LABORATORIUM 2
Interakcja i lokomocja w wirtualnej rzeczywistości

XR Interaction Toolkit • chwytanie • ray interaction • UI • teleportacja • haptyka

Problem inżynierski

Jak zaprojektować obsługę stanowiska VR tak, aby użytkownik mógł niezawodnie wskazywać, chwytać i odkładać elementy oraz przemieszczać się bez przypadkowych aktywacji i nadmiernego dyskomfortu?

Parametry laboratorium

Parametr	Wartość
Tryb pracy	pary; symulator na każdym stanowisku, Quest 3 rotacyjnie
Rezultat	działający scenariusz operatorski VR i trzy próby pomiarowe
Oddanie	commit + evidence/lab02.md; bez tradycyjnego sprawozdania
Punkt kontrolny	0–10 pkt; demonstracja do 90 s
Wersja	1.2 • 29 sierpnia 2026 r.

System oceniania: Checkpointy i oceny

Ocena wynika z osiągnięcia kolejnych checkpointów. Każdy checkpoint weryfikuje prowadzący w ciągu maksymalnie 90 sekund (z demonstracją).

Ocena	Główne wymagania
3.0	Chwyt + promień + lokomocja + pomiar (10 pkt)
3.5	Panel UI + sprzężenie zwrotne (wizualne, dźwięk, haptyka)
4.0	Pełny scenariusz + trzy próby pomiarowe + diagnoza
4.5	Rotacja Quest 3 + analiza różnic (pose, wejście, haptyka)
5.0	Dostępność, ergonomia, porównanie Movement Type, optymalizacja

Walidacja: Numer indeksu i wariant diagnostyczny

Aby uniknąć kopiowania wyników między parami, każdy student wpisuje swój numer indeksu do skryptu. Wariant błędu w Części F wynika ze wzoru:

$$\text{wariantDiagnostyczny} = (\text{numer_indeksu_1} + \text{numer_indeksu_2}) \bmod 6$$

Wynik: wariant A, B, C, D, E lub F (każda para ma inny błąd do naprawienia)

Evidence/lab02.md musi zawierać numery indeksów obu studentów. Prowadzący sprawdza, czy przydzielony wariant zgadza się z wyliczonym ze wzoru.

Wymagania wstępne

Przed zajęciami

Zaliczone Laboratorium 1 albo działająca scena bazowa z XR Origin i symulatorem

Repozytorium kursu w wersji wskazanej przez prowadzącego

Brak czerwonych błędów w Console

Umiejętność pracy z komponentami, zdarzeniami Unity i Rigidbody

Znajomość różnicy między układem lokalnym i globalnym

Konfiguracja projektu

Element	Wymaganie	Sposób sprawdzenia
Unity i pakiety	bez aktualizacji względem laboratorium 1	ProjectVersion.txt i packages-lock.json
XRI Starter Assets	zaimportowane akcje i prefab konfiguracji	Input Action Manager ma przypisane akcje

Symulator XRI	wariant zgodny z zamrożoną wersją XRI	Ruch głowy i kontrolerów w Play Mode
Scena Lab02_Interaction.unity	jedna aktywna kamera główna	Hierarchy po otworzeniu sceny

ŹRÓDŁO PRAWDY: Nazwy prefabów i komponentów różnią się między wersjami XRI. Stosuj wersję zapisaną w packages-lock.json oraz dokumentację odpowiadającą tej wersji. Nie aktualizuj pakietów podczas ćwiczenia.

Bezpieczeństwo i higiena pracy

- Przed założeniem gogli usuń przeszkody i sprawdź granicę bezpieczeństwa. Teleportacja w aplikacji nie zastępuje kontroli przestrzeni fizycznej.
- Podczas rotacji jedna osoba używa gogli, a druga obserwuje otoczenie i przewód. Nie wykonuj kroków poza wyznaczony obszar.
- Przerwij próbę po wystąpieniu nudności, zawrotów głowy, dezorientacji, bólu oczu albo utraty równowagi.
- Nie stosuj ciągłego ruchu ani płynnego obrotu w części obowiązkowej. Ich porównanie jest dozwolone wyłącznie jako krótka demonstracja prowadzącego.
- Nie zwiększaj amplitudy i czasu impulsu haptycznego bez potrzeby. Informacja zwrotna ma sygnalizować zdarzenie, a nie dominować nad zadaniem.
- Nie zapisuj obrazu passthrough ani danych użytkownika. Wyniki pomiarowe oznacz numerem pary, nie nazwą osoby badanej.

