

# Rzeczywistość wirtualna i rozszerzona

Laboratorium nr 3

**Temat:**

**VR w Unity Engine: konfiguracja OpenXR, lokomocja, teleportacja i manipulacja obiektami**

Katedra Mechaniki Stosowanej i Robotyki

Wydział Budowy Maszyn i Lotnictwa, Politechnika Rzeszowska

Liczba zajęć przeznaczona na temat: 2

## 1. Wstęp

Celem ćwiczenia jest przygotowanie działającej sceny VR w Unity na bazie szablonu `High Definition 3D Sample`. W trakcie realizacji ćwiczenia należy skonfigurować `OpenXR`, dodać profil kontrolerów `Oculus Touch Controller Profile`, zainstalować `XR Interaction Toolkit`, poprawić materiały kontrolerów, uruchomić widzenie tunelowe, przygotować teleportację oraz dodać chwytanie i skalowanie obiektu.

Instrukcja została rozpisana krok po kroku i prowadzi czytelnika po ścieżkach menu, nazwach komponentów oraz obiektach w `Hierarchy`. Po każdym zaznaczonym etapie testowym należy uruchomić aplikację i sprawdzić działanie funkcji. Pozwala to wychwycić błędy na wczesnym etapie, zanim zostaną przykryte przez kolejne zmiany.

## 2. Plan realizacji ćwiczenia

**Uwaga organizacyjna:** Ćwiczenie ma charakter praktyczny. Nie przechodź do następnego etapu, dopóki bieżący nie działa poprawnie w trybie `Play`.

**Przebieg ćwiczenia (instruktaż prowadzącego):**

1. Utworzenie projektu na szablonie `High Definition 3D Sample`.
2. Instalacja `XR Plug-in Management`, wybór `OpenXR` i naprawa błędów w `Project Validation`.
3. Dodanie profilu `Oculus Touch Controller Profile`.
4. Instalacja `XR Interaction Toolkit` oraz import `Starter Assets`.
5. Dodanie do sceny `XR Origin` i `Interaction Managera`.
6. Usunięcie zbędnego obiektu `Controllers` i poprawa shaderów kontrolerów.
7. Dodanie prefabów i komponentów odpowiedzialnych za `Tunneling Vignette`.
8. Skonfigurowanie `Teleportation Area` i `Teleportation Anchor`.
9. Dodanie `XR Grab Interactable` oraz `XR General Grab Transformer` do obiektu na scenie.
10. Testowanie działania po każdym ważnym etapie konfiguracji.

## 3. Wymagania zaliczeniowe

1. Projekt został utworzony na bazie `High Definition 3D Sample`, a `OpenXR` jest aktywne w ustawieniach projektu.

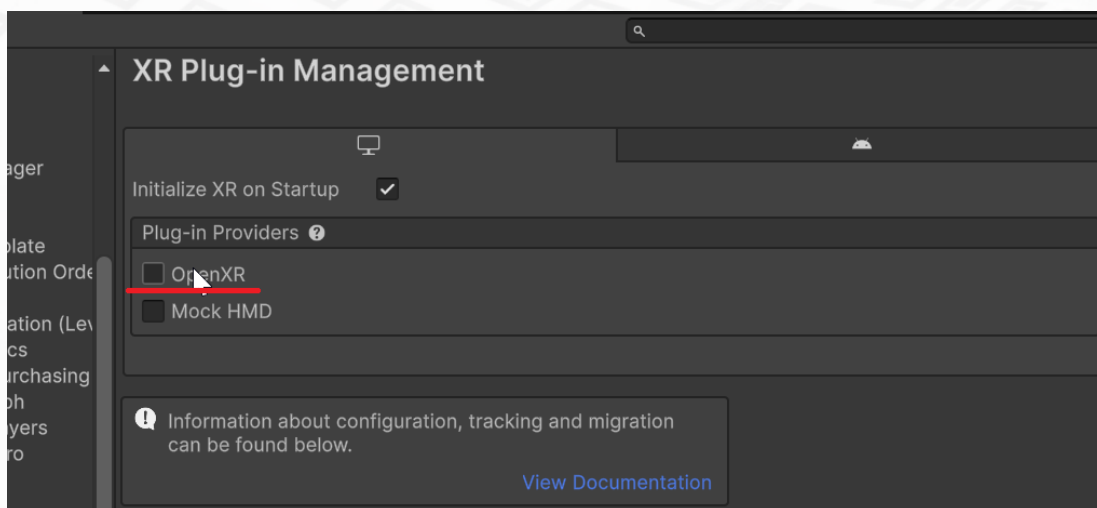
2. Wszystkie błędy z `Project Validation` zostały usunięte, a profil `Oculus Touch Controller Profile` został dodany.
3. W projekcie są zainstalowane `XR Interaction Toolkit` i `Starter Assets`, a scena zawiera `XR Origin` oraz `Interaction Manager`.
4. Kontrolery renderują się poprawnie, a ruch i obrót działają bez błędów.
5. `Tunneling Vignette` działa w czasie ruchu i ogranicza dyskomfort podczas przemieszczania.
6. Teleportacja działa zarówno na `Teleportation Area`, jak i na `Teleportation Anchor`.
7. Obiekt `UnityMaterialBallLarge` można chwytać oraz skalować obiema rękami.

