

Politechnika Świętokrzyska
w Kielcach
Wydział Elektrotechniki, Automatyki i Informatyki

Rzeczywistość Wirtualna i Rozszerzona

Instrukcja laboratoryjna 3

Lokomocja

Przygotował:
mgr inż. Michał Wójcik

Kielce, 2025

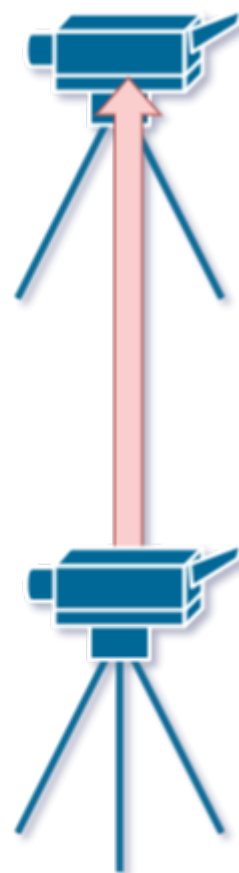
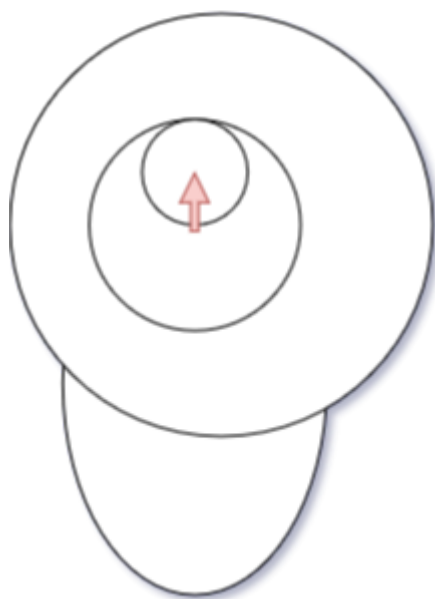
1. Lokomocja w Wirtualnej Rzeczywistości

Osadzenie w wirtualnym świecie użytkownika interfejsu człowiek-komputer jakim jest zestaw VR niesie za sobą problemy natury przestrzennej. W ograniczonej przestrzeni fizycznej jaką jest pokój lub sala laboratoryjna trudno o zapewnienie swobody ruchu dla niektórych doświadczeń, takich jak spacer po rozległych łąkach. Czerpiąc inspirację z klasycznych aplikacji i gier komputerowych, w wirtualnej rzeczywistości obchodzi się często te ograniczenia, pozwalając użytkownikowi poruszać się poprzez gesty i przyciski na kontrolerze ruchu.

Z uwagi na ograniczenia technologiczne, potrzeby projektowe lub komfort i odczucia graczy na przestrzeni lat opracowano wiele różnych metod poruszania się po wirtualnej przestrzeni.

1.1. Lokomocja ciągła

Pozornie najprostszą w realizacji jest lokomocja ciągła, jaką można spotkać w klasycznych grach komputerowych typu FPS (First-Person Shooter). Korzystając z dwuosioowego krzyżaka, gałki lub trackpada, użytkownik wybiera kierunek poruszania się, a jego pozycja w świecie rzeczywistym liniowo jest przesuwana o wektor o kierunku i zwrocie wyznaczonym przez kontroler.



