

Rzeczywistość wirtualna i rozszerzona

Laboratorium nr 6

Temat:

Budowa aplikacji VR w Unity Engine z wykorzystaniem zasobów z poprzednich laboratoriów

Katedra Mechaniki Stosowanej i Robotyki

Wydział Budowy Maszyn i Lotnictwa, Politechnika Rzeszowska

Liczba zajęć przeznaczona na temat: 3

1. Wstęp

Jest to ostatnie ćwiczenie laboratoryjne z cyklu VR. Student buduje kompletną aplikację VR w Unity Engine, korzystając z assetów przygotowanych i zaimportowanych do Unity podczas laboratoriów VR1–VR5. W tym ćwiczeniu **nie korzysta się już z programu Blender**. Mieszkanie, modele, materiały i tekstury mają znajdować się w poprzednim projekcie Unity i przed rozpoczęciem nowej aplikacji muszą zostać wyeksportowane jako pakiet assetów Unity. Aplikacja ma przedstawiać mieszkanie, zostać oświetlona, uzupełniona o dźwięki i dodatkowe obiekty, a następnie zbudowana jako plik wykonywalny `.exe`.

W laboratorium należy wykorzystać:

- umiejętność obsługi edytora Unity z laboratorium VR1,
- skrypty i sposób organizacji projektu poznane podczas wcześniejszych zajęć,
- materiały i tekstury, które zostały już zaimportowane do Unity we wcześniejszych laboratoriach,
- model mieszkania, okna, drzwi, fotel i stół z laboratorium VR5, ale wyłącznie w postaci assetów znajdujących się już w Unity,
- poprzedni projekt Unity, z którego przed utworzeniem nowej aplikacji zostanie wyeksportowany pakiet `.unitypackage`.

Celem końcowym jest aplikacja VR w trybie HDR/HDRP, w której użytkownik może wejść do mieszkania, obejrzeć scenę, usłyszeć tło dźwiękowe oraz wskazać drzwi kontrolerem VR, otrzymać krótką wibrację i triggerem otworzyć albo zamknąć jedne drzwi.

Powiązanie z poprzednimi laboratoriami

Instrukcja została przygotowana tak, aby można było wykonać ćwiczenie bez otwierania poprzednich plików. Poszczególne kroki wykorzystują jednak efekty wcześniejszej pracy:

- **VR1** — obsługa edytora Unity, okien **Project**, **Hierarchy**, **Inspector**, **Scene**, **Game** oraz testowanie **Play mode**,
- **VR2** — organizacja skryptów `C#` w projekcie, przypisywanie komponentów w **Inspector** i korzystanie z gotowych skryptów prowadzącego,
- **VR3–VR4** — tworzenie materiałów, tekstur, map normalnych, import modeli i przypisywanie materiałów w Unity,
- **VR5** — przygotowanie mieszkania oraz wcześniejsze przeniesienie go do Unity; w VR6 używany jest już tylko pakiet assetów Unity, bez ponownego otwierania Blendera.

2. Materiały wejściowe z poprzednich laboratoriów

Przed rozpoczęciem pracy należy przygotować komplet zasobów:

1. poprzedni projekt Unity, w którym znajdują się materiały, tekstury, modele mieszkania i testowana aplikacja VR,
2. pakiet assetów Unity `VR6_PreviousAssets.unitypackage` wyeksportowany z poprzedniego projektu przed rozpoczęciem nowej aplikacji,
3. materiały i tekstury znajdujące się już w Unity, np. tynk, drewno, szkło, metal, tkanina, granit,
4. pobrane z Internetu obiekty wyposażenia: lampy, obrazy, szafki, półki, rośliny lub dekoracje,
5. uprzednio dostarczony skrypt sterowania aplikacją VR,
6. skrypt prowadzącego do wskazywania drzwi promieniem Raycast, wywoływania wibracji kontrolera i obracania obiektu `DoorPivot` po naciśnięciu triggera.

Pobierz plik programu: [Pobierz Raycast.cs](#)

Skrypt do interakcji drzwi jest dołączony do instrukcji jako załącznik PDF, tak jak skrypty w laboratorium VR2. W instrukcji zakłada się, że działa on w oparciu o promień Raycast wysyłany z kontrolera VR, wibrację kontrolera po wskazaniu drzwi oraz kliknięcie triggerem. Zadaniem studenta jest poprawne przygotowanie obiektu `DoorPivot`, dodanie collidera do drzwi i uzupełnienie pól skryptu w Inspectorze.

3. Eksport assetów z poprzedniego projektu Unity

Pierwszym obowiązkowym krokiem jest przygotowanie pakietu assetów z poprzedniego projektu Unity. Należy to zrobić **zanim student zacznie tworzyć nową aplikację**. W tym laboratorium nie wykonuje się już żadnych operacji w Blenderze i nie eksportuje się ponownie plików FBX. Wszystkie potrzebne modele, materiały i tekstury powinny być już obecne w Unity.

Instrukcja eksportu pakietu:

1. Uruchom Unity Hub.
2. Otwórz poprzedni projekt Unity, w którym testowano materiały, modele mieszkania oraz aplikację VR.
3. W oknie Project odszukaj foldery i zasoby przygotowane wcześniej, np. `Models`, `Materials`, `Textures`, `Prefabs`, `DownloadedAssets`, `Audio` oraz potrzebne skrypty.
4. Zaznacz zasoby potrzebne do nowej aplikacji: mieszkanie, okna, drzwi, fotel, stół, gotowe materiały, tekstury, dodatkowe obiekty wyposażenia i dźwięki.
5. Z menu wybierz `Assets -> Export Package...`

6. W oknie eksportu zaznacz opcję **Include dependencies**, aby Unity dołączyło powiązane materiały, tekstury, prefaby i inne wymagane pliki.
7. Zapisz pakiet pod nazwą `VR6_PreviousAssets.unitypackage` w osobnym folderze, np. `VR6_ExportedAssets`.
8. Po eksporcie zamknij poprzedni projekt. Nowa aplikacja będzie tworzona w nowym projekcie Unity na podstawie tego pakietu.

Kontrola przed przejściem dalej: pakiet `VR6_PreviousAssets.unitypackage` musi istnieć przed utworzeniem nowej aplikacji. Jeżeli w pakiecie brakuje materiałów lub tekstur, należy wrócić do poprzedniego projektu, ponownie zaznaczyć właściwe zasoby i wyeksportować pakiet z opcją **Include dependencies**.

4. Utworzenie nowego projektu VR

Dopiero po przygotowaniu pakietu assetów należy utworzyć nową aplikację VR. Nowy projekt powinien być czysty, ale ma korzystać z assetów wyeksportowanych z poprzedniego projektu Unity.

Instrukcja:

1. W Unity Hub utwórz nowy projekt Unity przeznaczony do aplikacji VR, zgodny z HD, np. projekt HDR/HDRP.
2. Nazwij projekt `VR6_ApartmentVR`.
3. W folderze `Assets` utwórz podfoldery: `Models`, `Materials`, `Textures`, `Scripts`, `Audio`, `Prefabs`, `DownloadedAssets`.
4. Zaimportuj wcześniej przygotowany pakiet: `Assets -> Import Package -> Custom Package...` i wybierz `VR6_PreviousAssets.unitypackage`.
5. W oknie importu zaznacz wszystkie potrzebne zasoby i kliknij **Import**.
6. Utwórz nową scenę i nazwij ją `VR6_Apartment_Final`.
7. Sprawdź, czy w projekcie pojawiły się modele mieszkania, materiały, tekstury, prefaby, dodatkowe obiekty i dźwięki.

Jeżeli po imporcie brakuje assetów, nie należy otwierać Blendera. Należy wrócić do poprzedniego projektu Unity, poprawić eksport pakietu `.unitypackage` i ponownie zaimportować go do nowej aplikacji.