Podstawy teoretyczne

4.1. Model interakcji XRI

Interactor reprezentuje źródło działania użytkownika, np. dłoń lub promień kontrolera. Interactable reprezentuje obiekt, który może być wskazany, wybrany albo aktywowany. XR Interaction Manager pośredniczy między nimi i rozstrzyga zgodność warstw oraz stanów Hover, Select i Activate [1].

Stan	Znaczenie operacyjne	Przykładowa informacja zwrotna
Hover	Interaktor może wejść w interakcję z celem	Podświetlenie obrysu lub zmiana koloru
Select	Interaktor utrzymuje wybór obiektu	Chwyt, zatrzaśnięcie albo wyraźny stan przycisku
Activate	Wybrany obiekt otrzymuje dodatkowe polecenie	Uruchomienie narzędzia lub funkcji

4.2. Chwyt bezpośredni i wskazywanie promieniem

XR Direct Interactor odwzorowuje interakcję bliską: obszar kolizji interaktora musi zetknąć się z obiektem. XR Ray Interactor umożliwia wybór z odległości. Rozdzielenie masek interakcji ogranicza przypadkowe wskazywanie podłoża, elementów technicznych i paneli UI. XR Grab Interactable wyznacza sposób ruchu chwytanego obiektu, a punkt Attach Transform kontroluje ułożenie elementu w dłoni.

REGUŁA PROJEKTOWA: Do małych elementów montażowych użyj chwytu bezpośredniego. Promień przeznacz do panelu i odległych celów. Ta sama akcja Select może sterować oboma interaktorami, ale aktywna modalność i maski muszą zapobiegać konfliktom.

4.3. Lokomocja

W XRI dostawcy lokomocji składają żądania zmiany XR Origin za pośrednictwem mechanizmu lokomocji. Teleportation Area dopuszcza wiele punktów docelowych na powierzchni, natomiast Teleportation Anchor określa przygotowaną pozycję i opcjonalnie orientację. W części obowiązkowej stosuje się teleportację oraz obrót skokowy o 30°, ponieważ dają jednoznaczne stany do testowania i ograniczają czas ekspozycji na ruch niezgodny z sygnałem przedsionkowym.

4.4. Informacja zwrotna i pomiar

Stan interakcji powinien być komunikowany co najmniej w dwóch kanałach, gdy błąd ma znaczenie dla zadania. Symulator pozwala sprawdzić logikę i sygnały wizualne, ale nie potwierdza rzeczywistego impulsu haptycznego. Skuteczność scenariusza oceniaj za pomocą czasu wykonania oraz liczby błędów operacyjnych, np. upuszczeń, błędnych aktywacji i nieudanych prób teleportacji.

Plan pracy

Etap	Dowód ukończenia
Miniquiz i kontrola sceny	Odpowiedzi + scena bez błędów krytycznych
Interakcja bliska i odległa	Chwyt bezpośredni + wybór promieniem
Gniazdo i ograniczenia interakcji	Poprawne osadzenie modułu, brak chwytania tła
Panel przestrzenny i sprzężenie zwrotne	Start, Reset, Potwierdź + sygnały stanu
Teleportacja i obrót skokowy	Obszar, kotwica i snap turn
Scenariusz operatorski	Trzy próby, mediana i błędy
Rotacja Quest 3 lub test zastępczy	Weryfikacja wejścia i haptyki
Demonstracja i exit ticket	Evidence/lab02.md + odpowiedź indywidualna

Checkpoint 3.0

Wymagania do zaliczenia (Część A–E):

✓ Scena uruchamia się w Play Mode bez błędów krytycznych

✓ Jedna aktywna kamera główna, jeden właściwy XR Interaction Manager

✓ Module_A można chwycić bezpośrednio i poprawnie ułożyć w dłoni (Attach Transform)

