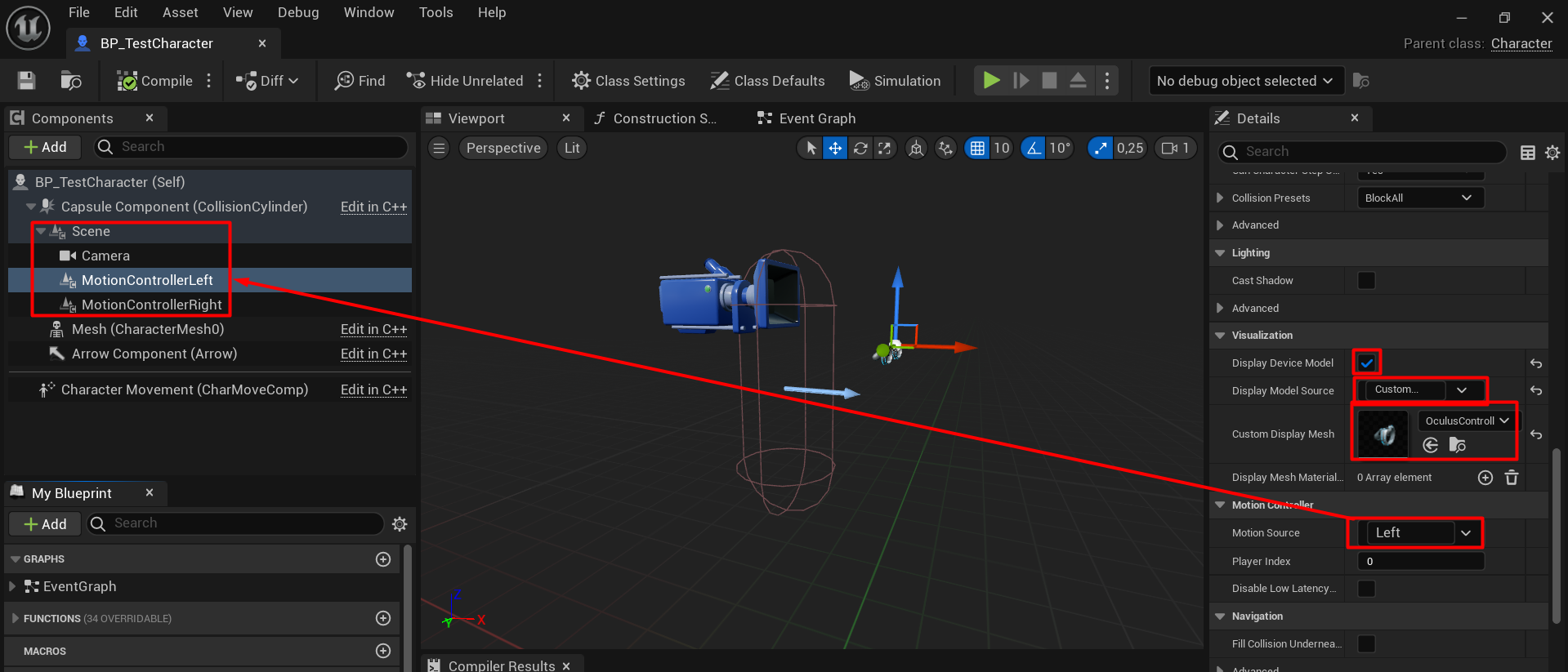
Лабораторна робота №7

**ТЕСТУВАННЯ ТА РОБОТА VR-ПРОЕКТУ БЕЗ VR-ШОЛОМУ. РЕЖИМ VR-PREVIEW ДЛЯ META QUEST 2. ТЕСТУВАННЯ ПРОЕКТУ З VR-ШОЛОМОМ. СТВОРЕННЯ ТА РОЗГОРТАННЯ APK НА META QUEST 2**

**Мета:** набуття практичних навичок роботи із режимами тестування проекту без та з VR-шоломом, створювати та інсталювати apk для Meta Quest 2.