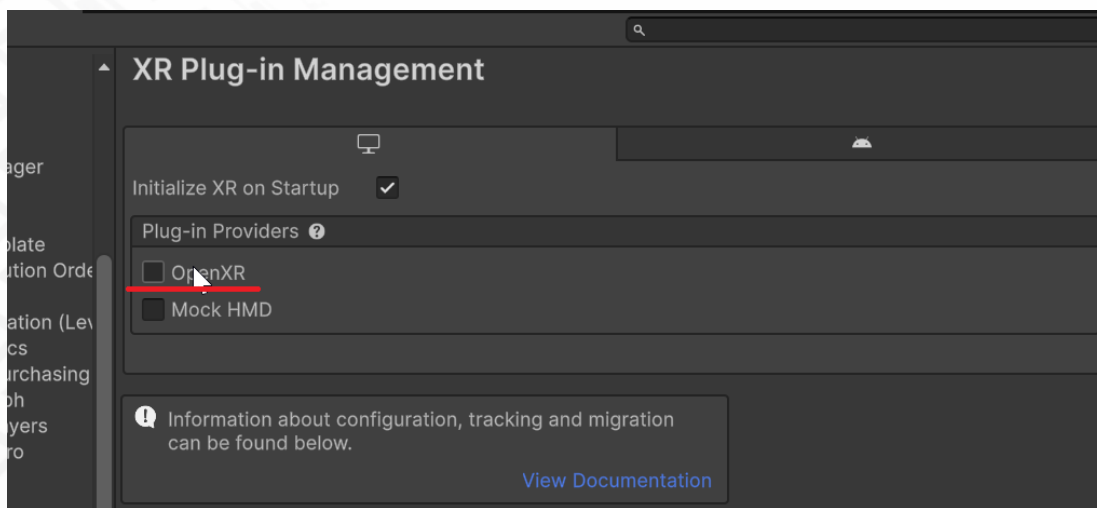
5. Szczegółowa konfiguracja VR z laboratorium VR3

Poniższa część została przeniesiona i dopasowana z laboratorium VR3. Zawiera szczegółowe kroki oraz rysunki potrzebne do odtworzenia konfiguracji VR: `OpenXR`, `XR Interaction Toolkit`, `XR Origin`, kontrolery, widzenie tunelowe, teleportację oraz chwytywanie obiektów. Jeżeli ostatni projekt VR działa poprawnie, sekcję należy potraktować jako listę kontrolną.

5.1 Przygotowanie projektu i konfiguracja OpenXR

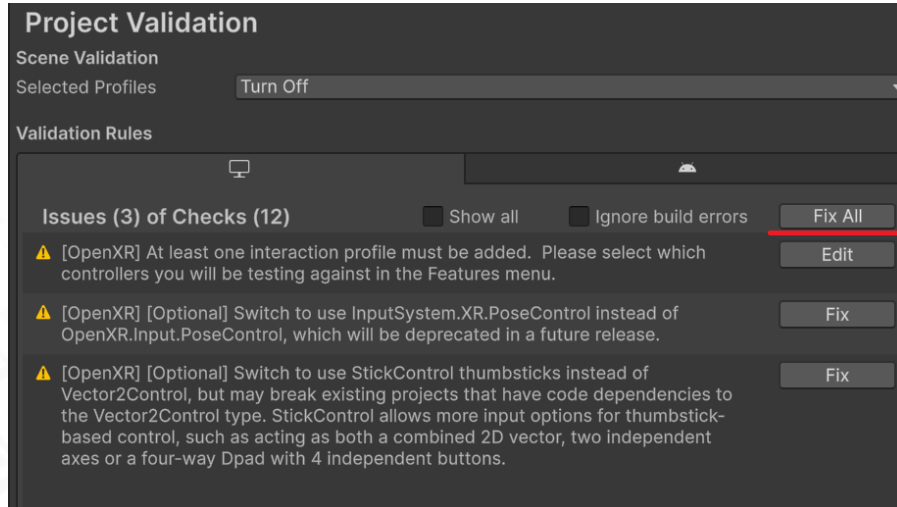
Na tym etapie należy sprawdzić lub odtworzyć konfigurację VR przygotowaną w laboratorium VR3.

- a) Pracować w nowo utworzonym projekcie VR6_ApartmentVR. Jeżeli projekt nie ma jeszcze konfiguracji HDR/HDRP i VR, odtworzyć ustawienia z laboratorium VR3 albo użyć szablonu zgodnego ze stanowiskiem laboratoryjnym.
- 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).



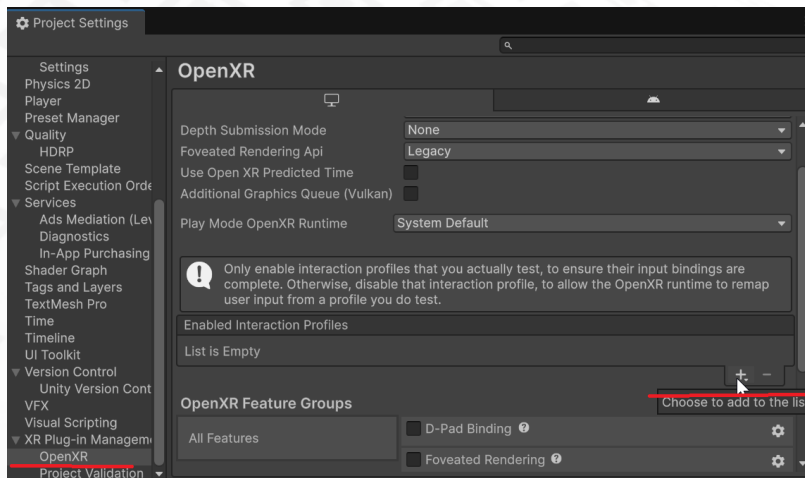
Rys. 1: Zaznaczenie OpenXR w ustawieniach XR Plug-in Management.

- 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.

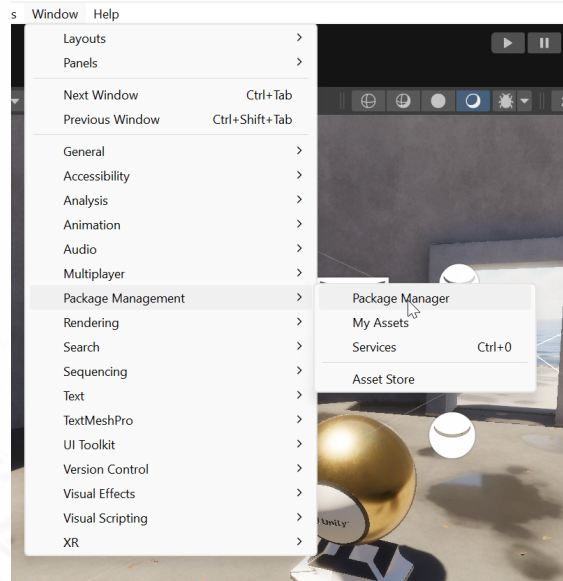
- b) Wybór Oculus Touch Controller Profile.

Rys. 3: Dodanie profilu kontrolerów w konfiguracji OpenXR.

5.2 Instalacja XR Interaction Toolkit i Starter Assets

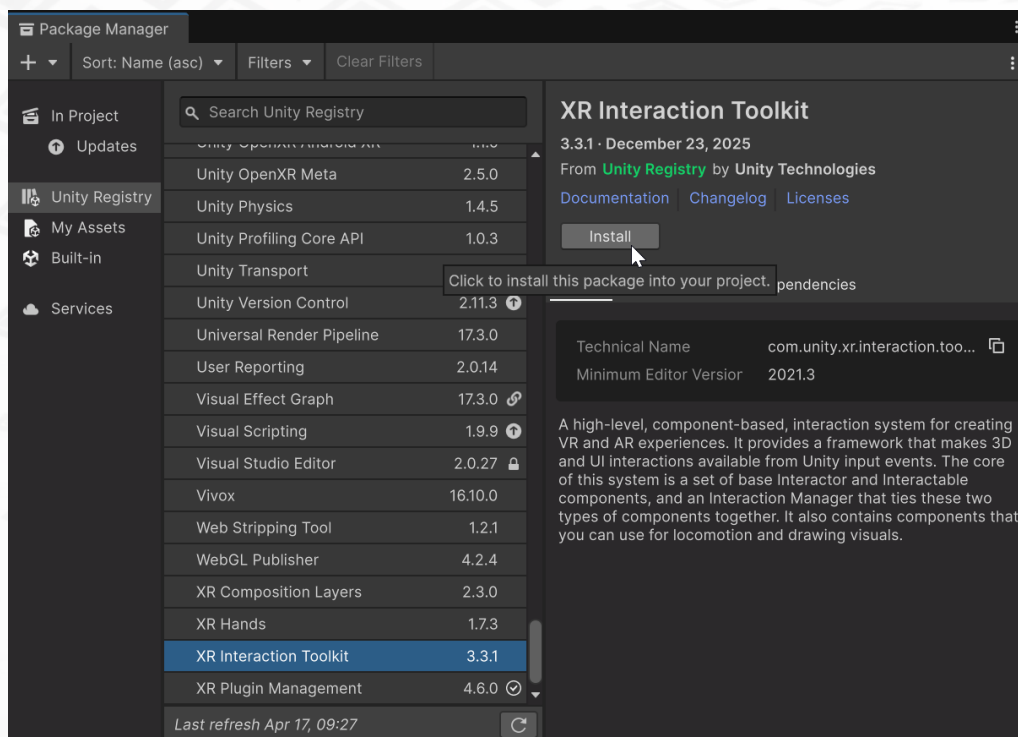
W tej części należy doinstalować pakiety odpowiedzialne za interakcję oraz gotowe zasoby startowe. Dzięki temu w kolejnych krokach możliwe będzie korzystanie z prefabów i komponentów dostarczonych razem z pakietem XR.