✓ Ray Interactor obsługuje panel bez wybierania środowiska

✓ Socket_A akceptuje wyłącznie właściwy moduł i jednoznacznie potwierdza osadzenie

✓ Teleportacja działa na dozwolonym obszarze, obrót skokowy = 30°

✓ Przeprowadzono trzy próby pomiarowe; zapisano medianę czasu i błędy

✓ Diagnoza: objaw, dwie hipotezy, test rozstrzygający, poprawka

✓ Evidence/lab02.md zawiera: numery indeksów, wersje, wyniki pomiarów, wariant diagnostyczny

Część A. Uruchomienie i kontrola wejścia

A.1. Miniquiz (10 minut, indywidualnie)

Pięć pytań ocenianych automatycznie. Po oddaniu testu porównaj uzasadnienie odpowiedzi z konfiguracją własnej sceny.

1. Wskaż element, który rozstrzyga zgodność interaktora i obiektu interaktywnego. (Odp: XR Interaction Manager + maski)
2. Dobierz metodę interakcji do małego modułu 20 cm od dłoni. (Odp: chwyt bezpośredni)
3. Wyjaśnij różnicę między Teleportation Area i Teleportation Anchor. (Odp: swobodny punkt vs przygotowana pozycja)
4. Wskaż, czego nie potwierdza poprawne działanie symulatora. (Odp: haptyka lub zachowanie konkretnego runtime)
5. Rozpoznaj błąd: promień zaznacza podłogę zamiast przycisku. (Odp: sprawdzić maski i kolejność trafień)

A.2. Gałąź i scena robocza (5 minut)

1. Przejdź na stan końcowy Lab 1 i pobierz aktualne znaczniki
2. Utwórz gałąź: git switch -c team-<nr>/lab02-<nazwisko1>-<nazwisko2>
3. Skopiuj scenę: cp Assets/Scenes/Lab01_Baseline.unity Assets/Scenes/Lab02_Interaction.unity
4. Play Mode: poruszeń głową i kontrolerami - powinno działać

PUNKT KONTROLNY A: Scena uruchamia się, jest jedna kamera główna, jeden XR Interaction Manager. Symulator pozwala sterować głową i dwoma kontrolerami.

Część B. Interakcja bliska, promień i gniazdo

B.1. Przygotowanie stanowiska (15 minut)

1. Utwórz obiekt Workbench (1.4m × 0.7m × 0.9m blatu) z Collider

2. Dodaj Module_A (0.12 × 0.06 × 0.18 m) z Collider i Rigidbody - spoczywa na blacie
3. Dodaj Socket_A z Collider (trigger) i czytelny obrys miejsca docelowego
4. Utwórz warstwy interakcji: DirectGrab, RayUI, Teleport, Environment
5. Przypisz obiekty do właściwych warstw

B.2. Chwyt bezpośredni (10 minut)

1. Dodaj XR Grab Interactable do Module_A
2. Utwórz potomny Attach_Module w miejscu naturalnego chwytu
3. Dobierz Movement Type (Velocity Tracking vs Kinematic) - moduł stabilny, poprawne kolizje
4. Dodaj wizualną zmianę stanu Hover i Select (kolor/materiał)

B.3. Promień i filtracja celów (10 minut)

1. Włącz XR Ray Interactor dla dominującej dłoni
2. Ogranicz promień do panelu UI i przygotowanych celów (Raycast Mask)
3. Promień nie zaznacza blatu, ścian ani dekoracji
4. Ustal regułę współistnienia Direct + Ray Interactor (brak konfliktu przy zbliżeniu dłoni)

B.4. Gniazdo (10 minut)

1. Dodaj XR Socket Interactor do Socket_A i ogranicz maskę do Module_A
2. Ustaw punkt zaczepienia (Attach Transform) - moduł przyjmuje jednoznaczną pozycję
3. Dodaj sygnał poprawnego osadzenia (zmiana koloru obrys + zdarzenie Select Entered)

PUNKT KONTROLNY B: Module_A można chwycić tylko bezpośrednio, poprawnie układa się, zatrzaskuje się wyłącznie w Socket_A. Promień nie wybiera środowiska.

