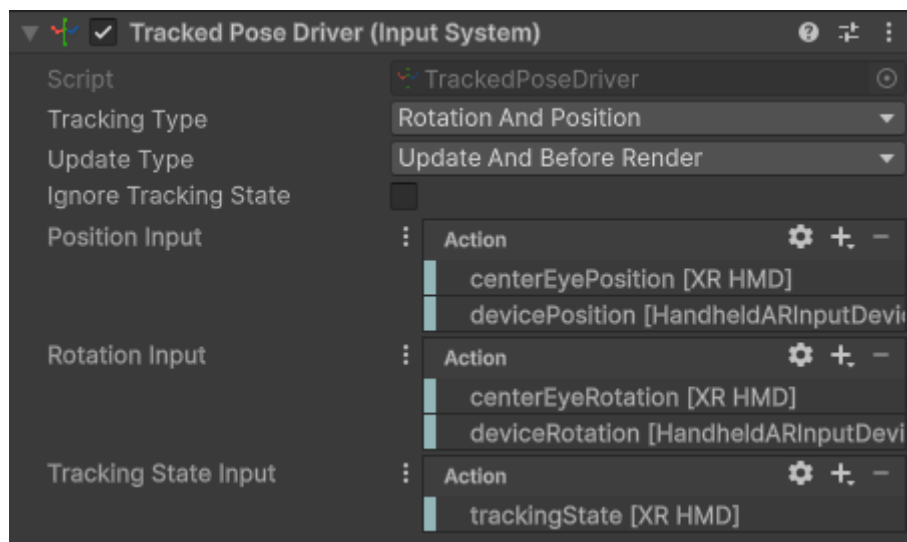


Ustawienia silnika zostały poprawnie zaaplikowane, teraz należy przejść do ustawienia sceny. Można utworzyć nową scenę (File > New Scene lub skrót Ctrl + N) lub skorzystać ze sceny **SampleScene**. Z drzewa obiektów usuń **Main Camera**, a następnie dodaj nowy obiekt klikając prawym przyciskiem myszy na drzewo: **XR > XR Origin (VR)**. Automatycznie zostanie dodany również **XR Interaction Manager**.



XR Origin (XR Rig) to obiekt, który reprezentuje przestrzeń użytkownika. W jego hierarchii znajduje się **Camera Offset**, obiekt służący ustaleniu początkowej odległości śledzonych obiektów od podłogi - dlatego obiekt **Main Camera** reprezentujący głowę użytkownika znajduje się bezpośrednio w hierarchii Offsetu. Wszelkie inne obiekty stanowiące cyfrową reprezentację śledzonych urządzeń, takie jak kontrolery VR, również tutaj zostaną umieszczone.

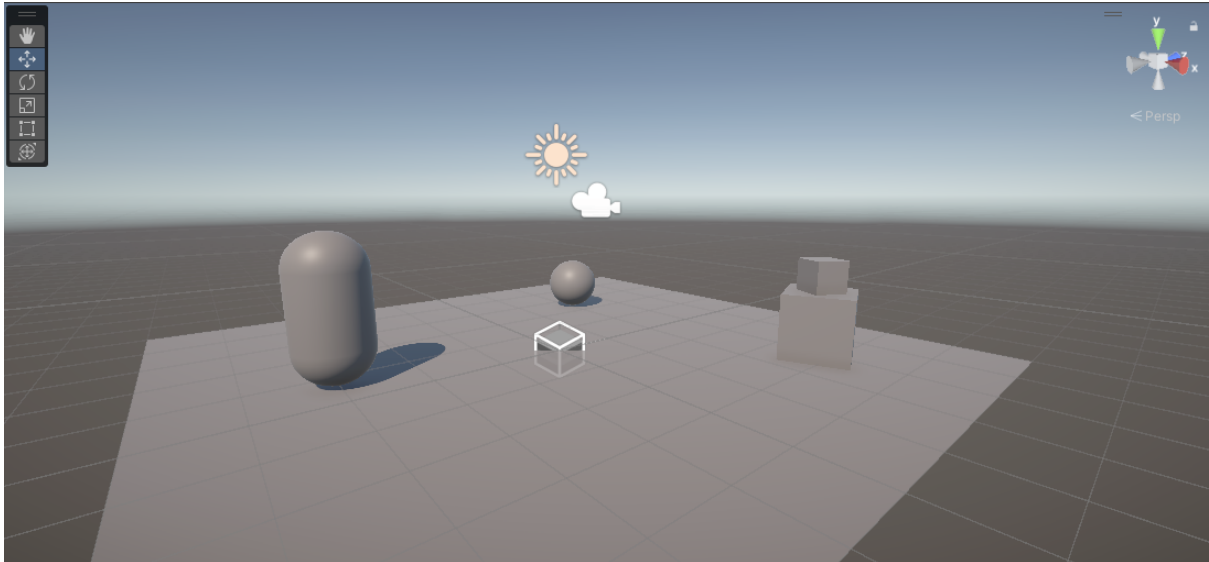
Obiekt kamery **Main Camera** otrzymał przy tworzeniu dodatkowy komponent **Tracked Pose Driver (Input System)**. Ten komponent automatycznie aktualizuje pozycję i obrót obiektu kamery na podstawie danych z zestawu HMD: pozycja zmienia się na podstawie akcji <XRHMD>/centerEyePosition, a obrót na podstawie akcji <XRHMD>/centerEyeRotation.



Tak ustawiony projekt jest gotowy do użytku.

3. Zadania

1. Utwórz nowy projekt i ustaw scenę dla twojego urządzenia VR według instrukcji. [1 pkt]
 2. Dodaj kilka dowolnych obiektów dookoła kamery i rozejrzyj się. [1 pkt]
- Przykładowa scena:



3. Korzystając z zaimportowanych zasobów startowych, możesz zaimportować gotowy XR Origin zamiast tworzyć nowy od zera. Przeciągnij prefabrykat **XR Origin (XR Rig)** z folderu **Assets/Samples/XR Interaction Toolkit/3.2.1/Starter Assets/Prefabs** do hierarchii sceny oraz rozpakuj go (opcja **Prefab>Unpack** w menu kontekstowym obiektu) Następnie dodaj do sceny dwie różnokolorowe kule, które będą “przyklejone” do twoich kontrolerów ruchu i będą poruszać się razem z nimi, gdy poruszyysz dłońmi [3 pkt].

4. Źródła

Dokumentacja XR Interaction Toolkit:

<https://docs.unity3d.com/Packages/com.unity.xr.interaction.toolkit@3.0/manual/index.html>