- 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.

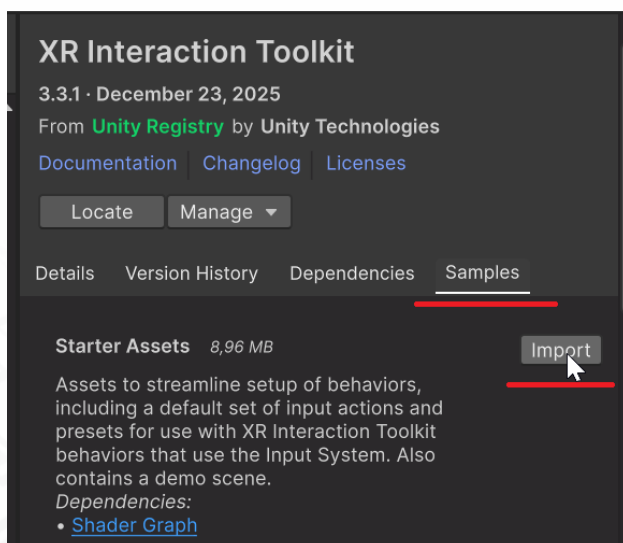
- b) W źródle pakietów wybrać Unity Registry, wyszukać XR Interaction Toolkit i kliknąć Install (patrz rys. 5).



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.

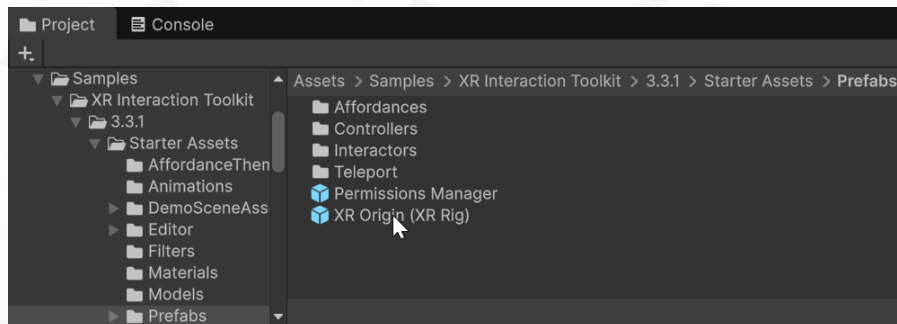


Rys. 6: Import pakietu Starter Assets w XR Interaction Toolkit.

5.3 Dodanie XR Origin i podstawowych obiektów sceny

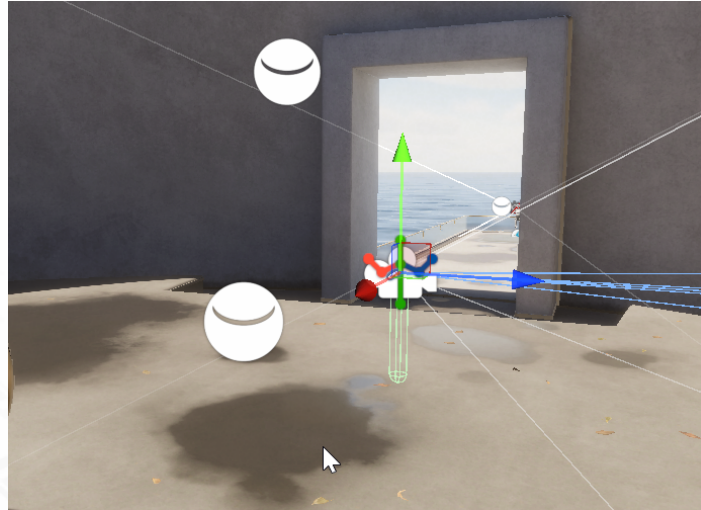
Na tym etapie należy przygotować podstawowy układ obiektów potrzebnych do działania sceny VR. To tutaj scena otrzymuje kamerę gracza, obiekt pochodzenia ruchu oraz menedżera interakcji.

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



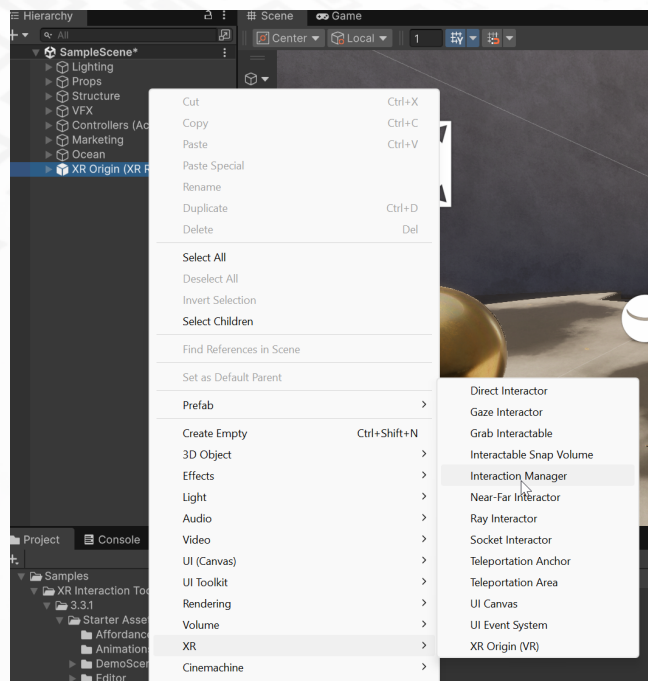
Rys. 7: Położenie prefabu XR Origin w zasobach projektu.

- Przeciagnąć XR Origin na scenę w miejscu startu użytkownika. W mieszkaniu powinien to być korytarz albo pokój wejściowy (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.

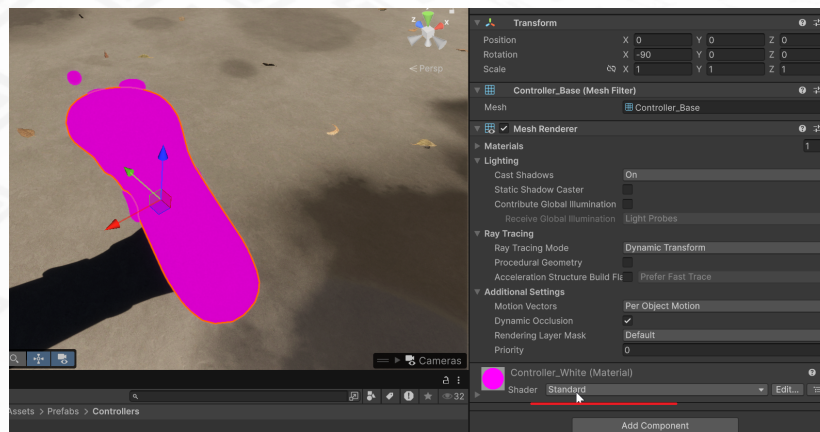


Rys. 10: Obiekt Controllers przeznaczony do usunięcia z Hierarchy.

5.4 Poprawa shaderów materiałów kontrolerów

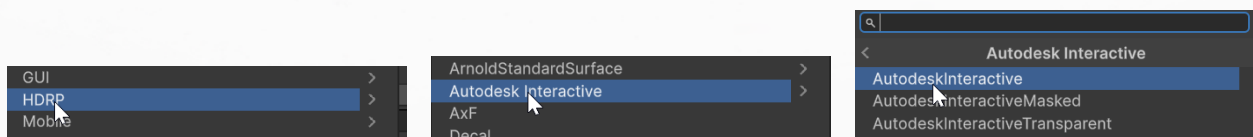
W tym etapie należy usunąć problem niepoprawnego renderowania materiałów kontrolerów. Dzięki temu kontrolery będą czytelne zarówno w edytorze, jak i po uruchomieniu aplikacji w goglach VR.

- 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.