## 4. Przebieg ćwiczenia

### 4.1 Przygotowanie projektu i konfiguracja OpenXR

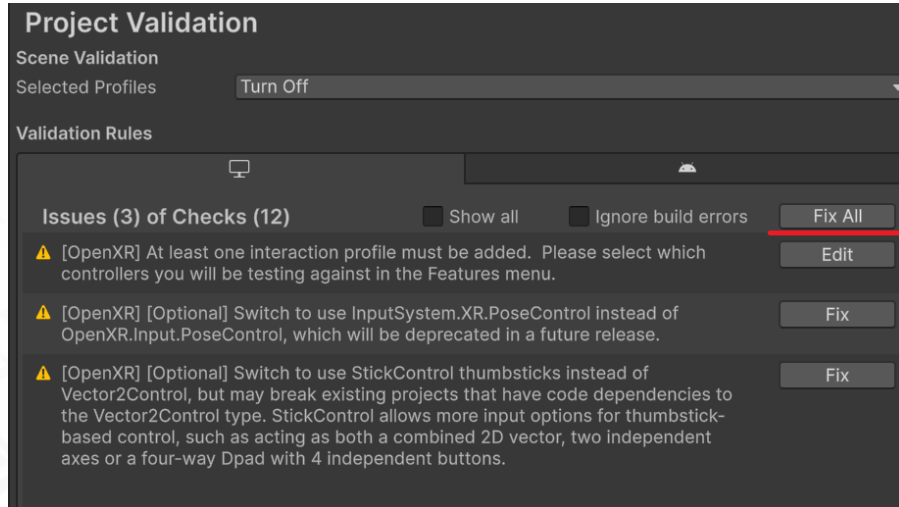
Na tym etapie należy przygotować projekt do pracy z VR i aktywować podstawowe wsparcie `OpenXR`. Jest to najważniejsza konfiguracja startowa, ponieważ błędy popełnione w tym miejscu najczęściej blokują działanie kontrolerów i kolejnych komponentów XR.

- a) Uruchomić Unity i utworzyć nowy projekt na szablonie `High Definition 3D Sample`.
- b) Przejść do `Edit -> Project Settings -> XR Plug-in Management`. Jeżeli komponent nie jest jeszcze zainstalowany, kliknąć `Install XR Plug-in Management`, a następnie zaznaczyć `OpenXR` (patrz rys. 1).



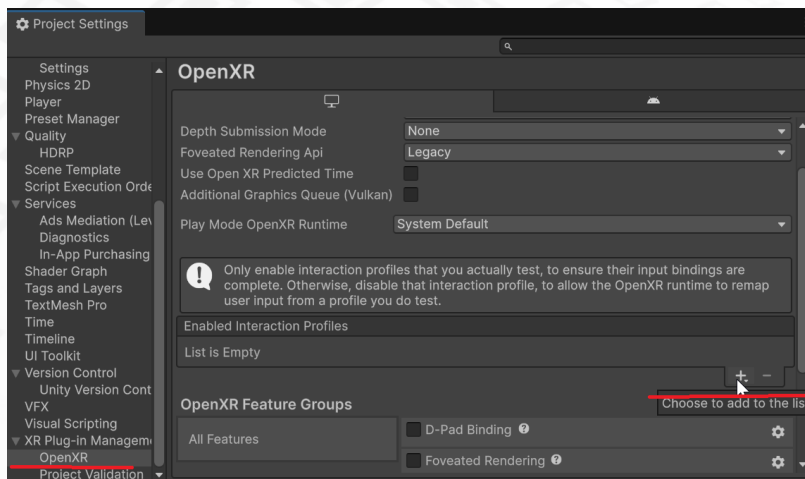
Rys. 1: Zaznaczenie `OpenXR` w ustawieniach `XR Plug-in Management`.

- c) Otworzyć sekcję `Project Validation`. Jeżeli Unity wyświetla błędy lub brakujące zależności, kliknąć `Fix All` (patrz rys. 2). Za każdym razem, gdy to okno pojawi się ponownie, należy usunąć błędy przed przejściem dalej.



Rys. 2: Przycisk Fix All w oknie Project Validation.

- d) W ustawieniach OpenXR dodać profil kontrolerów. Kliknąć przycisk + przy liście profili interakcji, a następnie wybrać Oculus Touch Controller Profile (patrz rys. 3).



a) Dodawanie nowego profilu kontrolerów.



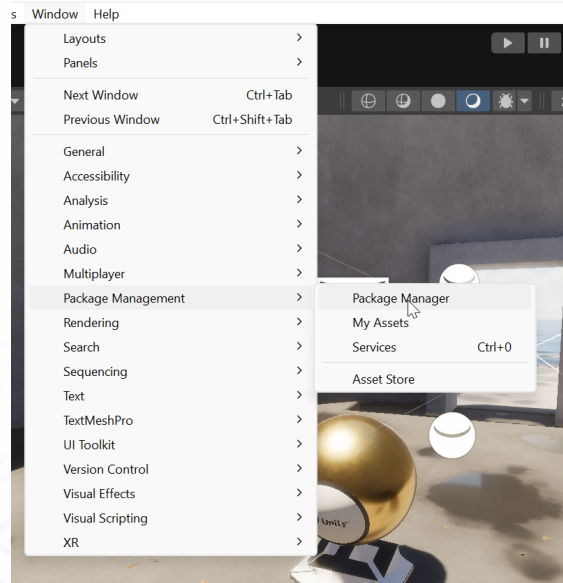
b) Wybór Oculus Touch Controller Profile.

Rys. 3: Dodanie profilu kontrolerów w konfiguracji OpenXR.

## 4.2 Instalacja XR Interaction Toolkit i Starter Assets

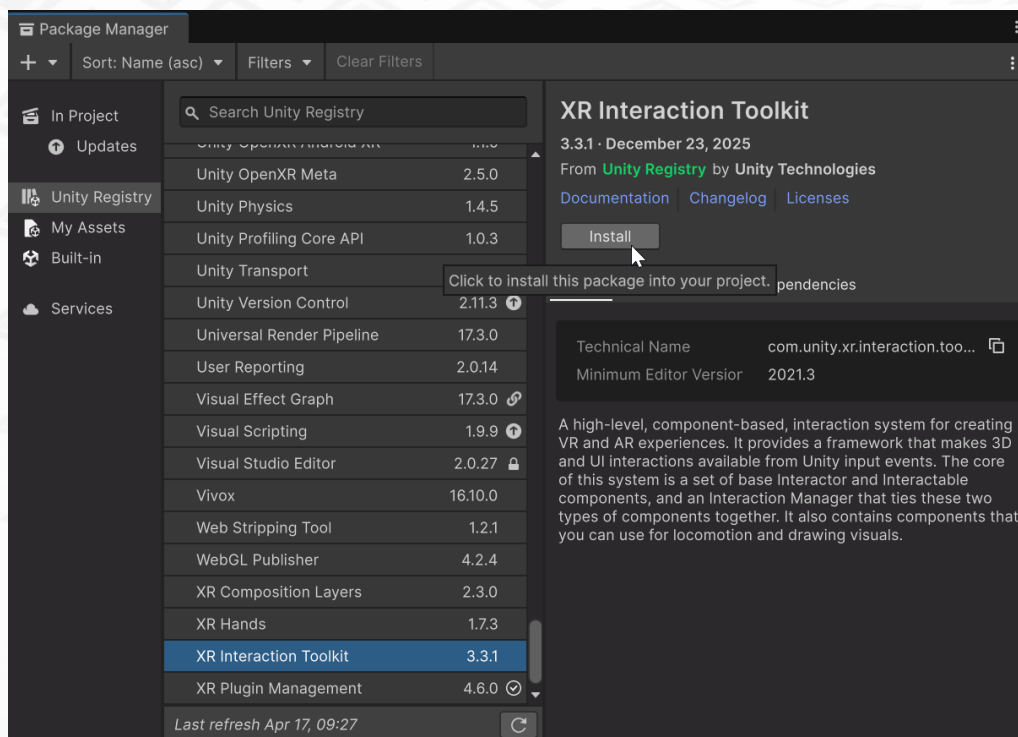
W tej części należy doinstalować pakiety odpowiedzialne za interakcję oraz gotowe zasoby startowe. Dzięki temu w kolejnych krokach możliwe będzie korzystanie z prefabów i komponentów dostarczonych razem z pakietem XR.