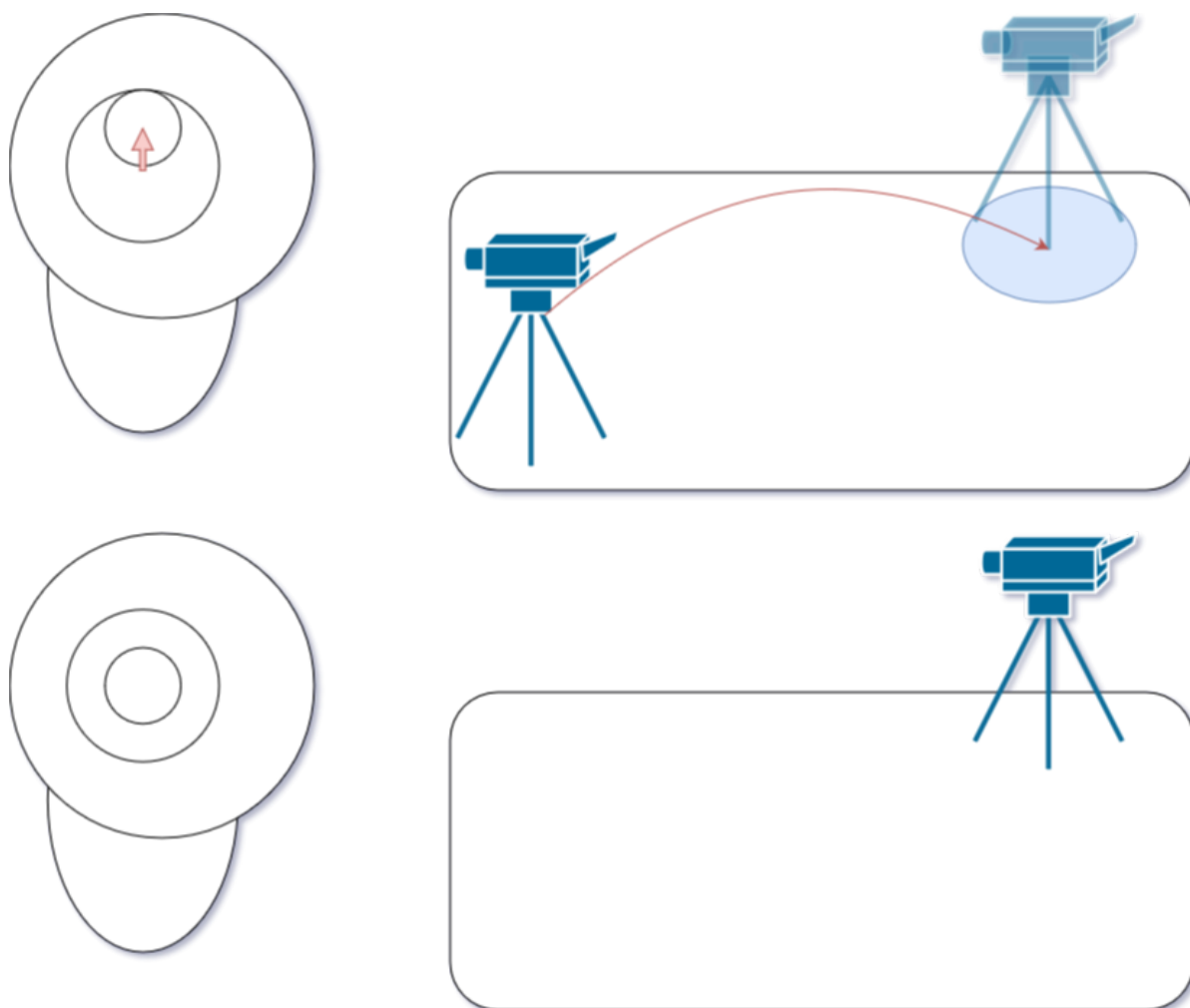
W przypadku lokomocji ciągłej należy jednak wziąć pod uwagę kierunek, w którym użytkownik zamierza iść, czyli jego orientację; ta z kolei może być ustalona zarówno na podstawie rotacji głowy jak i kontrolerów. Zazwyczaj programista wybiera lub udostępnia użytkownikowi wybór między trzema metodami określania orientacji, a każda ma swoje wady, zalety i zastosowania:

Metoda orientacji	Wady	Zalety	Zastosowania
Według głowy	Brak możliwości rozglądania się na boki podczas poruszania się	Mniejsza szansa na wystąpienie choroby symulatorowej niż przy orientacji według kontrolera	Wirtualne spacer y i doświadczenia wymagające ciągłego ruchu kontrolerami w trakcie poruszania się; gdy gracz chce iść wzdłuż linii wzroku
Według kontrolera	Możliwy dysonans między spodziewanym a faktycznym ruchem, powodujący rozkojarzenie i mdłości	Możliwość dowolnego rozglądania się na boki i precyzyjnego poruszania się w dowolnym kierunku.	Doświadczenia z nastawieniem na celowanie z użyciem kontrolera, np. gry strzelankowe; gdy gracz chce iść w kierunku wskazywanym dłonią
Według tułowia	Wymaga dodatkowego sprzętu lub algorytmów aproksymacyjnych	Najbardziej naturalny sposób poruszania się, pozwala na dowolne poruszanie kontrolerami i głową	Zaawansowane doświadczenia i gry, stawiające na realizm i wolność poruszania się

1.2. Teleportacja

Lokomocja ciągła może powodować lub potęgować tak zwaną chorobę symulatorową, (ang. *motion sickness*, *simulator sickness*) u niektórych użytkowników. Alternatywnym sposobem poruszania się w VR jest teleportacja, czyli nagły przeskok z jednego miejsca w drugie. Jest to jedna z najbardziej komfortowych metod lokomocji, ponieważ nie sprawia wrażenia ruchu, gdy użytkownik nie porusza własnym ciałem.

Lokomocja poprzez teleportację implementowana jest z użyciem wskaźnika, najczęściej zakrzywionego po łuku jak trajektoria pocisku, z okręgiem na samym końcu wskazującym pozycję, na którą użytkownik ma trafić. Użytkownik celuje wskaźnikiem przez kontroler ruchu, przytrzymując przycisk. Puszczanie tego przycisku akceptuje teleportację i przenosi użytkownika natychmiast w nowe miejsce.



Teleportacja oprócz mniejszego wpływu na zmysły ma własne zastosowania, niemożliwe przy lokomocji ciągłej - pozwala na przykład na ustawienie **punktów zainteresowania** (ang. *Point of Interest*), czyli wyznaczonych przez twórcę doświadczenia pozycji, do których teleportacja może zostać albo zupełnie ograniczona albo ułatwiona względem teleportacji po przestrzeni otwartej. Z drugiej strony teleportacja umożliwia poruszanie się pomiędzy poziomami, coś czego lokomocja ciągła nie może zapewnić bez dodatkowych funkcjonalności (np. skok) lub wspomagających metod lokomocji.

Teleportacji jako świadomego wyboru projektowego warto użyć w doświadczeniach spokojnych, pozwalających użytkownikowi na dowolną eksplorację świata przedstawionego, np. wirtualnych muzeach, przestrzennych interfejsach użytkownika (przykład: SteamVR Home) lub przestrzennych grach łamiących granice.

1.3. Lokomocja skokowa (Cloudstep)

Lokomocja skokowa łączy w sobie cechy zarówno lokomocji ciągłej jak i teleportacji. Sterowanie odbywa się podobnie jak w przypadku lokomocji ciągłej - poruszenie kontrolką kierunkową przesuwają gracza w danym kierunku, lecz zamiast ruchu o stałej prędkości kamera krokowo, jednorazowo przesuwają się o określoną odległość. Lokomocja skokowa pozwala na realizację doświadczeń zaprojektowanych z myślą o lokomocji ciągłej, jednocześnie minimalizując dysonans między oczami a błędnikiem, a

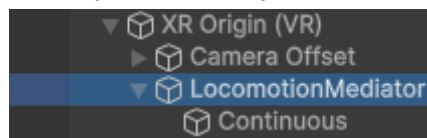
przez to również minimalizując objawy choroby symulatorowej. W zależności od implementacji, "skoki" mogą być realizowane natychmiastowo - tak, jak w przypadku teleportacji - lub szybkim ruchem, "wślizgiem" (*ang. dash*).