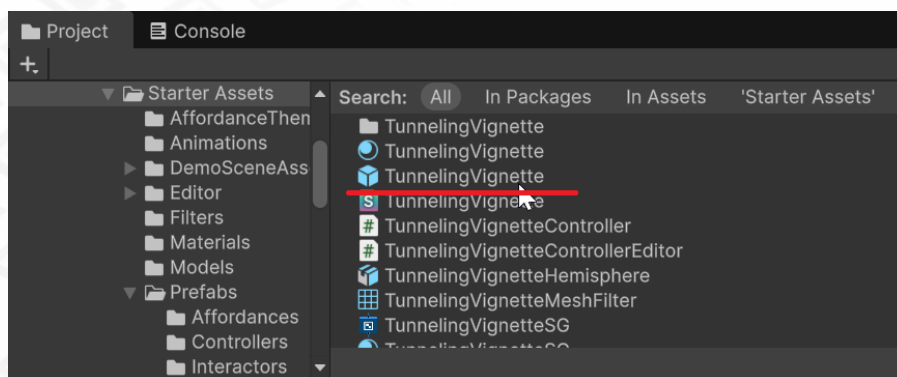
TEST APLIKACJI: Uruchomić Play i sprawdzić, czy kontrolery renderują się poprawnie oraz czy ruch i obrót działają z joysticków. Jeżeli coś nie działa, należy zatrzymać pracę i poprawić błąd przed przejściem dalej.

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

5.5 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.



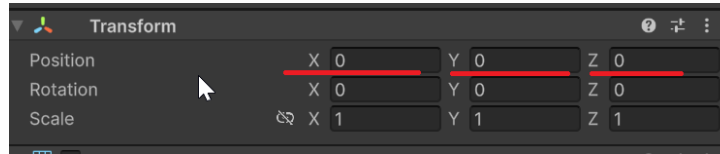
Rys. 13: Położenie prefabu TunnelingVignette w zasobach projektu.

- 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.



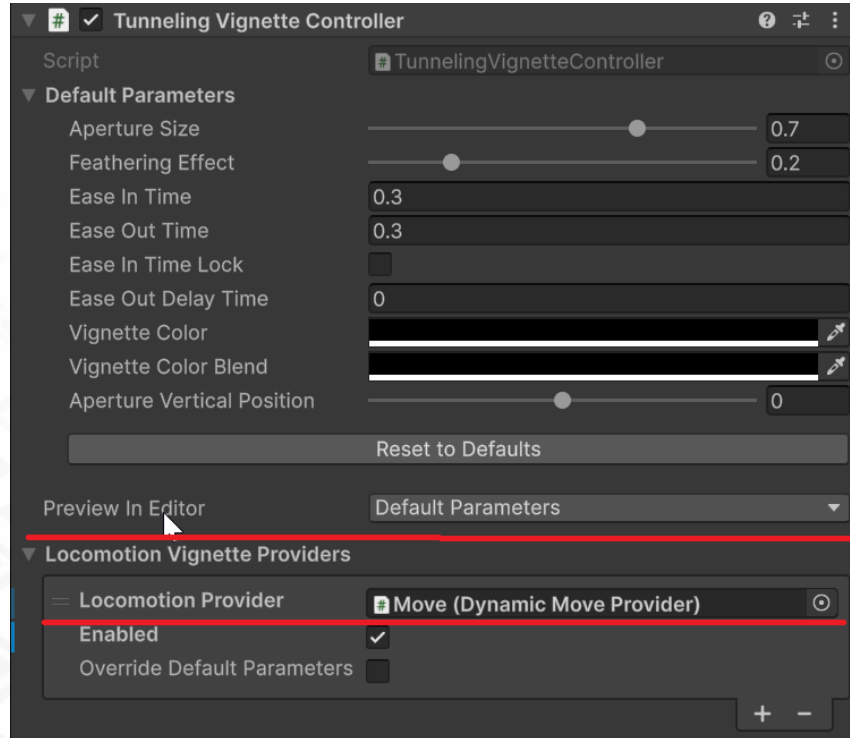
Rys. 15: Komponent Transform obiektu TunnelingVignette z wyzerowaną pozycją.

- 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 polu Locomotion Provider ustawić Move (Dynamic Move Provider) i włączyć Preview In Editor (rys. 17).



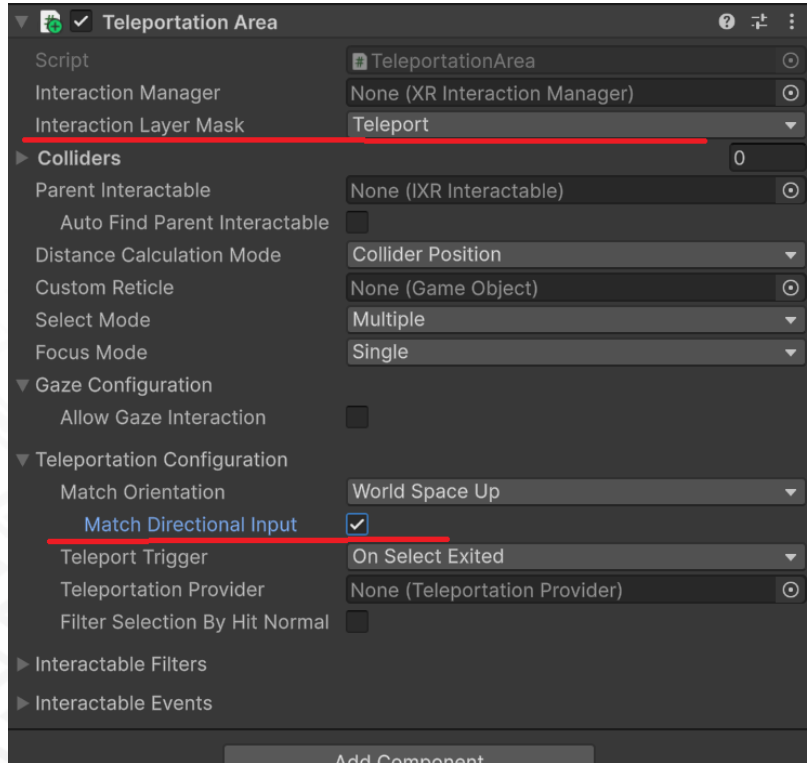
Rys. 17: Konfiguracja komponentu Tunneling Vignette Controller.

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.

5.6 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.

- Zaznaczyć w projekcie dużą powierzchnię (podłogę) na którą będzie można się teleportować.
- W Inspectorze dodać komponent Teleportation Area. Następnie ustawić Interaction Layer Mask = Teleport oraz zaznaczyć opcję Match Directional Input (patrz rys. 18).



Rys. 18: Konfiguracja komponentu Teleportation Area.

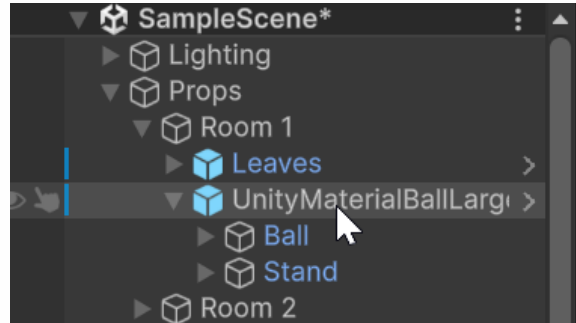
TEST APLIKACJI: Uruchomić Play i sprawdzić teleportację. Wychylić joystick odpowiedzialny za teleport, poczekać na pojawienie się łuku, wskazać platformę i zatwierdzić teleport zwolnieniem joysticka. Jeżeli teleport nie działa, nie należy przechodzić dalej, dopóki nie zostanie poprawiona konfiguracja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