- a) Otworzyć Window -> Package Manager (patrz rys. 4). Jest to podstawowa ścieżka do instalacji pakietów XR w Unity.



Rys. 4: Ścieżka do okna Package Manager.

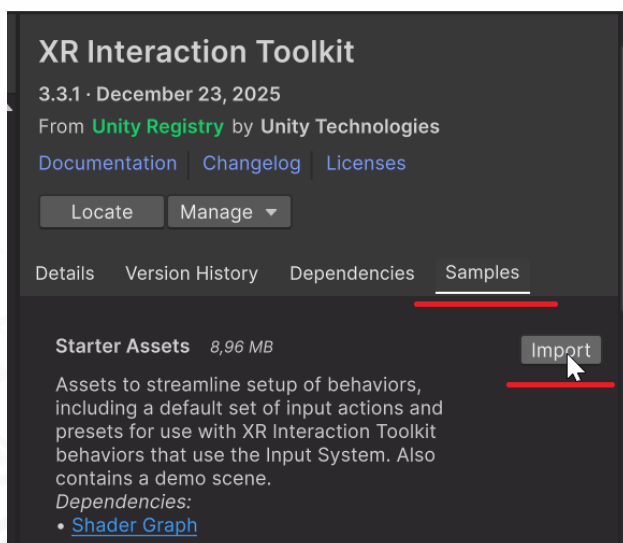
- b) W źródle pakietów wybrać Unity Registry, wyszukać XR Interaction Toolkit i kliknąć Install (patrz rys. 5).



Rys. 5: Pakiet XR Interaction Toolkit i przycisk Install.

- c) Po instalacji przejść do zakładki Samples, odnaleźć Starter Assets i kliknąć Import (patrz

rys. 6). Dzięki temu w projekcie pojawią się potrzebne prefabrykaty i gotowe ustawienia sterowania.

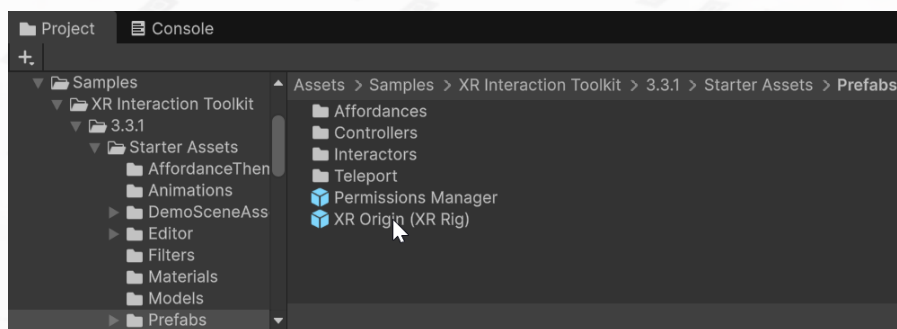


Rys. 6: Import pakietu Starter Assets w XR Interaction Toolkit.

### 4.3 Dodanie XR Origin i podstawowych obiektów sceny

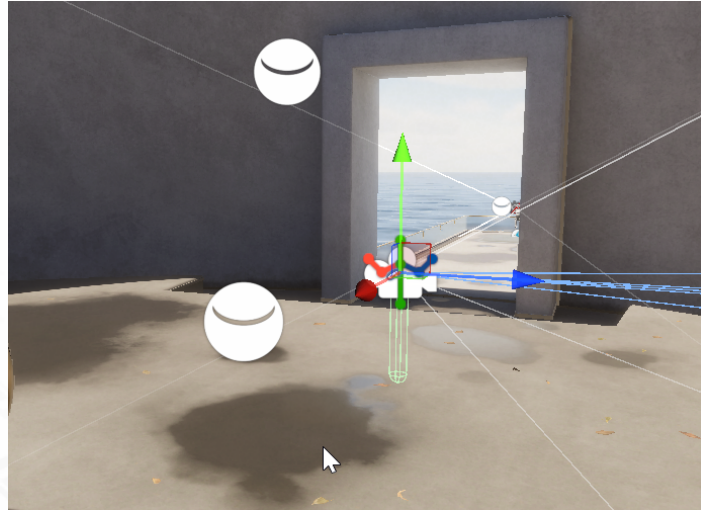
Na tym etapie należy przygotować podstawowy układ obiektów potrzebnych do działania sceny VR. To tutaj scena otrzymuje kamerę gracza, obiekt pochodzenia ruchu oraz menedżera interakcji.

- a) W oknie Project przejść do Starter Assets -> Prefabs i odszukać prefab XR Origin (patrz rys. 7).