Część C. Panel przestrzenny i informacja zwrotna

C.1. Panel operatorski (10 minut)

1. Utwórz Canvas w trybie World Space, pionowo obok stanowiska
2. Dodaj przyciski: Start, Reset, Potwierdź alarm (czytelne etykiety)
3. Połącz Start z pomiarem, Reset z odtworzeniem stanu, Potwierdź z zakończeniem
4. Dodaj tekst Status: gotowy / montaż / alarm / ukończono

C.2. Sprzężenie zwrotne (10 minut)

1. Wizualne sygnały Hover i Select dla przycisków i modułu (zmiana koloru)
2. Dźwiękowe: krótki dźwięk dla poprawnego osadzenia, inny dla błędu
3. Haptyka: Simple Haptic Feedback dla Select i osadzenia (log w Edytorze, rzeczywisty impuls na Quest 3)

Część D. Teleportacja i obrót skokowy

D.1. Konfiguracja lokomocji (10 minut)

1. Dodaj lub aktywuj komponenty lokomocji z prefabu XRI
2. Utwórz Teleportation Area na podłożu (promień wskazuje dozwoloną powierzchnię)
3. Dodaj Teleportation Anchor przed stanowiskiem (Match Orientation = true)
4. Włącz obrót skokowy = 30° (sprawdź mapowanie osi)
5. Wyłącz ciągłe przemieszczanie i Continuous Turn Provider

PUNKT KONTROLNY C: Teleportacja na obszar/kotwicę, obrót 30°, próba wskazania blatu jest jednoznacznie odrzucana.

Część E. Scenariusz operatorski i pomiar

E.1. Sekwencja zadania (30 minut)

1. Reset: użytkownik na StartPose, moduł na blacie, gniazdo puste, timer = 0
2. Teleportacja: do kotwicy przed stanowiskiem
3. Start: przycisk na panelu (promień) - timer rozpoczyna
4. Chwyt: Module_A bezpośrednio
5. Osadzenie: w Socket_A - status zmienia się na „alarm"
6. Potwierdzenie: przycisk Potwierdź alarm - timer zatrzymuje się

E.2. Rejestracja wyników (10 minut)

Wykonaj jedną próbę treningową, a następnie trzy próby pomiarowe.

Próba	Czas [s]	Upuszczenia	Błędne aktywacje	Nieudane teleportacje
1	...	...	...	...
2	...	...	...	...
3	...	...	...	...
Mediana / suma	...	-	...	...

E.3. Rejestrator próby (skrypt)

```
using UnityEngine;
public class Lab02Trial : MonoBehaviour{
   int pairIndexSum = 0; // WPISZ: suma numerów indeksów
   private float startedAt;
   private int errors;
   private bool running;
   public void StartTrial() { startedAt = Time.unscaledTime; errors = 0; running = true; }
   public void RegisterError() {if (running) errors++;}
   public void CompleteTrial() {if (!running) return;float seconds=Time.unscaledTime-startedAt; running=false;
      Debug.Log($"LAB02_RESULT seconds={seconds:F2} errors={errors} seed={pairIndexSum}"); }}
```

1. Wpisz sumę numerów indeksów pary
2. Przyciski UI wywołują StartTrial(), RegisterError(), CompleteTrial()
3. Zanotuj trzy czasy i błędy; oblicz medianę
4. Zapisz interpretację: który koszt dominuje (upuszczenia / błędne aktywacje / teleportacja)

PUNKT KONTROLNY D:

Trzy próby, mediana czasu, błędy policzone, wniosek oparty na obserwacji.