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

5.7 Manipulacja obiektem i skalowanie dwuręczne

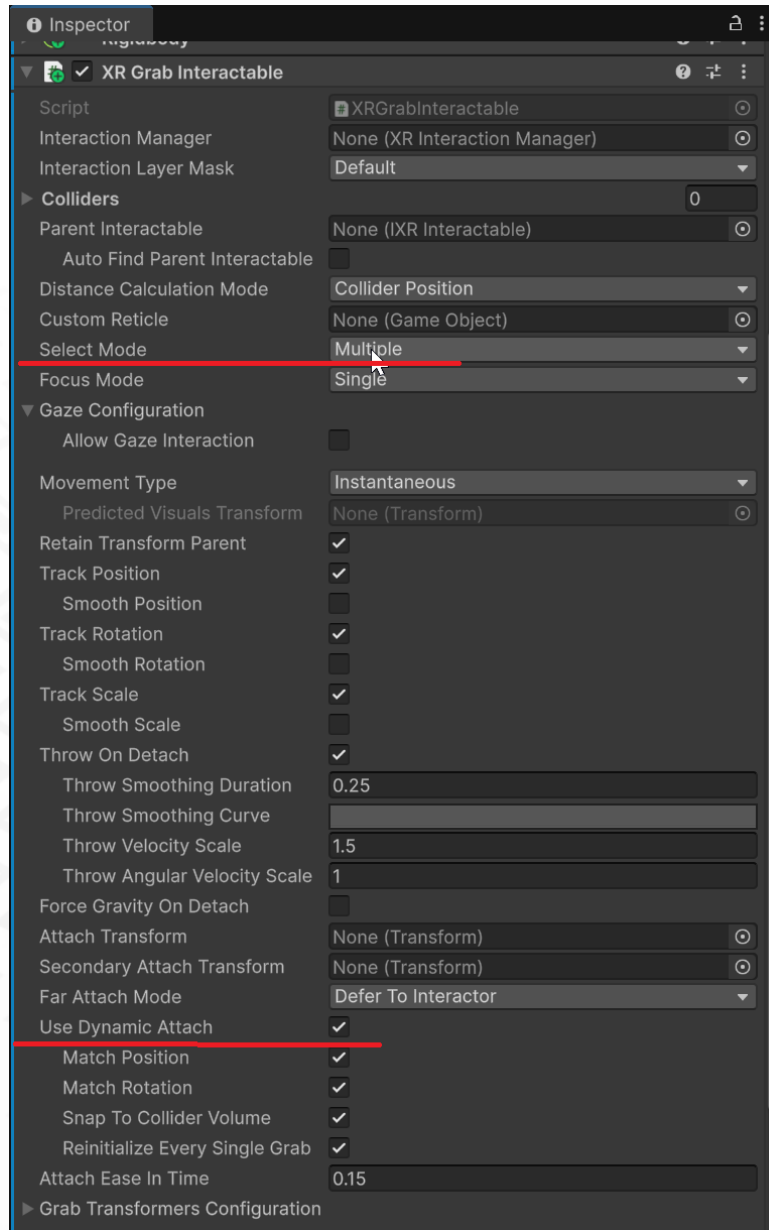
Na końcu należy skonfigurować chwytanie i skalowanie obiektu obiema rękami. W VR3 testowano to na złotej kuli, a w projekcie końcowym tę samą konfigurację należy zastosować do stołu, fotela albo wybranego obiektu pobranego z Internetu.

- a) W Hierarchy zaznaczyć obiekt testowy. W VR3 był to `UnityMaterialBallLarge`; w projekcie końcowym może to być stół, fotel albo wybrany obiekt pobrany z Internetu (patrz rys. 19).



Rys. 19: Obiekt UnityMaterialBallLarge w drzewie Hierarchy.

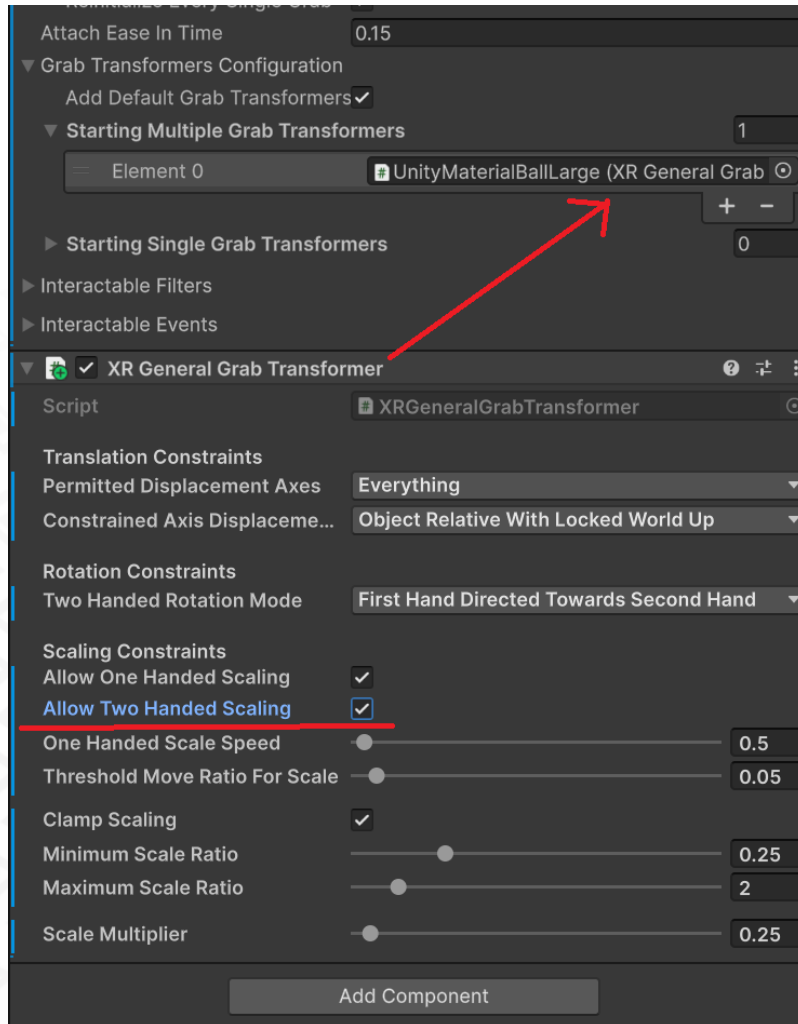
- b) Dodać komponent XR Grab Interactable. Ustawić Select Mode = Multiple i zaznaczyć opcję Use Dynamic Attach (patrz rys. 20). Dzięki temu obiekt będzie można chwytać więcej niż jedną ręką i bardziej naturalnie ustawiać punkt chwytu.



Rys. 20: Konfiguracja komponentu XR Grab Interactable.

TEST APLIKACJI: Uruchomić Play i sprawdzić, czy wybrany obiekt 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. 21).



Rys. 21: 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.

6. Import pakietu assetów z poprzedniego projektu

Model mieszkania powinien odpowiadać założeniom z poprzedniej instrukcji: minimum trzy pokoje, korytarz, otwory na okna, minimum trzy drzwi, okna, fotel i stół. W tej instrukcji nie przygotowuje się już tych modeli w Blenderze. Student korzysta z assetów, które zostały wcześniej zaimportowane do Unity i wyeksportowane do pakietu `VR6_PreviousAssets.unitypackage`. Układ mieszkania można oprzeć na rys. 22.

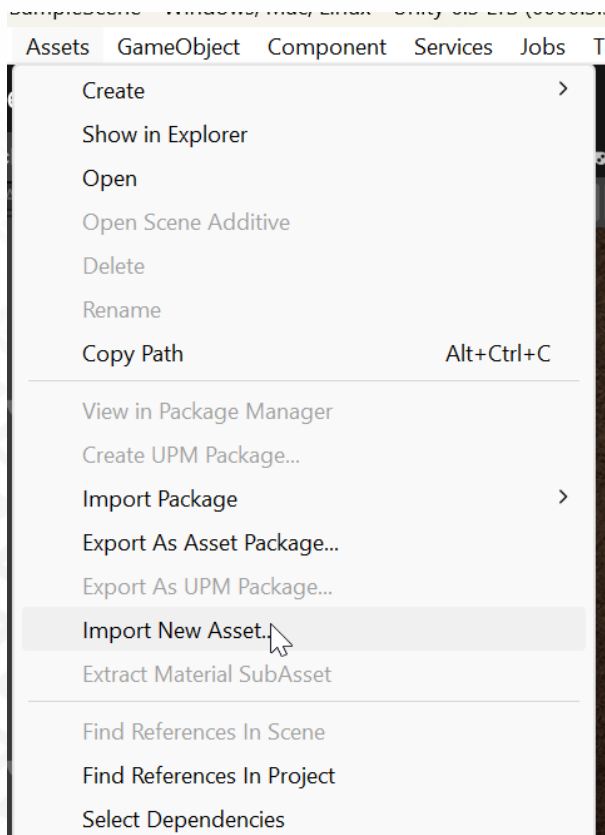


Rys. 22: Przykładowy układ mieszkania i wyposażenia wykorzystany jako punkt odniesienia

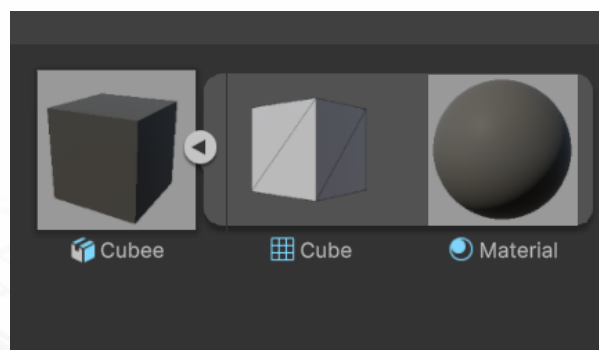
6.1 Import pakietu Unity Package

Instrukcja:

1. W nowym projekcie VR6_ApartmentVR wybierz **Assets -> Import Package -> Custom Package....**
2. Wskaż plik `VR6_PreviousAssets.unitypackage` przygotowany przed rozpoczęciem nowej aplikacji.
3. W oknie importu sprawdź, czy na liście znajdują się modele mieszkania, prefaby, materiały, tekstury, dźwięki i dodatkowe obiekty.
4. Pozostaw zaznaczoną całą wymaganą zawartość pakietu i kliknij **Import**.
5. Po imporcie sprawdź foldery w oknie **Project**, w szczególności **Models**, **Materials**, **Textures**, **Prefabs**, **DownloadedAssets** oraz **Audio**.
6. Jeżeli brakuje materiałów lub tekstur, nie twórz ich od nowa. Wróć do poprzedniego projektu Unity i ponownie wyeksportuj pakiet z opcją **Include dependencies**.