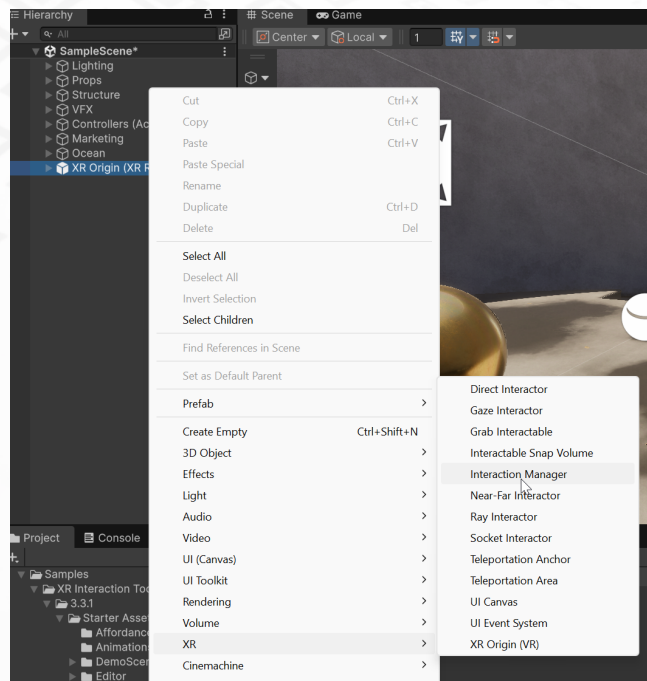
Rys. 7: Położenie prefabu XR Origin w zasobach projektu.

- b) Przeciągnąć XR Origin na scenę w pobliżu dużej złotej kuli (patrz rys. 8). Po dodaniu sprawdzić, czy obiekt pojawił się w Hierarchy.



Rys. 8: Scena po przeciągnięciu obiektu XR Origin.

- c) W Hierarchy kliknąć PPM i wybrać XR -> Interaction Manager (patrz rys. 9). Ten obiekt zarządza interakcjami pomiędzy kontrolerami a obiektami XR.



Rys. 9: Ścieżka dodawania obiektu Interaction Manager.

- d) W drzewie Hierarchy usunąć obiekt Controllers (patrz rys. 10). Pozostawienie domyślnego sterowania może powodować konflikt z nową konfiguracją sceny VR.

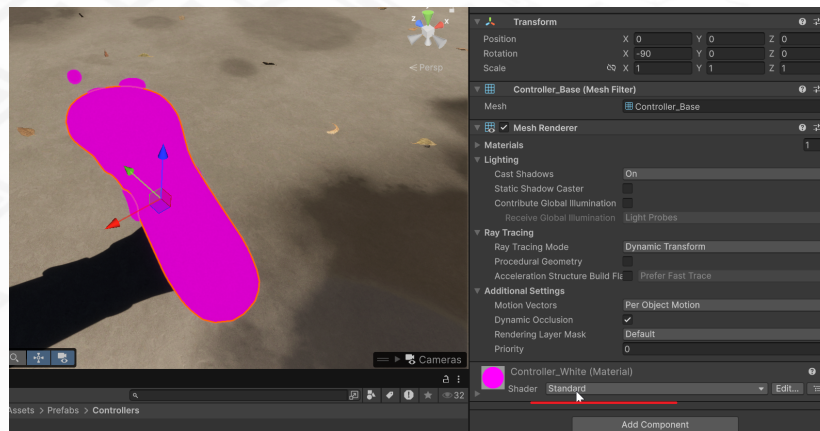


Rys. 10: Obiekt **Controllers** przeznaczony do usunięcia z Hierarchy.

#### 4.4 Poprawa shaderów materiałów kontrolerów

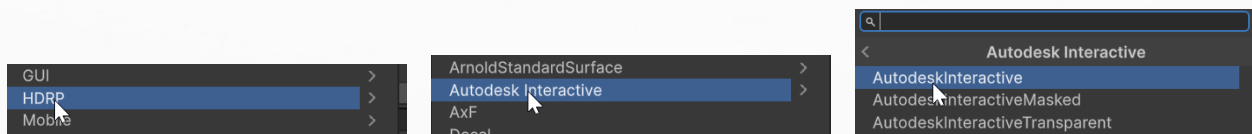
W tym etapie należy usunąć problem niepoprawnego renderowania materiałów kontrolerów. Dzięki temu kontrolery będą czytelne zarówno w edytorze, jak i po uruchomieniu aplikacji w goglach VR.

- a) Przybliżyć kamerę edytora do kontrolerów i sprawdzić, czy któryś z nich nie ma różowego materiału. Jeżeli tak się stało, zaznaczyć materiał kontrolera i kliknąć pole **Shader** w **Inspectorze** (patrz rys. 11).



Rys. 11: Pole **Shader** w **Inspectorze** materiału kontrolera.

- b) Dla każdego materiału z różową teksturą ustawić shader HDRP -> **Autodesk Interactive** -> **AutodeskInteractive** (patrz rys. 12). Zmianę należy wykonać dla wszystkich źle renderowanych elementów kontrolera.



- a) Otworzenie listy shaderów.
- b) Przejście do gałęzi **Autodesk Interactive**.
- c) Wybór **AutodeskInteractive**.

Rys. 12: Wybór poprawnego shadera dla materiału kontrolera w HDRP.

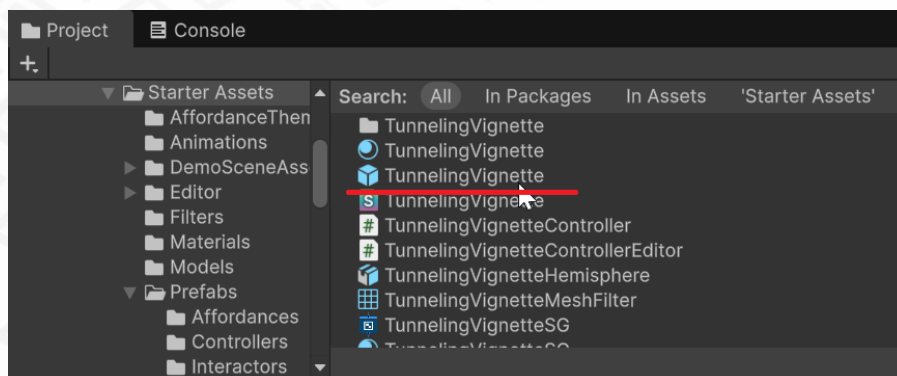
**TEST APLIKACJI: Uruchomić Play i sprawdzić, czy kontrolery renderują się poprawnie oraz czy ruch i obrót działają z joysticków. Jeżeli coś nie działa, należy zatrzymać pracę i poprawić błąd przed przejściem dalej.**

Na tym etapie najczęściej wychodzą błędy związane z OpenXR, brakującym profilem kontrolerów lub niepoprawnym shaderem materiału.