**Хід роботи**

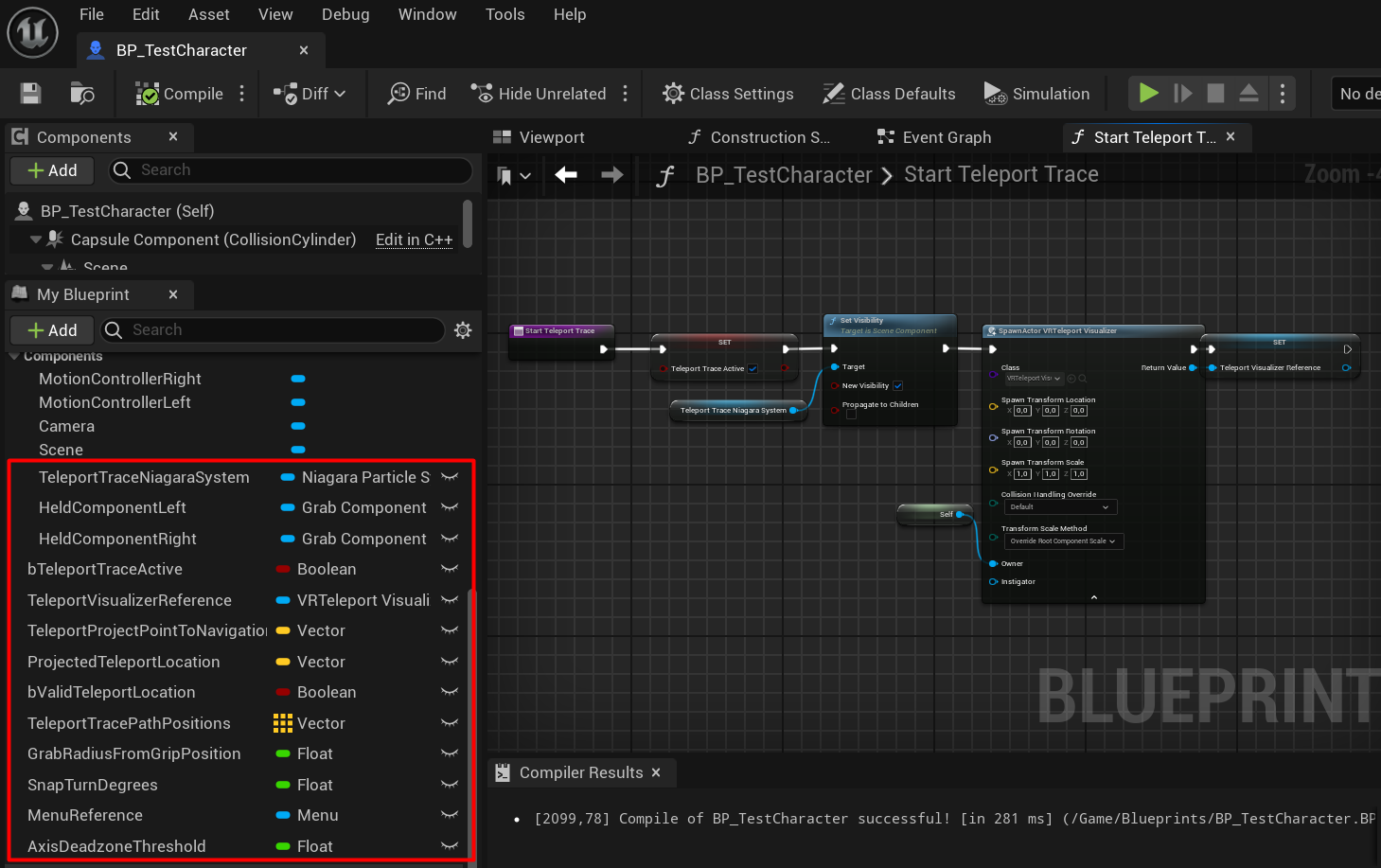
Для реалізації тестування без шолома створюємо нового персонажа *Blueprint* класу *Character*. Додаємо до нього компонент *Scene*, у який додаємо компоненти *Camera* та два компоненти *MotionController* (VR контролери). Сцену та камеру залишаємо зі стандартними налаштуваннями. А для контролерів встановлюємо флаг Display Device Model, додаємо Custom Display Mesh та обираємо відповідний Motion Source (Left для лівого, Right для правого) (рис. 1).