(a) Import zasobu lub pakietu do Unity



(b) Modele i tekstury w folderze projektu

Rys. 23: Kontrola zaimportowanych assetów w nowym projekcie Unity

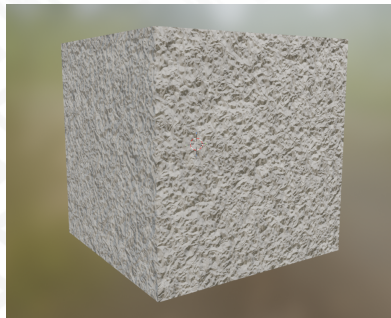
6.2 Umieszczenie mieszkania w scenie

Instrukcja:

1. W folderze **Prefabs** albo **Models** odszukaj asset mieszkania z poprzedniego laboratorium.
2. Przeciągnij mieszkanie do sceny **VR6_Apartment_Final**.
3. Utwórz pusty obiekt nadrzędny **Apartment_Root** i przenieś do niego wszystkie elementy mieszkania, jeżeli nie są jeszcze uporządkowane.
4. Sprawdź skalę mieszkania względem **XR Origin**. Użytkownik powinien swobodnie przechodzić korytarzem i wejść do minimum trzech pokoi.
5. Sprawdź, czy drzwi, okna, fotel, stół i dodatkowe obiekty znajdują się w poprawnych miejscach.
6. Ustaw jedno drzwi jako obiekt przygotowany do animacji otwierania i zamykania. Pozostałe elementy konstrukcyjne pozostają nieruchome.
7. Zapisz scenę.

7. Materiały i tekstury z wcześniejszych laboratoriów

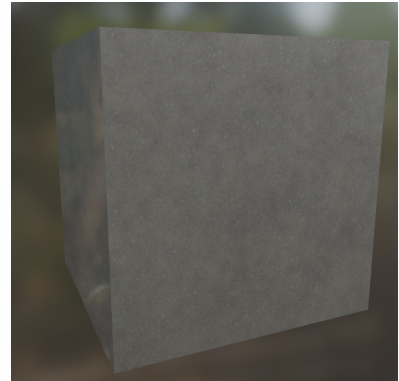
Po imporcie pakietu należy wykorzystać materiały, które zostały już przygotowane i zaimportowane do Unity we wcześniejszych laboratoriach. W tym ćwiczeniu nie wypala się nowych tekstur i nie tworzy materiałów od początku. Wykorzystać można między innymi materiały ściany, drewna, szkła, metalu, kamienia, paneli i tkaniny.



(a) Materiał ściany



(b) Materiał drewna



(c) Materiał kamienia

Rys. 24: Przykładowe materiały przygotowane we wcześniejszych laboratoriach

Instrukcja:

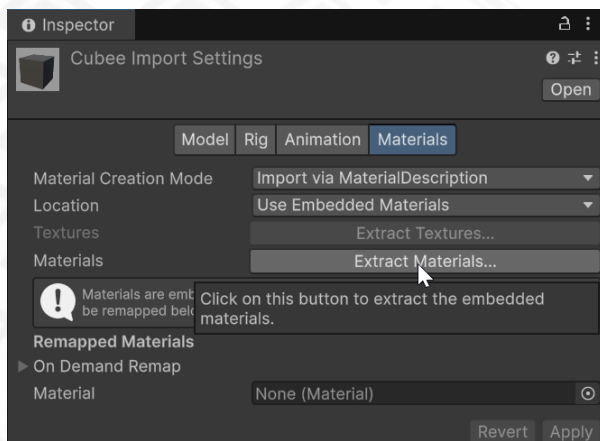
1. W folderze `Assets/Materials` odzyskaj materiały wyeksportowane z poprzedniego projektu, np. `Wall_Paint`, `Wood`, `Glass`, `Metal`, `Fabric`, `Stone`.
2. Jeżeli po imporcie pakietu materiał nie ma przypisanych tekstur, wróć do poprzedniego projektu i wyeksportuj pakiet ponownie z opcją `Include dependencies`; nie przygotowywaj nowych tekstur w Blenderze.
3. Do ścian przypisz materiał tynku lub farby.
4. Do drzwi, stołu i elementów okien przypisz materiał drewna albo metalu.
5. Do szyb przypisz materiał szkła z przezroczystością.
6. Do fotela przypisz materiał tkaniny lub skóry.
7. Sprawdź wygląd materiałów w `Scene view` i `Game view`.

8. Obiekty nieruchome i ruchome

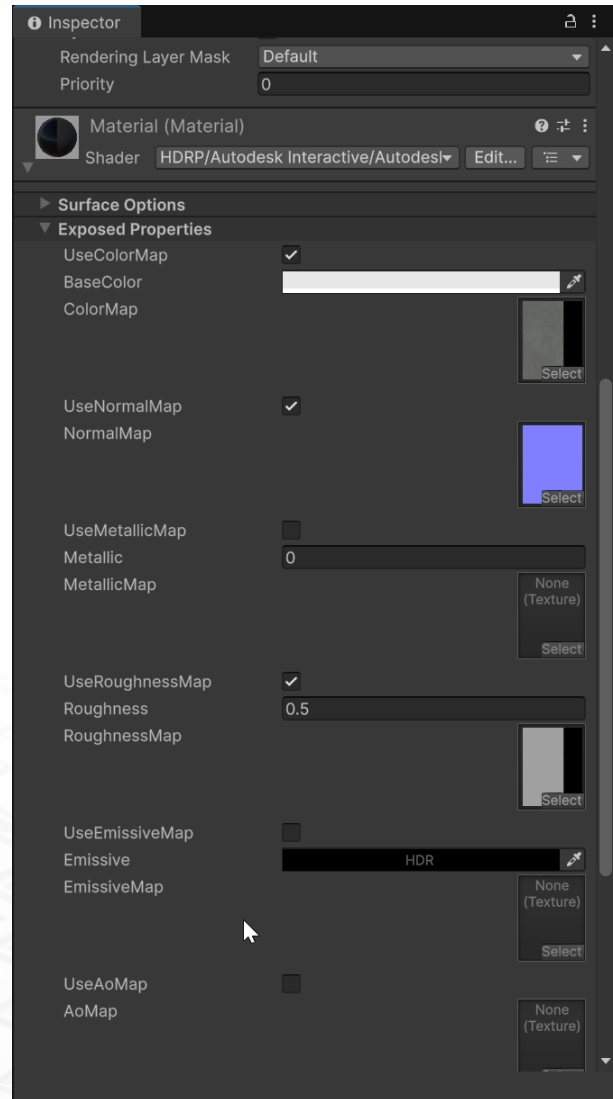
W scenie należy jasno rozdzielić elementy statyczne i elementy możliwe do interakcji.

Obiekty nieruchome:

- ściany, podłoga, sufit i korytarz,



(a) Wyodrębnianie materiałów modelu



(b) Przypisywanie tekstur do materiału

Rys. 25: Praca z materiałami po imporcie modeli do Unity

- okna,
- ościeżnice,
- drzwi jako obiekty konstrukcyjne, z wyjątkiem jednego skrzydła animowanego przez skrypt,
- stałe elementy tła i oświetlenia.

Obiekty mogące być ruchome lub interaktywne:

- stół,
- fotel,
- wybrane obiekty pobrane z Internetu, np. lampa, obraz, dekoracja lub mała szafka,

- jedne drzwi, ale wyłącznie przez animację otwierania i zamykania.

Instrukcja ustawienia:

1. Dla ścian, okien i elementów stałych zaznacz opcję **Static**, jeżeli nie będą zmieniane podczas działania aplikacji.
2. Nie dodawaj komponentu **Rigidbody** do elementów konstrukcyjnych mieszkania.
3. Dla stołu, fotela i wybranych pobranych obiektów można dodać **Rigidbody** oraz komponent interakcji VR używany w projekcie.
4. Jeżeli obiekt ma być tylko dekoracją, pozostaw go nieruchomy.
5. Ustaw drzwi w pozycji początkowej zgodnej z działaniem skryptu, zwykle zamkniętej; otwarcie ma nastąpić po wskazaniu drzwi i naciśnięciu triggera.