#### 4.5 Dodanie i konfiguracja Tunneling Vignette

W tej części należy dodać widzenie tunelowe, które ogranicza dyskomfort i zmniejsza ryzyko cybersickness podczas ruchu. Efekt powinien uaktywniać się w czasie lokomocji, a nie stale przez cały czas.

- a) W oknie Project odszukać prefab `TunnelingVignette` (patrz rys. 13). Najwygodniej zrobić to przez wyszukiwarkę w folderze `Starter Assets`.



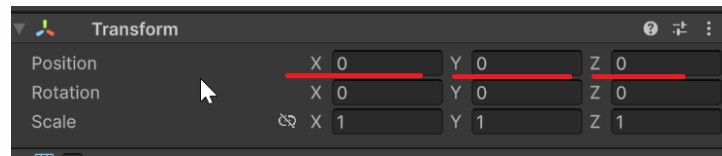
Rys. 13: Położenie prefabu `TunnelingVignette` w zasobach projektu.

- b) Przeciągnąć prefab `TunnelingVignette` jako dziecko obiektu kamery, czyli pod `Main Camera` w Hierarchy (patrz rys. 14). Dzięki temu winieta będzie poruszać się razem z kamerą gracza.



Rys. 14: Prefab `TunnelingVignette` podpięty jako dziecko obiektu `Main Camera`.

- c) Zaznaczyć obiekt `TunnelingVignette` i w komponencie `Transform` wyzerować pozycję  $X = 0$ ,  $Y = 0$ ,  $Z = 0$  (patrz rys. 15). Pozwoli to ustawić obiekt dokładnie przed kamerą.



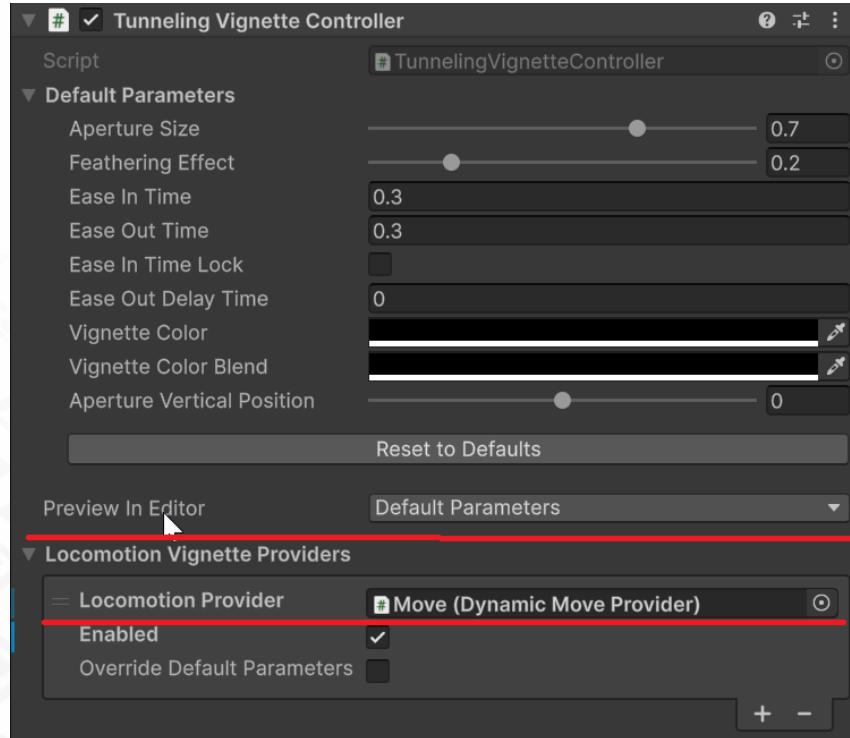
Rys. 15: Komponent `Transform` obiektu `TunnelingVignette` z wyzerowaną pozycją.

- d) Sprawdzić w widoku sceny, czy obiekt `TunnelingVignette` znajduje się poprawnie przed kamerą (patrz rys. 16).



Rys. 16: Prawidłowe położenie obiektu `TunnelingVignette` względem kamery.

- e) W komponencie `Tunneling Vignette Controller` dodać dostawcę ruchu w sekcji `Locomotion Vignette Providers`. W polu `Locomotion Provider` ustawić `Move (Dynamic Move Provider)` i włączyć `Preview In Editor` (patrz rys. 17).



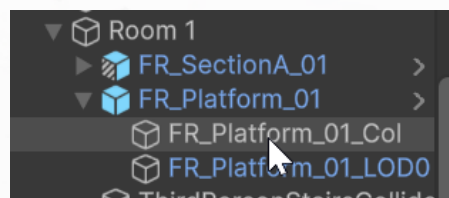
Rys. 17: Konfiguracja komponentu Tunneling Vignette Controller.