  
Рисунок 1 – Налаштування компонентів нового персонажа

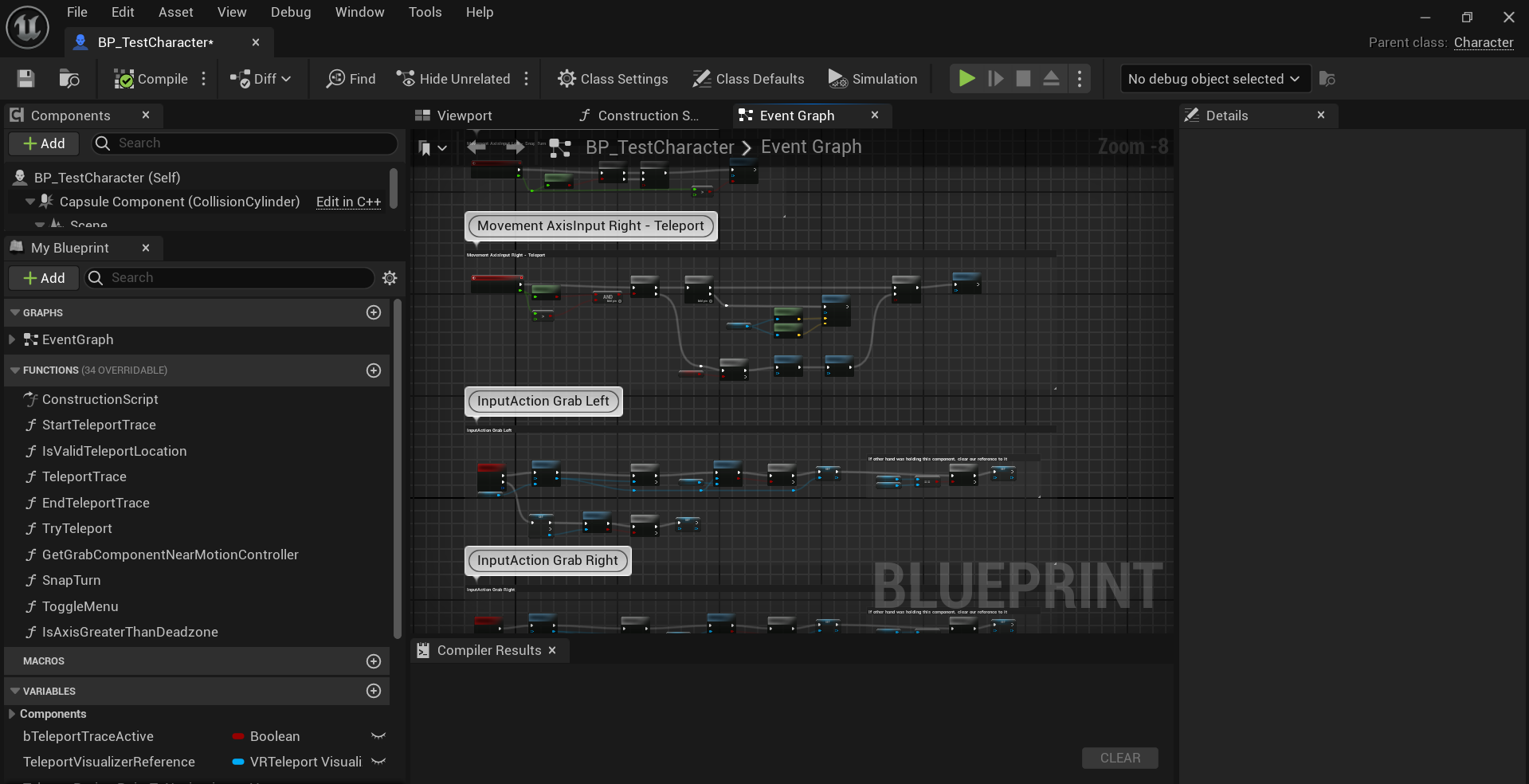
Далі відкриваємо *VR Pawn* та копіюємо усі функції до створеного персонажа (рис. 2).

  
Рисунок 2 – Копіювання функційVR Pawn до нового персонажа

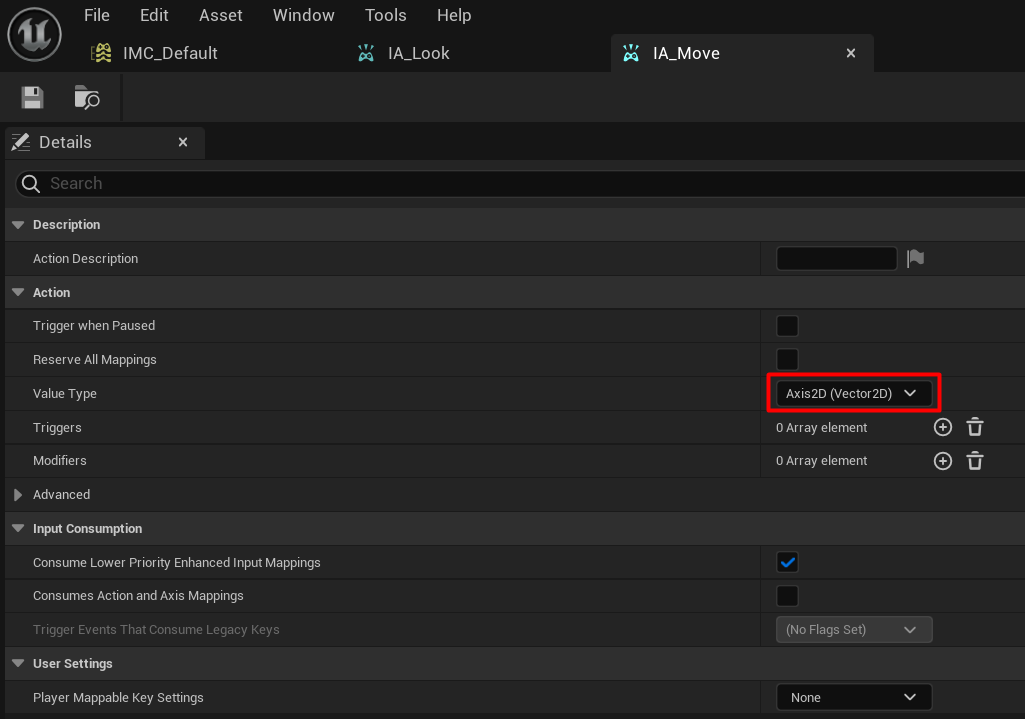
У разі виникнення помилок при компіляції слід додавати усі необхідні змінні для роботи функцій (рис. 3).