9. Przygotowanie interakcji drzwi skryptem Raycast

Jedne drzwi mają reagować na wskazanie kontrolerem VR i naciśnięcie triggera. W tej wersji ćwiczenia **nie tworzy się osobnych klipów animacji ani kontrolera Animator**. Animacja jest wbudowana w skrypt prowadzącego i polega na płynnym obrocie obiektu **DoorPivot**, który pełni rolę zawiasu.

9.1 Zasada działania skryptu

Skrypt **Raycast** służy do obsługi interakcji z drzwiami w aplikacji VR. Wysyła niewidzialny promień z kontrolera VR i sprawdza, czy użytkownik wskazuje nim drzwi. Jeżeli promień trafi w collider drzwi, kontroler wykonuje krótką wibrację. Dzięki temu użytkownik otrzymuje informację, że obiekt może zostać użyty.

Po naciśnięciu triggera skrypt uruchamia otwieranie albo zamykanie drzwi. Jeżeli drzwi są zamknięte, zostają otwarte. Jeżeli są otwarte, zostają zamknięte. Ruch wykonywany jest przez płynny obrót obiektu **DoorPivot** pomiędzy pozycją zamkniętą i otwartą.

9.2 Przygotowanie drzwi w scenie

Aby drzwi obracały się poprawnie, należy przygotować odpowiednią strukturę obiektów. Najpierw utwórz pusty obiekt **DoorPivot**, który będzie zawiasem drzwi, i ustaw go dokładnie przy krawędzi drzwi, wokół której mają się obracać.

DoorPivot DoorModel — dziecko obiektu DoorPivot

1. W Hierarchy utwórz pusty obiekt **DoorPivot**.

2. Ustaw DoorPivot dokładnie w miejscu zawiasu, czyli przy krawędzi drzwi.
3. Przeciągnij model drzwi jako dziecko obiektu DoorPivot; model może nazywać się np. DoorModel.
4. Upewnij się, że ręczny obrót DoorPivot wokół osi pionowej obraca całe drzwi w poprawnym kierunku.
5. Nie dodawaj komponentu Animator i nie twórz klipów Door_Open ani Door_Close; obrót wykonuje skrypt.

9.3 Collider na obiekcie drzwi

Na model drzwi, czyli DoorModel, należy dodać collider. Collider jest wymagany, ponieważ Raycast wykrywa tylko obiekty posiadające collider. Bez niego skrypt nie będzie wiedział, że użytkownik wskazuje drzwi.

1. Zaznacz model drzwi w Hierarchy.
2. W Inspectorze kliknij Add Component.
3. Wyszukaj Box Collider.
4. Dodaj komponent i dopasuj rozmiar collidera do drzwi.
5. Jeżeli model drzwi składa się z kilku elementów, collider można dodać do głównego obiektu drzwi albo do osobnego pustego obiektu znajdującego się w strukturze drzwi.

9.4 Podpięcie skryptu do kontrolera VR

Skrypt Raycast należy dodać do obiektu kontrolera VR, najczęściej do prawego kontrolera użytkownika.

```
XR Origin
  Camera Offset
    Right Controller
```

1. Skopiuj skrypt dostarczony do tej instrukcji, pobierz go z załącznika PDF i zapisz w projekcie Unity.
2. W Hierarchy odszukaj prawy kontroler, np. XR Origin -> Camera Offset -> Right Controller.
3. Zaznacz obiekt prawego kontrolera.
4. W Inspectorze kliknij Add Component, wyszukaj skrypt Raycast i dodaj go do kontrolera.

9.5 Ustawienia skryptu w Inspectorze

Po dodaniu skryptu należy ustawić jego pola:

- **Controller Node** — określa, z którego kontrolera pobierany jest trigger i wibracja; dla prawej ręki ustaw **Right Hand**,
- **Ray Origin** — punkt, z którego wysyłany jest promień; najczęściej transform prawego kontrolera,
- **Ray Distance** — maksymalna odległość, z której można wskazać drzwi, np. 5,
- **Door Layer** — warstwy obiektów wykrywanych przez **Raycast**; na początek można zostawić **Everything**,
- **Door Pivot** — najważniejsze pole; przeciągnij tutaj pusty obiekt **DoorPivot**, czyli zawias drzwi,
- **Open Rotation** — kąt, o jaki mają obrócić się drzwi, np. 0, 90, 0; jeżeli drzwi otwierają się w złą stronę, ustaw np. 0, -90, 0,
- **Animation Time** — czas otwierania lub zamykania drzwi, np. 1,
- **Rotation Curve** — krzywa animacji; domyślna wartość daje płynne rozpoczęcie i zakończenie ruchu,
- **Haptic Amplitude** — siła wibracji kontrolera po wskazaniu drzwi, np. 0.4; jeżeli wibracja jest za słaba, można ustawić np. 0.8,
- **Haptic Duration** — czas trwania pojedynczej wibracji, np. 0.08.

9.6 Zasada działania krok po kroku

1. Użytkownik kieruje kontroler w stronę drzwi.
2. Skrypt wysyła promień z obiektu **Ray Origin**.
3. Jeżeli promień trafi w **collider** drzwi, kontroler wykonuje krótką wibrację.
4. Użytkownik naciska **trigger**.
5. Skrypt sprawdza, czy drzwi są aktualnie otwarte czy zamknięte.
6. Obiekt **DoorPivot** obraca się płynnie do pozycji otwartej albo zamkniętej.

9.7 Test i najczęstsze problemy

1. Uruchom **Play mode**.
2. Skieruj prawy kontroler VR na model drzwi.
3. Sprawdź, czy po trafieniu promieniem w drzwi kontroler wykonuje krótką wibrację.

4. Naciśnij trigger i sprawdź, czy DoorPivot płynnie obraca drzwi do pozycji otwartej.
5. Naciśnij trigger ponownie i sprawdź, czy drzwi wracają do pozycji zamkniętej.
6. Jeżeli drzwi się nie otwierają, sprawdź: `Box Collider`, przypisanie `Door Pivot`, obiekt `Ray Origin`, kierunek osi `forward` kontrolera oraz ustawienie `Door Layer`.
7. Jeżeli drzwi obracają się wokół środka zamiast zawiasu, przesuń `DoorPivot` na krawędź drzwi, tam gdzie powinien znajdować się zawias.

Wymaganie: drzwi mają być obsługiwane przez skrypt `Raycast`, wibrację kontrolera po wskazaniu drzwi oraz kliknięcie triggerem. Nie należy przygotowywać osobnych klipów animacji w oknie `Animation`; animacja jest realizowana przez skrypt jako obrót `DoorPivot`.

10. Oświetlenie i dźwięk

Scena VR nie powinna być pusta ani ciemna. Należy przygotować światło i dźwięk, aby mieszkanie było czytelne i przyjemne do oglądania.

10.1 Oświetlenie

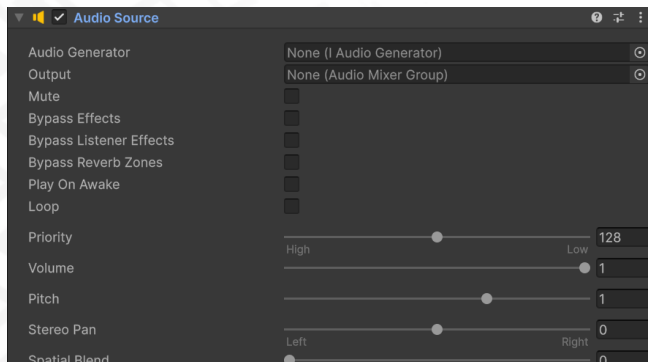
Oświetlenie powinno zapewniać dobrą widoczność wnętrza mieszkania w goglach VR. Nie powinno prześwieślać sceny ani pozostawiać ciemnych miejsc utrudniających poruszanie się po pomieszczeniach.

1. Dodaj światło kierunkowe `Directional Light`, które symuluje ogólne światło zewnętrzne.
2. Dodaj światła punktowe lub obszarowe w pomieszczeniach, np. przy lampach, korytarzu i ważnych elementach sceny.
3. Dla HDR/HDRP sprawdź ekspozycję w `Global Volume`, aby wewnątrz nie było prześwietlone ani zbyt ciemne.
4. Ustaw intensywność światła tak, aby mieszkanie pozostało najważniejszym i dobrze widocznym elementem sceny.
5. W oknie `Window` -> `Rendering` -> `Lighting` przelicz lub odśwież oświetlenie, jeżeli projekt tego wymaga.
6. Sprawdź widoczność sceny w goglach VR, nie tylko w oknie `Scene view`.