Część F. Zadanie diagnostyczne

Wyznaczenie wariantu:

$$\text{wariant} = (\text{numer_indeksu_1} + \text{numer_indeksu_2}) \bmod 6$$

Wariant	Objaw	Hipotezy do rozróżnienia
A	Moduł wypada z dłoni albo drga	Rigidbody; Movement Type; kolizje; dwa interaktory
B	Moduł obraca się nienaturalnie po chwycie	Błędny Attach; skala rodzica; śledzenie obrotu
C	Promień wybiera blat zamiast panelu	Raycast Mask; Interaction Layer; kolejność trafień
D	Teleportacja działa na niedozwolonej powierzchni	Warstwa podłoża; maska promienia; Collider
E	Przycisk świeci się, lecz nie reaguje	Brak UI Press; zły Event System; zduplikowane zdarzenie
F	Obrót skokowy i płynny działają równocześnie	Dwa providery; wspólna akcja; duplikat prefabu

1. Opisz objaw mierzalnie (np. „Moduł pada co 2 sekundy")
2. Sformułuj dwie hipotezy z różnymi przewidywaniami
3. Wykonaj test rozstrzygający (log, Inspector, API)
4. Wprowadź jedną minimalną poprawkę
5. Powtórz pomiar, zapisz wynik przed i po
6. git commit -m "fix(lab02): <przyczyna>"

PUNKT KONTROLNY E:
Diagnoza zawiera objaw, dwie hipotezy, test, przyczynę, wynik. Commit opisuje przyczynę.

Checkpoint 3.5

Wymagania dodatkowe (oprócz checkpointu 3.0):

✓ Panel UI: wszystkie trzy przyciski działają dokładnie raz na poprawną aktywację

✓ Tekst Status zmienia się: gotowy → montaż → alarm → ukończono

✓ Wizualna informacja zwrotna: Hover i Select zmieniają kolory przycisków i modułu

✓ Dźwiękowa informacja: różne dźwięki dla osadzenia i błędu

✓ Haptyka: zarejestrowana w logu (Event System ma dostęp do haptic feedback)

✓ Brak wielokrotnych wywołań zdarzeń (zbadano logem)

Weryfikacja (prowadzący, max 90s):
✓ Checkpoint 3.0: wszystkie kryteria
✓ Panel UI: przyciski działają, tekst zmienia się
✓ Sprzężenie zwrotne: wizualne + dźwięk + haptyka (log)
Zaliczenie: 2/3 dodatkowe kryteria (+ Checkpoint 3.0)

Checkpoint 4.0

Wymagania dodatkowe (oprócz checkpointów 3.0 + 3.5):

✓ Pełny scenariusz operatorski bez błędów logiki

✓ Trzy próby pomiarowe z medianą czasu i liczbą błędów

✓ Interpretacja wyników: którzy koszt jest dominujący i dlaczego

✓ Diagnoza Części F: hipoteza potwierdzenia, test rozstrzygający

✓ Commit z poprawką, wynik przed i po porównane

Weryfikacja (prowadzący, max 90s demonstracja):
✓ Checkpointy 3.0 + 3.5: wszystkie kryteria
✓ Scenariusz: Reset → Start → Chwyt → Osadzenie → Potwierdź
✓ Trzy próby: mediana + błędy + interpretacja
✓ Diagnoza: test rozstrzygający i poprawka
Zaliczenie: 2/4 dodatkowe kryteria (+ poprzednie). Ocena finalna: 4.0

Checkpoint 4.5

Wymagania dodatkowe (oprócz checkpointów 3.0 + 4.0):

✓ Rotacja Quest 3 (lub demonstracja prowadzącego): jedną skróconą próbę

✓ Weryfikacja rzeczywistej haptyki (kontroler transmituje impuls)

✓ Porównanie symulator ↔ Quest: rozbieżności w poza, przyciskach, haptyce, opóźnieniu

✓ Propozycja optymalizacji: wybór Movement Type (Velocity vs Kinematic) z pomiarem

✓ Analiza ergonomii: punkt Attach, kąt chwytu, odległość od panelu

Weryfikacja (prowadzący, max 90s):
✓ Checkpointy 3.0–4.0: wszystkie kryteria
✓ Quest 3: haptyka + obserwacja rozbieżności
✓ Porównanie Movement Type: mediana czasu dla obu trybów
✓ Analiza ergonomii: wniosek dla następnego projektu
Zaliczenie: 2/4 dodatkowe kryteria (+ poprzednie). Ocena finalna: 4.5

Checkpoint 5.0

Wymagania dodatkowe (oprócz checkpointów 3.0 + 4.0 + 4.5)

✓ Tryb dostępności: zamiana ręki dominującej, regulacja kąta snap turn, alternatywny kanał potwierdzenia