2. Lokomocja w Unity XR Interaction Toolkit

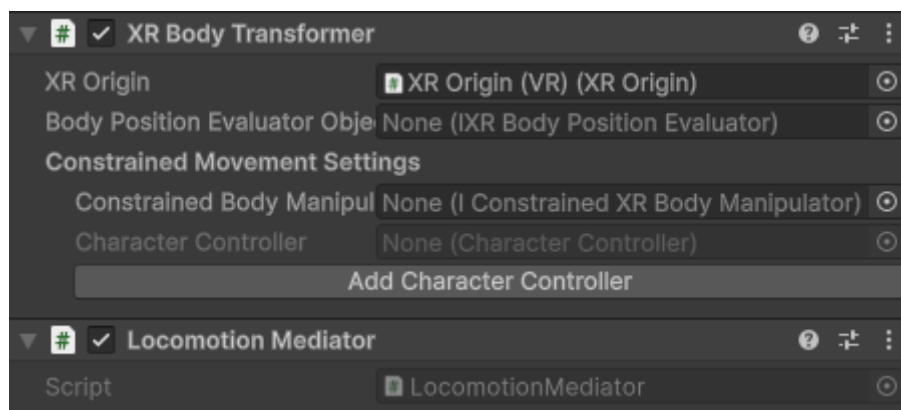
XR Interaction Toolkit posiada wbudowane rozwiązania dla podstawowych rodzajów lokomocji: ciągłej i teleportacji. Na laboratorium 3 omówiona zostanie implementacja lokomocji ciągłej, gdyż lokomocja poprzez teleportację wymaga też znajomości systemu interakcji.

2.1. Locomotion Mediator i Locomotion Provider

Mediator Lokomocji (**LocomotionMediator.cs**) to przygotowany komponent pośredniczący w procesie zarządzania poruszaniem kamery przez XR Origin. Dawniej jego rolę spełniał LocomotionSystem, który wciąż jest dostępny w XR Interaction Toolkit, ale jest oznaczony jako *deprecated*, to znaczy wycofany. Locomotion Provider to klasa komponentów, które komunikują się z XR Body Transformerem poprzez Locomotion Mediator i reagują na parametry wejściowe, takie jak akcje Input Systemu (przykład - Continuous Move Provider (**ContinuousMoveProvider.cs**)). Komponent LocomotionMediator należy utworzyć jako obiekt w hierarchii obiektu XR Origin i przekazać go przez dowiązanie w Unity.

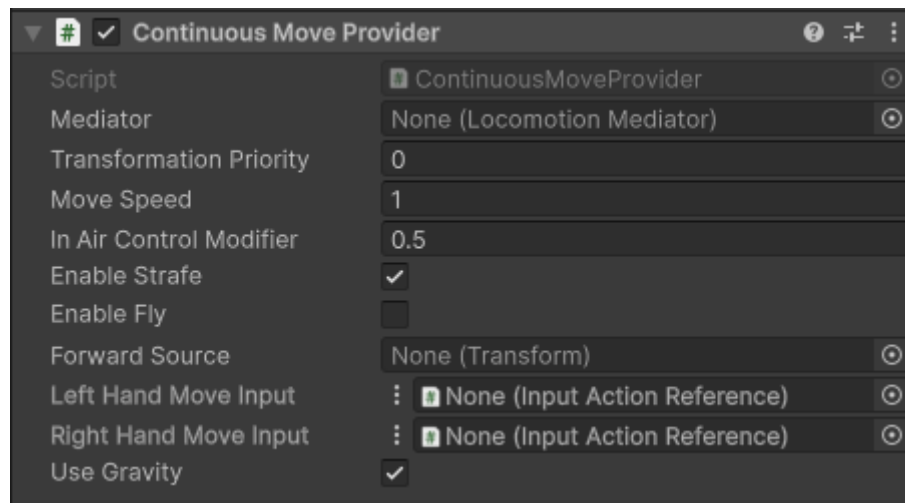


W odróżnieniu od dawnego LocomotionSystemu, Mediator potrzebuje do działania dodatkowego komponentu - **XR Body Transformer**. Komponent ten bezpośrednio zarządza poruszaniem obiektami w XR Origin na podstawie informacji od Providerów, którzy mogą zlecać mu transformacje (ruchy) do wykonania w następnej klatce animacji.