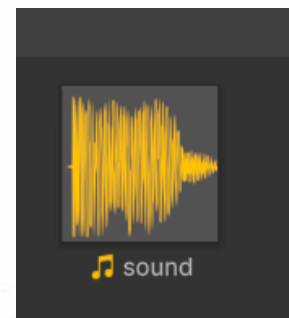
10.2 Dźwięki

W tej części należy przygotować dźwięk tła sceny. Informacja zwrotna dla drzwi jest realizowana przez wibrację kontrolera w skrypcie `Raycast`, dlatego osobny dźwięk drzwi nie jest wymagany, chyba że prowadzący dostarczy dodatkową wersję skryptu z obsługą audio.

1. Utwórz folder `Assets/Audio`.
2. Zaimportuj dźwięk tła, np. cichy szum mieszkania, klimatyzacji albo odgłos otoczenia. Plik może mieć format `.wav`, `.ogg` albo `.mp3`.
3. Utwórz pusty obiekt `Ambient_Audio` w scenie.
4. Dodaj do niego komponent `Audio Source`.
5. Przypisz zaimportowany plik do pola `AudioClip`.
6. Włącz `Loop`, ustaw niewielką głośność i sprawdź, czy dźwięk nie przeszkadza użytkownikowi.



(a) Komponent `AudioSource` i podstawowe ustawienia



(b) Zaimportowany plik jako `AudioClip`

Rys. 26: Konfiguracja dźwięku tła w Unity na podstawie instrukcji VR2

Kontrola: dźwięk tła powinien działać w pętli z małą głośnością. Wibrację drzwi należy sprawdzić w ustawieniach skryptu, w polach `Haptic Amplitude` oraz `Haptic Duration`.

Wszystkie dźwięki pobrane z Internetu powinny pochodzić ze źródeł pozwalających na użycie ich w projekcie edukacyjnym. Należy zapisać źródła pobranych plików.

11. Dodatkowe obiekty z Internetu

Scenę należy uzupełnić o kilka gotowych obiektów, tak aby mieszkanie wyglądało jak kompletna przestrzeń VR. Mogą to być:

- lampy,
- obrazy,
- szafki,
- półki,
- rośliny,

- drobne dekoracje.

Instrukcja:

1. Pobierz obiekty w formacie obsługiwanym bezpośrednio przez Unity, np. jako pakiet Unity, prefab lub gotowy asset niewymagający edycji w Blenderze.
2. Umieść je w folderze `Assets/DownloadedAssets`.
3. Sprawdź skalę i liczbę wieloboków; obiekty nie mogą nadmiernie obciążać aplikacji VR.
4. Przypisz materiały zgodne z resztą mieszkania.
5. Ustaw obiekty w logicznych miejscach: lampy przy ścianach, obrazy na ścianach, szafki w pokojach, dekoracje na stole.
6. Część z nich można ustawić jako ruchome lub interaktywne, jeżeli nie zaburza to działania sceny.
7. Zapisz listę źródeł lub linków do pobranych zasobów.

12. Test aplikacji VR

Przed zbudowaniem pliku `.exe` należy wykonać pełny test w edytorze Unity.

1. Uruchom `Play mode`.
2. Sprawdź, czy użytkownik startuje w odpowiednim miejscu mieszkania.
3. Przejdź przez korytarz i minimum trzy pokoje.
4. Sprawdź, czy ściany, podłoga, okna i większość drzwi są nieruchome.
5. Sprawdź, czy stół, fotel lub wybrane pobrane obiekty zachowują się zgodnie z założeniem interakcji.
6. Skieruj kontroler na interaktywne drzwi, sprawdź wibrację po trafieniu promieniem i kliknij triggerem, aby sprawdzić otwieranie oraz zamykanie przez obrót `DoorPivot`.
7. Sprawdź, czy dźwięk tła działa i nie jest zbyt głośny.
8. Sprawdź, czy scena jest czytelna w HDR/HDRP i w goglach VR.
9. Sprawdź `Console`; projekt nie powinien generować błędów.



Rys. 27: Przykład obiektu z poprawnie przypisanymi teksturami w Unity

13. Budowanie aplikacji do pliku EXE

Po zakończeniu testów należy zbudować aplikację jako plik wykonywalny.

Instrukcja:

1. Zapisz scenę `VR6_Apartment_Final`.
2. Otwórz `File` -> `Build Profiles`.
3. Sprawdź, czy na liście scen znajduje się jedna scena projektu. Aktualnie otwarta scena jest zwykle dodana automatycznie.
4. Upewnij się, że jako aktywny profil budowania wybrany jest `Windows`. Jeżeli nie, wybierz profil `Windows` i ustaw go jako aktywny.
5. Dla systemu Windows sprawdź architekturę `x86_64`.
6. Otwórz `Player Settings` i sprawdź nazwę aplikacji, ikonę, ustawienia jakości, HDR/HDRP oraz XR.
7. Kliknij `Build`.
8. Po kliknięciu `Build` utwórz folder `Build_VR6` i wskaż go jako miejsce zapisania aplikacji.

9. Po zakończeniu budowania uruchom plik `.exe` i sprawdź działanie aplikacji poza edytorem Unity.

Do oddania należy przygotować:

- folder nowego projektu Unity,
- pakiet assetów `VR6_PreviousAssets` w formacie `.unitypackage`,
- folder z buildem `.exe`,
- krótki opis źródeł pobranych obiektów oraz dźwięków.

14. Zadania do wykonania

W ramach laboratorium student wykonuje następujące zadania:

1. Otworzyć poprzedni projekt Unity, w którym znajdują się mieszkanie, materiały, tekstury i testowana aplikacja VR.
2. Przed utworzeniem nowej aplikacji wyeksportować potrzebne zasoby do pakietu assetów Unity.
3. Nazwać pakiet `VR6_PreviousAssets` i zapisać go w formacie `.unitypackage` z opcją `Include dependencies`.
4. Utworzyć nowy projekt `VR6_ApartmentVR` i zaimportować do niego przygotowany pakiet assetów.
5. Skonfigurować nowy projekt do pracy w HDR/HDRP oraz sprawdzić działanie VR.
6. Umieścić w scenie mieszkanie i modele z zaimportowanego pakietu Unity.
7. Przypisać gotowe materiały i tekstury zaimportowane z poprzedniego projektu Unity.
8. Dodać oświetlenie i dźwięki.
9. Dodać obiekty pobrane z Internetu, np. lampy, obrazy, szafki, półki lub dekoracje.
10. Ustawić obiekty nieruchome i ruchome zgodnie z instrukcją.
11. Przygotować strukturę `DoorPivot` oraz dodać `Box Collider` do modelu drzwi.
12. Wpiąć skrypt prowadzącego oparty o `Raycast`, trigger kontrolera, wibrację i obrót `DoorPivot`.
13. Przetestować aplikację w goglach VR.
14. Zbudować gotową aplikację do pliku `.exe`.

OCENA

Student dostaje:

- ocenę **3** za eksport pakietu assetów z poprzedniego projektu Unity, uruchomienie nowego projektu VR, import mieszkania, przypisanie podstawowych materiałów i poprawną scenę testową,
- ocenę **4** za kompletną scenę mieszkania utworzoną w nowym projekcie na podstawie pakietu assetów Unity, z HDR/HDRP, światłem, dźwiękiem i obiektami z poprzednich laboratoriów,
- ocenę **5** za kompletną aplikację VR utworzoną bez ponownego używania Blendera: z poprawnie wyeksportowanym i zaimportowanym pakietem `.unitypackage`, materiałami, dodatkowymi obiektami z Internetu, poprawnie ustawionymi obiektami nieruchomymi i ruchomymi, drzwiami obsługiwanymi skryptem `Raycast`, z wibracją kontrolera po wskazaniu drzwi i płynnym obrotem `DoorPivot` po kliknięciu triggerem oraz zbudowanym plikiem `.exe`.

Uwagi końcowe

Podczas pracy należy regularnie zapisywać projekt i scenę. Przed oddaniem ćwiczenia trzeba sprawdzić działanie w goglach VR oraz uruchomić przygotowany plik `.exe`. Projekt nie powinien zawierać błędów w `Console`. Wszystkie pobrane obiekty i dźwięki powinny mieć zapisane źródła, a ich licencja musi pozwalać na wykorzystanie w projekcie edukacyjnym.