✓ Test: zmiana trybu nie psuje podstawowego scenariusza

✓ Porównanie dwóch Movement Type: wyniki pomiarowe (czas, stabilność, kolizje)

✓ Zaproponowana zmiana UI lub przepływu opartą na wynikach pomiarów

✓ Wnioski dla następnych projektów interaktywnych w VR

Weryfikacja (prowadzący, max 90s):
✓ Checkpointy 3.0–4.5: wszystkie kryteria
✓ Tryb dostępności: funkcjonalny, nie psuje zadania
✓ Porównanie Movement Type: dane + interpretacja
✓ Propozycja zmian + wnioski
Zaliczenie: 2/4 dodatkowe kryteria (+ poprzednie). Ocena finalna: 5.0

Oddanie

Nie przygotowuj osobnego sprawozdania. Uzupełnij plik evidence/lab02.md i wykonaj commit.

```
# Laboratorium 2 - <para / nazwiska>
## Identyfikacja pary
- Numer indeksu student 1: _____
- Numer indeksu student 2: _____
- Suma numerów: _____
- Przypisany wariant diagnostyczny: ___ (A/B/C/D/E/F)
## Wersje i scena
- Unity / XRI / OpenXR: ___ / ___ / ___
- Scena i tryb uruchomienia: ___ / ___
## Interakcje
- Direct Grab: obiekt / maska / Attach: ___ / ___ / ___
- Ray: cele / maska / reguła konfliktu: ___ / ___ / ___
- Socket: akceptowany obiekt / zdarzenie potwierdzenia: ___ / ___
## Lokomocja
- Teleportation Area / Anchor: ... / …
- Obrót skokowy: ...°
- Aktywne providery: …
## Wyniki
| Próba | Czas [s] | Upuszczenia | Błędne aktywacje | Nieudane teleportacje |
|---:|---:|---:|---:|---:|
| 1 | ... | ... | ... | ... |
| 2 | ... | ... | ... | ... |
| 3 | ... | ... | ... | ... |
- Mediana czasu: ... s
- Dominujący koszt i dowód: …
## Quest 3 (jeśli dostępny)
- Zweryfikowane: …
- Różnica względem symulatora: …
## Diagnoza (wariant ___)
- Objaw / hipotezy: ... / …
- Test / wynik: ... / …
- Poprawka / wynik po poprawce: ... / …
## Wnioski
1. …
2. …
3. …
![Scena laboratorium 2](img/lab02-scene.png)
```

Każda osoba w parze musi mieć co najmniej jeden merytoryczny commit.

Szybka weryfikacja (prowadzący, max 90 s na checkpoint)

Kryterium	Dowód podczas demonstracji	Punkty

Chwyt i gniazdo	Bezpośredni chwyt, poprawny Attach, selektywny Socket	0–2
Promień i UI	Panel działa, środowisko nie przypadkowo wybierane	0–2
Lokomocja	Dozwolony obszar, kotwica, obrót 30°, brak konfliktu providerów	0–2
Pomiar i interpretacja	Trzy próby, mediana, błędy i wniosek oparty na wyniku	0–2
Diagnoza i Git	Hipotezy, test, minimalna poprawka, wkład obu osób	0–2

ZASADA ZALICZENIA: Punkt kontrolny jest zaliczony od 6/10 pkt, pod warunkiem uzyskania co najmniej 1 pkt w każdym kryterium. Brak testu na Quest 3 nie obniża wyniku, jeżeli para poprawnie oznaczy ograniczenie w evidence.

Pytania kontrolne

1. Wyjaśnij różnicę między Interactor, Interactable i XR Interaction Manager.
2. Dlaczego Collider i Interaction Layer Mask pełnią inne role?
3. Kiedy punkt Attach Transform jest ważniejszy niż środek bryły?
4. Dlaczego promień i chwyt bezpośredni mogą wejść w konflikt?
5. Czym różni się Teleportation Area od Teleportation Anchor?
6. Dlaczego symulator nie wystarcza do potwierdzenia haptyki?
7. Która miara lepiej opisuje typowy czas z trzech prób przy jednej wartości odstającej?
8. Dlaczego jednoczesny Snap Turn i Continuous Turn utrudniają diagnozę?