**TEST APLIKACJI: Uruchomić Play i sprawdzić, czy podczas ruchu pojawia się widzenie tunelowe. Jeżeli efekt nie działa, najpierw należy sprawdzić pozycję obiektu TunnelingVignette oraz pole Locomotion Provider.**

#### 4.6 Konfiguracja Teleportation Area

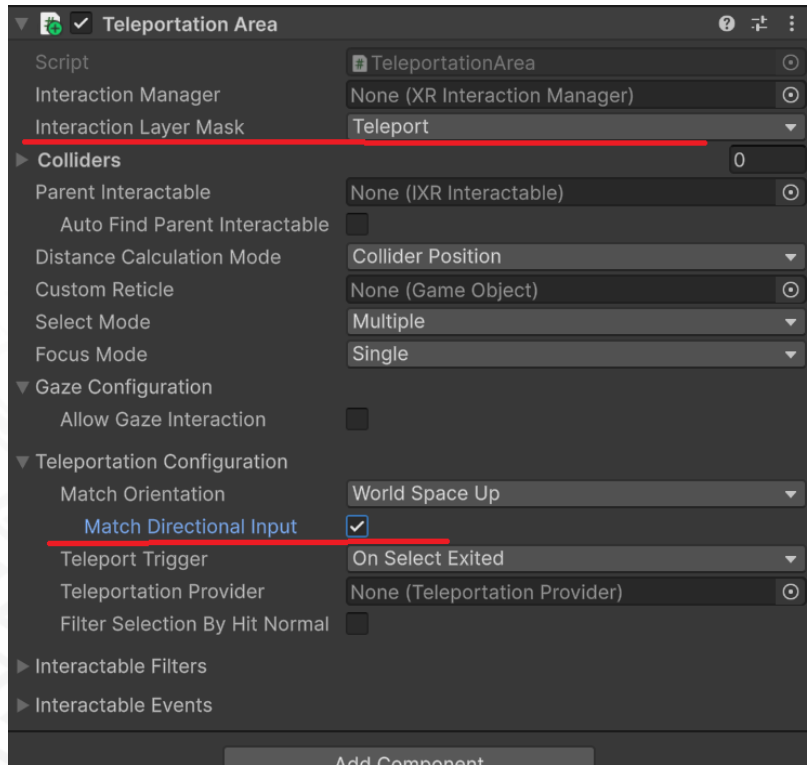
Na tym etapie należy uruchomić teleportację na dużej powierzchni sceny. Najpierw trzeba skonfigurować obszar, po którym gracz może bezpiecznie przemieszczać się przy użyciu łuku teleportacyjnego.

- a) W Hierarchy rozwinąć Room 1 -> FR\_Platform\_01 i zaznaczyć obiekt FR\_Platform\_01\_Col (patrz rys. 18).



Rys. 18: Obiekt FR\_Platform\_01\_Col w drzewie Hierarchy.

- b) W Inspectorze dodać komponent Teleportation Area. Następnie ustawić Interaction Layer Mask = Teleport oraz zaznaczyć opcję Match Directional Input (patrz rys. 19).



Rys. 19: Konfiguracja komponentu Teleportation Area.

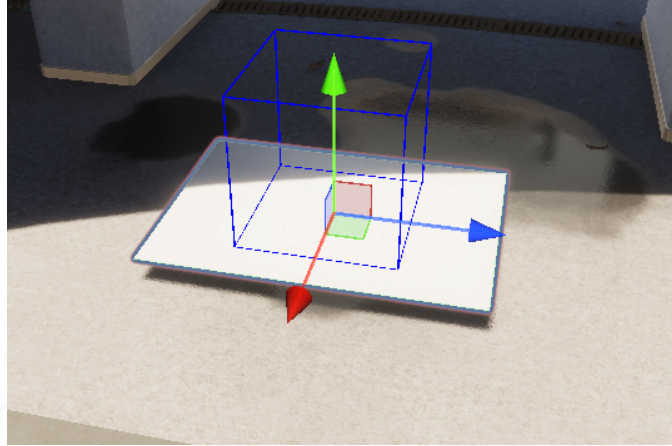
**TEST APLIKACJI: Uruchomić Play i sprawdzić teleportację. Wchylić joystick odpowiedzialny za teleport, poczekać na pojawienie się łuku, wskazać platformę i zatwierdzić teleport zwolnieniem joysticka. Jeżeli teleport nie działa, nie należy przechodzić dalej, dopóki nie zostanie poprawiona konfiguracja pola teleportacji.**

Opcja Match Directional Input pozwala od razu ustawić kierunek patrzenia po teleportacji, dlatego warto zweryfikować również orientację gracza po przeniesieniu.

#### 4.7 Konfiguracja Teleportation Anchor

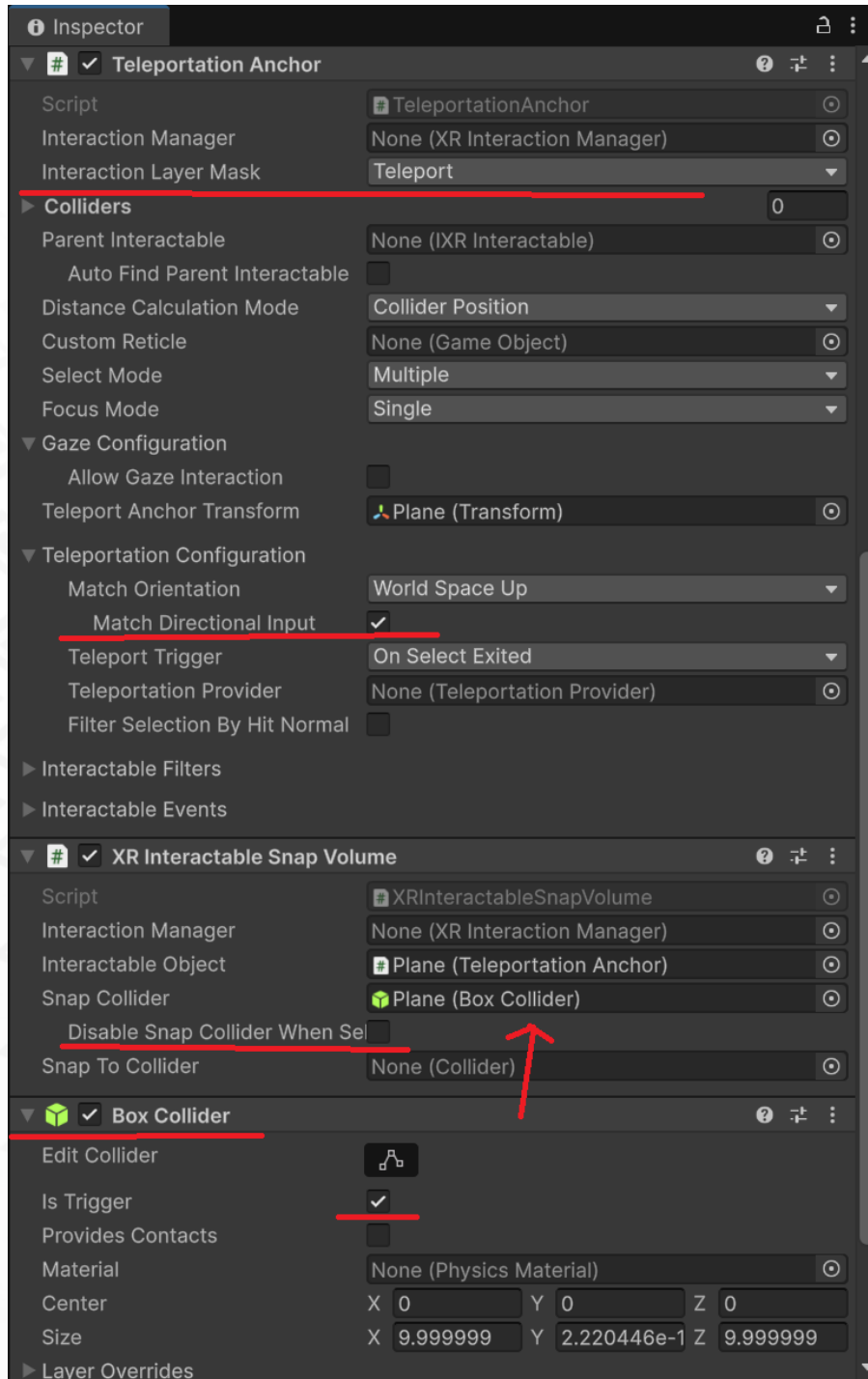
W tej części należy przygotować teleportację do konkretnego, z góry wyznaczonego miejsca. Anchor przydaje się wtedy, gdy gracz ma lądować dokładnie na małej platformie lub w precyzyjnie określonym punkcie sceny.