XR Body Transformer potrzebuje odwołania do XR Origin, a pozostałe parametry są opcjonalne i nie zostaną poruszone na laboratorium.

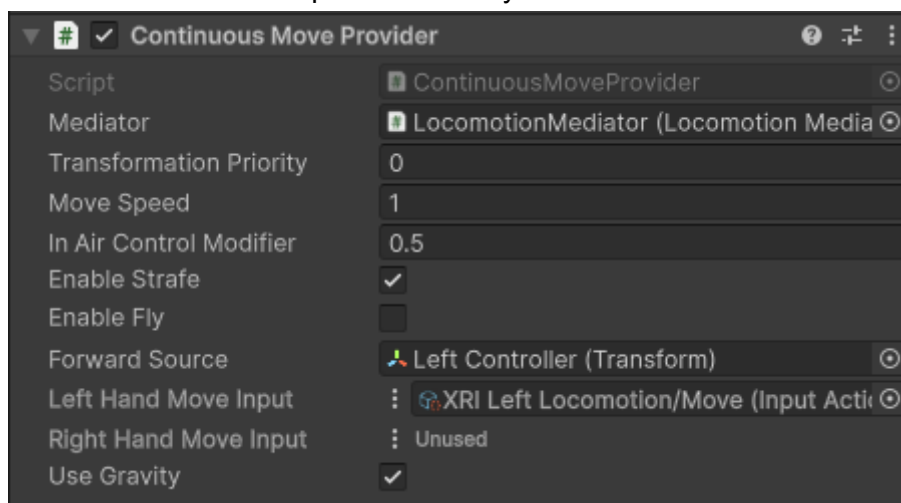
Z kolei Continuous Move Provider (Action-Based) trzeba utworzyć w hierarchii LocomotionSystemu i przekazać go przez dowiązanie w Unity

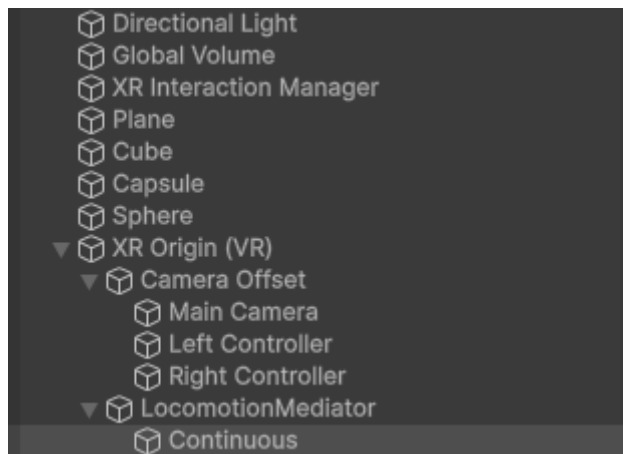


W ustawieniach Continuous Move Provider należy też przekazać:

- **Transformation Priority** - w przypadku obsługi wielu Providerów, parametr ten oznacza priorytet w obsłudze żądań przez XR Body Transformer: im niższa liczba tym wyższy priorytet,
- **Move Speed** - Prędkość, z jaką maksymalnie można się poruszać,
- **In Air Control Modifier** - parametr regulujący stopień kontroli nad ruchem gdy użytkownik znajduje się w powietrzu; przyjmuje wartości od 0 do 1 włącznie,
- **Enable Strafe** - Czy można poruszać się na boki,
- **Enable Fly** - Ciekawostka: latanie poprzez skierowanie wektora ruchu w górę lub w dół (automatycznie wyłącza grawitację),
- **Use Gravity** - Czy Provider ma respektować grawitację
- **Gravity Application Mode** - Kiedy kontroler ma zaaplikować grawitację, natychmiast po rozpoczęciu gry czy po pierwszej próbie poruszenia się
- **Forward Source** - Obiekt źródłowy, na podstawie którego określana będzie orientacja ruchu
- **Left/Right Hand Move Input** - Akcja, która wywołuje ruch na lewym lub prawym kontrolerze.

Przykładowe ustawienia komponentu i sceny:



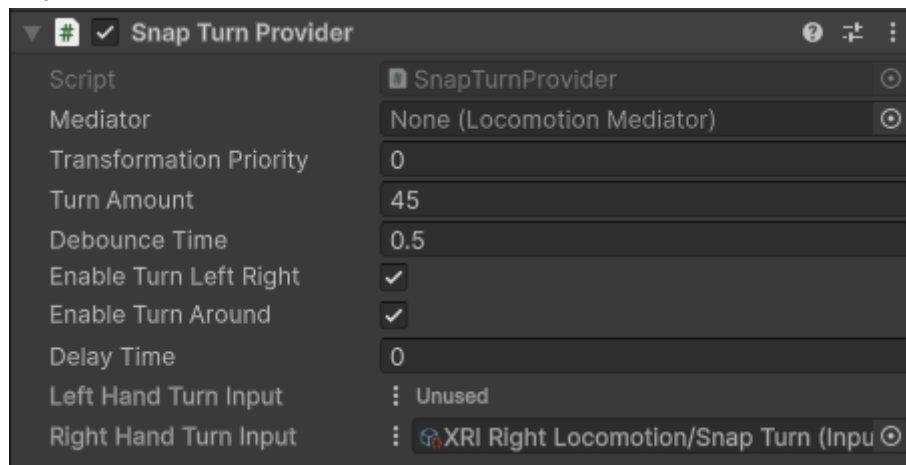


2.2. Turn Provider

Podczas interakcji ze światem w wirtualnej rzeczywistości warto zadbać o sposób na wirtualne obracanie się w miejscu, by uniknąć np. zaplątania kabli. XR Interaction Toolkit umożliwia w łatwy sposób implementację obrotu poprzez kolejny Provider dla LocomotionMediatora. Do wyboru mamy Continuous Turn Provider

(**ContinuousTurnProvider.cs**) oraz Snap Turn Provider (**SnapTurnProvider.cs**):

- Continuous Turn Provider - pozwala na obrót ciągły podczas trzymania kontrolki, daje więcej wolności, ale może być dezorientujący,
- Snap Turn Provider - obraca kamerę w inkrementach o określony kąt w reakcji na wciśnięcie kontrolki.



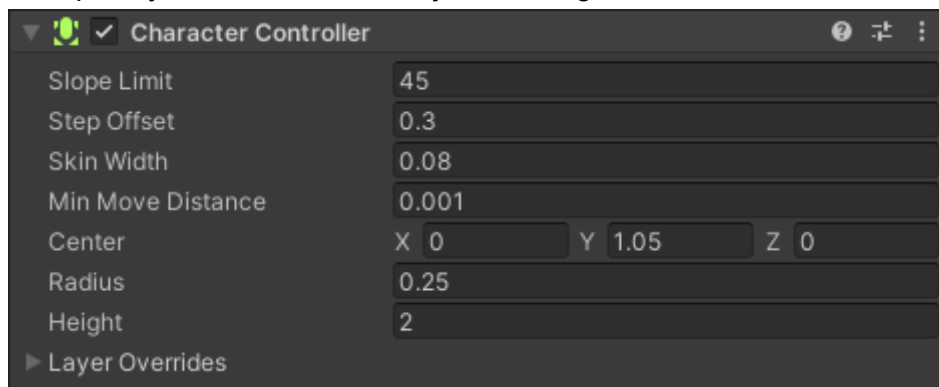
Na przykładzie Snap Turn Providera:

- **Turn Amount** - o jaki kąt ma się obrócić użytkownik przy wciśnięciu kontrolki
- **Debounce Time** - ile czasu musi minąć od puszczenia kontrolki, by móc wykonać następny obrót
- **Enable Turn Left Right** - zezwól na obrót w lewo i prawo
- **Enable Turn Around** - zezwól na obrót o 180 stopni (zazwyczaj przez wciśnięcie gałki/touchpada do siebie)
- **Delay Time** - opóźnienie między otrzymaniem sygnału od użytkownika a wykonaniem obrotu

- **Left/Right Hand Snap Turn Input** - akcja, która zostanie wykorzystana do sterowania obrotem.

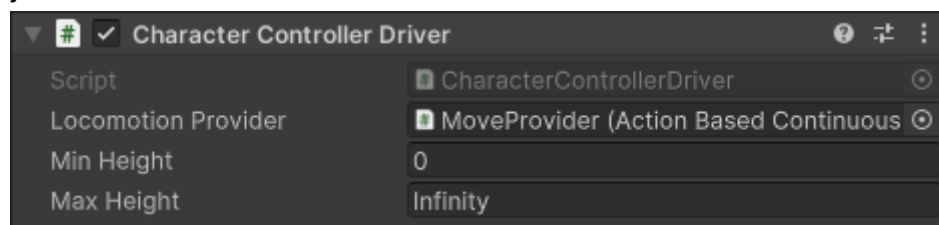
2.3. Character Controller

Grawitacja oraz kolizje z otoczeniem nie będą działać, jeśli LocomotionSystem nie będzie miał odniesienia do awatara użytkownika, lub “postaci gracza”. Rolę fizycznej reprezentacji postaci gracza w Unity spełnia wbudowany komponent Character Controller, zazwyczaj podpinany do obiektu reprezentującego taką postać. W przypadku aplikacji VR, takim obiektem jest XR Origin.



Warto zwrócić uwagę na parametry **Center** oraz **Radius**. CharacterController opisuje tymi parametrami **Collider**. Jeśli Radius będzie za szeroki lub za wąski, przeszkody będą wydawać się za duże lub za małe. Jeśli środek Collidera (Center) będzie za nisko w momencie inicjalizacji gry, można przez przypadek zapaść się pod ziemię. Aby dodać Character Controller do XR Originu można wcisnąć przycisk “Add Character Controller” w inspektorze komponentu XR Body Transformer.

Dawniej elementem spinającym Character Controller z Locomotion Providerem był komponent Character Controller Driver (**CharacterControllerDriver.cs**), który należało dołączyć to tego samego obiektu co Character Controller i przekazać mu odwołanie do Providera od ruchu. Obecnie tą rolę przejął XR Body Transformer. Ten komponent kontrolował wysokość i punkt środkowy Character Controllera na podstawie ruchu głowy - obecne systemy wbudowane w XR Interaction Toolkit są bardziej zaawansowane.



3. Zadania

1. Zaimplementuj lokomocję ciągłą zorientowaną wzdłuż kontrolera lub głowy [2 pkt].
2. Zadbaj o to, by działała grawitacja i kolizje z otoczeniem [1 pkt].
3. Zaimplementuj lokomocję ciągłą zorientowaną wzdłuż kontrolera oraz głowy: napisz skrypt umożliwiający przełączanie trybu poprzez przycisk na kontrolerze; może okazać się koniecznym dziedziczenie po klasie ContinuousMoveProvider. [4 pkt]

4. Źródła

Unity Technologies, *XR Interaction Toolkit Manual*, wersja 3.0 [online]. [n.d.], dostęp: 11.10.2025. Dostępny w Internecie:

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