Exit ticket (odpowiedź indywidualnie)

Jedno zdanie: Które ograniczenie interakcji w Twojej scenie zapobiega najpoważniejszemu błędowi operatora i jaki test dowodzi, że ograniczenie rzeczywiście działa?

Materiały rozszerzające (po ukończeniu checkpointu 3.0)

1. Porównaj dwa tryby Movement Type: wykonaj trzy próby dla każdego, zestawienie medianę czasu, upuszczeń, stabilności kolizji. Uzasadnij wybór dla stanowiska przemysłowego.
2. Dodaj dwuręczne przenoszenie większego elementu albo ograniczenie ruchu modułu do jednej osi. Udowodnij testem, że transformacja nie narusza ograniczenia.
3. Zaprojektuj tryb dostępności: możliwość zamiany ręki, regulacja kąta snap turn, alternatywny kanał potwierdzenia. Sprawdź, czy zmiana nie psuje podstawowego scenariusza.

Diagnostyka typowych problemów

Objaw	Sprawdź w tej kolejności	Nie rób
Brak Hover/Select	Collider → Interactor → maski → Input Actions → Manager	Nie dodawaj drugiego Managera
Obiekt przeskakuje po chwycie	Attach → skala rodziców → track pos/rot → Movement Type	Nie kompensuj offsetem
Obiekt przenika przez blat	Collider → Rigidbody → Collision Detection → Movement Type	Nie wyłączaj wszystkich kolizji
Promień trafia zły cel	Raycast Mask → Interaction Layer → kolejność → UI raycaster	Nie ustawiaj Everywhere/Everything
UI świeci, lecz nie reaguje	Event System → Input Module → Graphic Raycaster → UI Press	Nie duplikuj Event System
Teleportacja nie działa	Provider → akcja → ray → maska → Collider → Area/Anchor	Nie dodawaj drugiego XR Origin
Obrót wykonuje dwa skoki	Liczba providerów → callbacków → mapowanie osi	Nie zmieniaj kąta bez diagnozy
Brak haptyki na Quest	Obsługa urządzenia → source → zdarzenie → amplituda → runtime	Nie traktuj logu jako dowodu fizycznego

Lista końcowa studenta - przed oddaniem

☐ Pracuję na zamrożonych wersjach Unity i pakietów.

☐ W scenie istnieje jedna aktywna kamera główna i jeden właściwy XR Interaction Manager.

☐ Module_A ma Collider, Rigidbody, poprawny Attach i ograniczoną maskę interakcji.

☐ Direct Interactor chwyta moduł, Ray obsługuje panel bez wybierania środowiska.

☐ Socket_A akceptuje wyłącznie właściwy moduł i jednoznacznie potwierdza osadzenie.

☐ Panel pokazuje stan zadania; Start, Reset, Potwierdź działają dokładnie raz.

☐ Teleportacja działa wyłącznie w dozwolonym obszarze; obrót skokowy = 30°.

☐ Zapisałem trzy próby, medianę czasu, błędy i wniosek oparty na obserwacji.

☐ Diagnoza zawiera objaw, dwie hipotezy, test, minimalną poprawkę i wynik.

☐ Evidence/lab02.md i zrzut znajdują się w repozytorium.

☐ Każda osoba w parze wykonała co najmniej jeden merytoryczny commit.

☐ W skrypcie Lab02Trial wpisałem poprawnie sumę numerów indeksów pary.

Bibliografia

1. Unity Technologies, „XR Interaction Toolkit 3.1.3: podręcznik".
2. Unity Technologies, „XR Grab Interactable".
3. Unity Technologies, „Create a scene with locomotion".
4. Unity Technologies, „Teleportation Provider".
5. Unity Technologies, „Simple Haptic Feedback".
6. Unity Technologies, „XR Interaction Simulator overview".

Uwaga o wersjach: W razie rozbieżności interfejsu pierwszeństwo ma wersja pakietu zapisana w Packages/packages-lock.json oraz odpowiadająca jej dokumentacja. Model pracy XRI jest stabilny, ale nazwy prefabów mogą różnić się między wydaniami.