- a) Dodać do sceny nowy obiekt Plane przez Hierarchy -> PPM -> 3D Object -> Plane. Zmienić jego skalę tak, aby przypominał niewielką platformę teleportacyjną (patrz rys. 20).



Rys. 20: Platforma przygotowana jako cel teleportacji.

- b) Do obiektu Plane dodać komponent Teleportation Anchor. Ustawić Interaction Layer Mask = Teleport oraz zaznaczyć Match Directional Input (patrz rys. 21).
- c) Dodać także komponenty XR Interactable Snap Volume oraz Box Collider. W Box Colliderze zaznaczyć Is Trigger. W XR Interactable Snap Volume przypisać box collider do pola Snap Collider i odznaczyć Disable Snap Collider When Selected (patrz rys. 21).



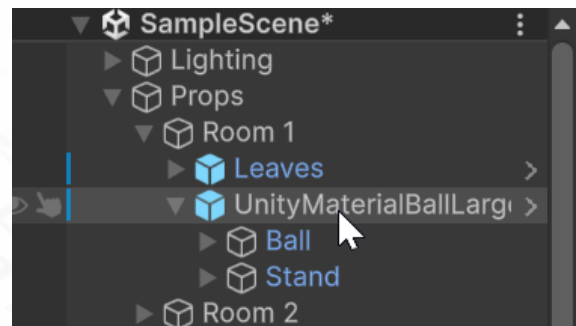
Rys. 21: Konfiguracja komponentów Teleportation Anchor, XR Interactable Snap Volume oraz Box Collider.

**TEST APLIKACJI:** Uruchomić Play i sprawdzić, czy można teleportować się na nową platformę. Ten etap jest krytyczny, bo błędy w Interaction Layer Mask, Is Trigger lub Snap Collider od razu blokują teleportację.

## 4.8 Manipulacja obiektem i skalowanie dwuręczne

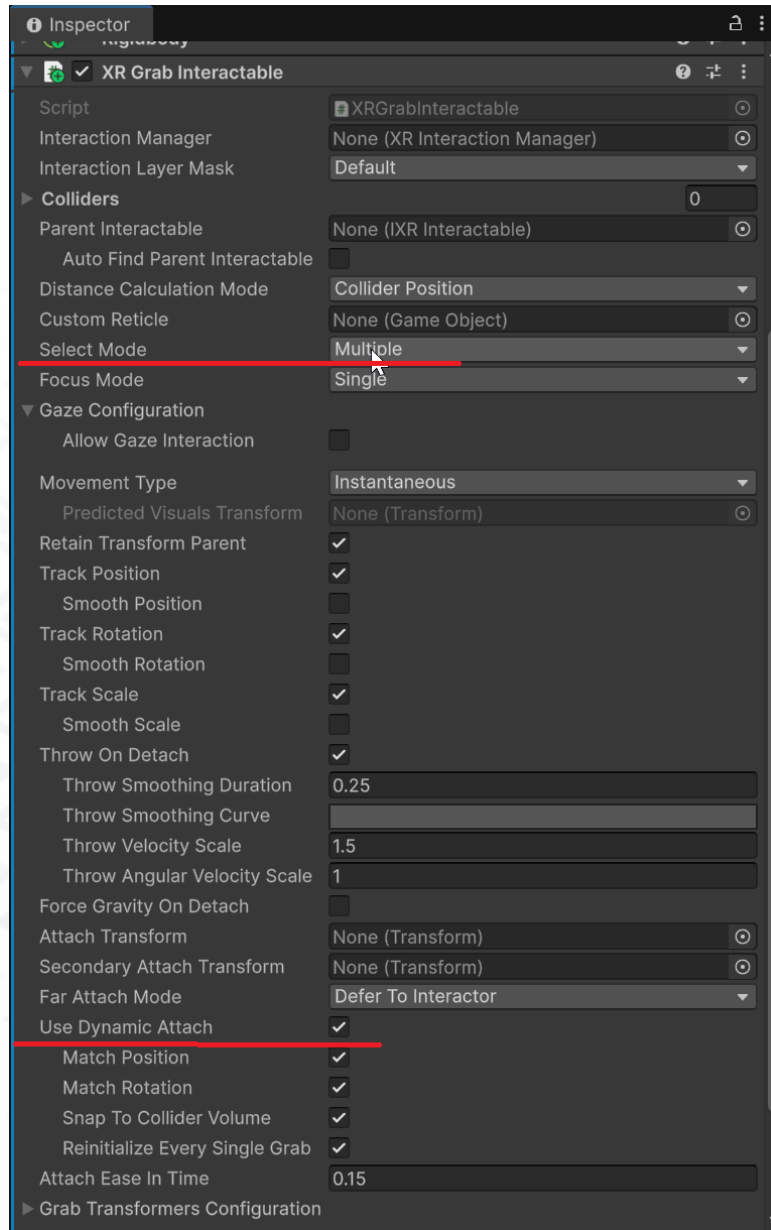
Na końcu należy skonfigurować chwytanie i skalowanie obiektu obiema rękami. Jest to etap, który łączy klasyczną interakcję XR z transformacją obiektu w czasie rzeczywistym.

- a) W Hierachy zaznaczyć obiekt `UnityMaterialBallLarge` znajdujący się w gałęzi `Props` -> `Room 1`. Jest to duża złota kula, na której będą testowane interakcje (patrz rys. 22).



Rys. 22: Obiekt `UnityMaterialBallLarge` w drzewie Hierachy.

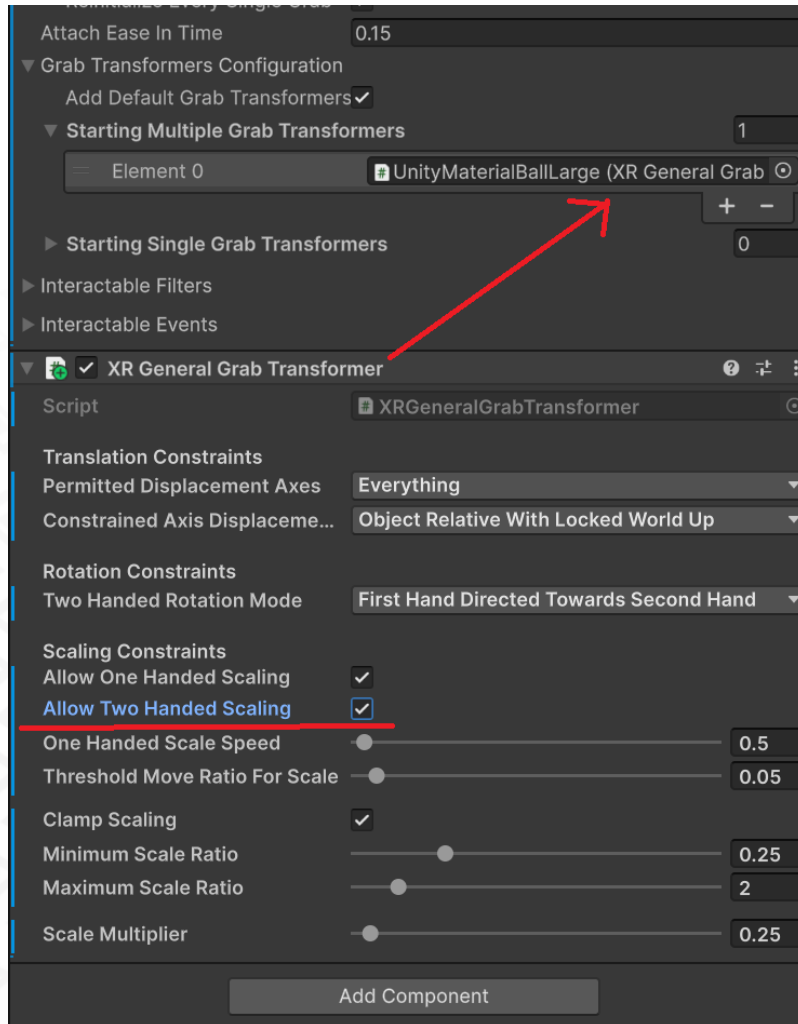
- b) Dodać komponent `XR Grab Interactable`. Ustawić `Select Mode = Multiple` i zaznaczyć opcję `Use Dynamic Attach` (patrz rys. 23). Dzięki temu obiekt będzie można chwycić więcej niż jedną ręką i bardziej naturalnie ustawiać punkt chwytu.



Rys. 23: Konfiguracja komponentu XR Grab Interactable.

**TEST APLIKACJI: Uruchomić Play i sprawdzić, czy złotą kulę można chwycić kontrolerem. Jeżeli chwytanie nie działa, najpierw należy sprawdzić komponent XR Grab Interactable, collider obiektu oraz ustawienie Select Mode.**

- c) Dodać komponent XR General Grab Transformer. W nim zaznaczyć opcję Allow Two Handed Scaling. Następnie wrócić do komponentu XR Grab Interactable i w sekcji Grab Transformers Configuration -> Starting Multiple Grab Transformers dodać przed chwilą utworzony transformer (patrz rys. 24).



Rys. 24: Konfiguracja XR General Grab Transformer i podpięcie go do XR Grab Interactable.

**TEST APLIKACJI: Uruchomić Play i sprawdzić, czy obiekt można skalować obiema rękami. To ostatni etap, na którym trzeba wychwycić błędy konfiguracji transformera przed oddaniem ćwiczenia.**

## 5. Zadania do wykonania

1. Wykonać pełną konfigurację sceny VR zgodnie z instrukcją: OpenXR, profil kontrolerów, XR Interaction Toolkit, XR Origin, Tunneling Vignette, Teleportation Area, Teleportation Anchor oraz manipulację dużą złotą kulą.
2. Rozbudować scenę o dodatkowe elementy:
  - a) dodać drugą platformę teleportacyjną z komponentem Teleportation Anchor,
  - b) dodać drugi obiekt możliwy do chwytania,
  - c) skonfigurować dla nowego obiektu skalowanie dwuręczne.

## OCENA

### Student dostaje:

- ocenę **5** za poprawne wykonanie konfiguracji podstawowej oraz rozbudowanie sceny o dodatkową platformę teleportacji i dodatkowy obiekt możliwy do chwytania i skalowania.

## Uwagi końcowe

W rozwiązywaniu zadań można posiłkować się materiałami dydaktycznymi, dokumentacją Unity oraz dokumentacją Blender:

- <https://docs.blender.org/manual/en/latest/modeling/index.html>
- <https://docs.unity3d.com/Manual/index.html>