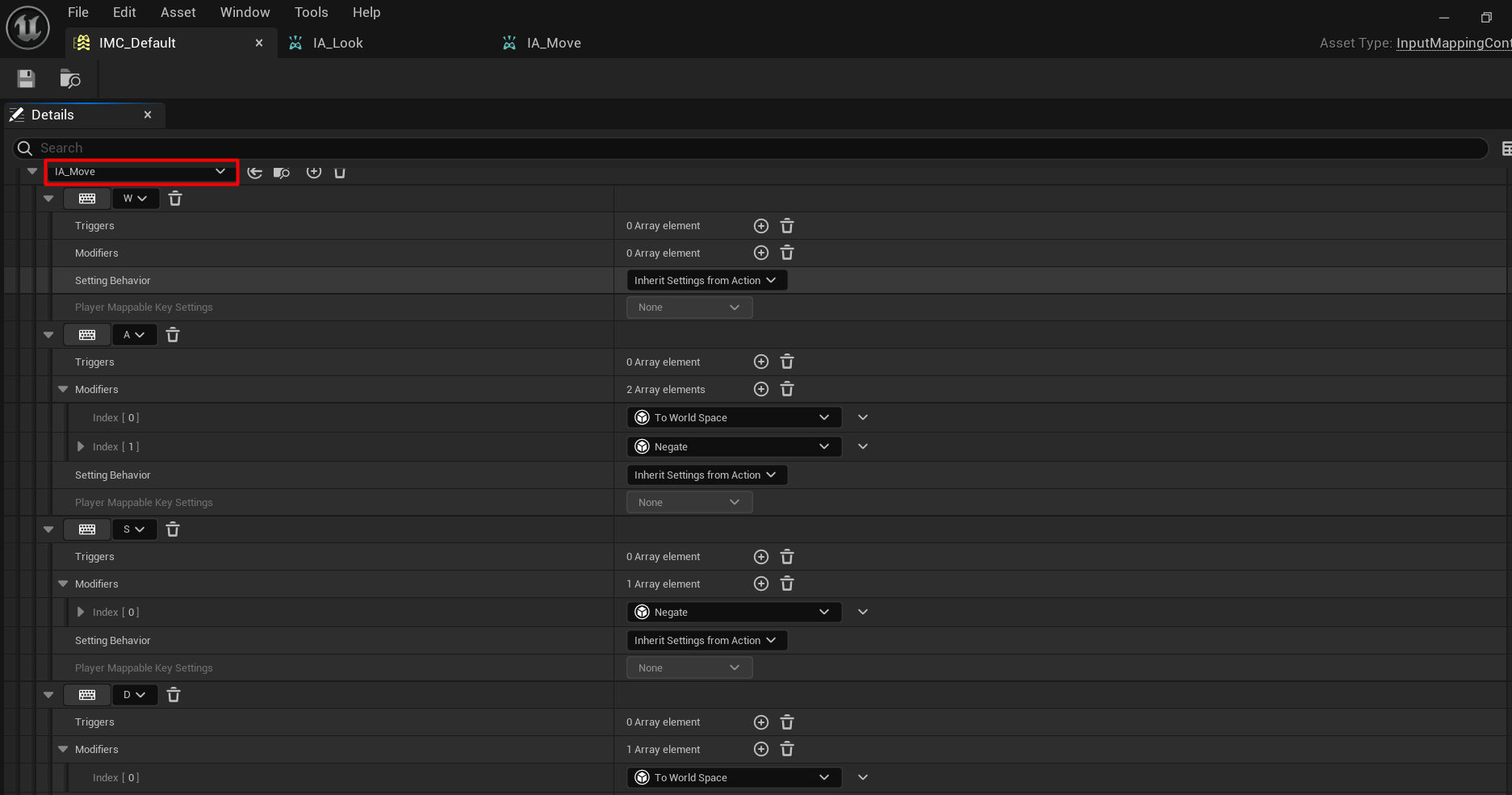
  
Рисунок 3 – Копіювання змінних

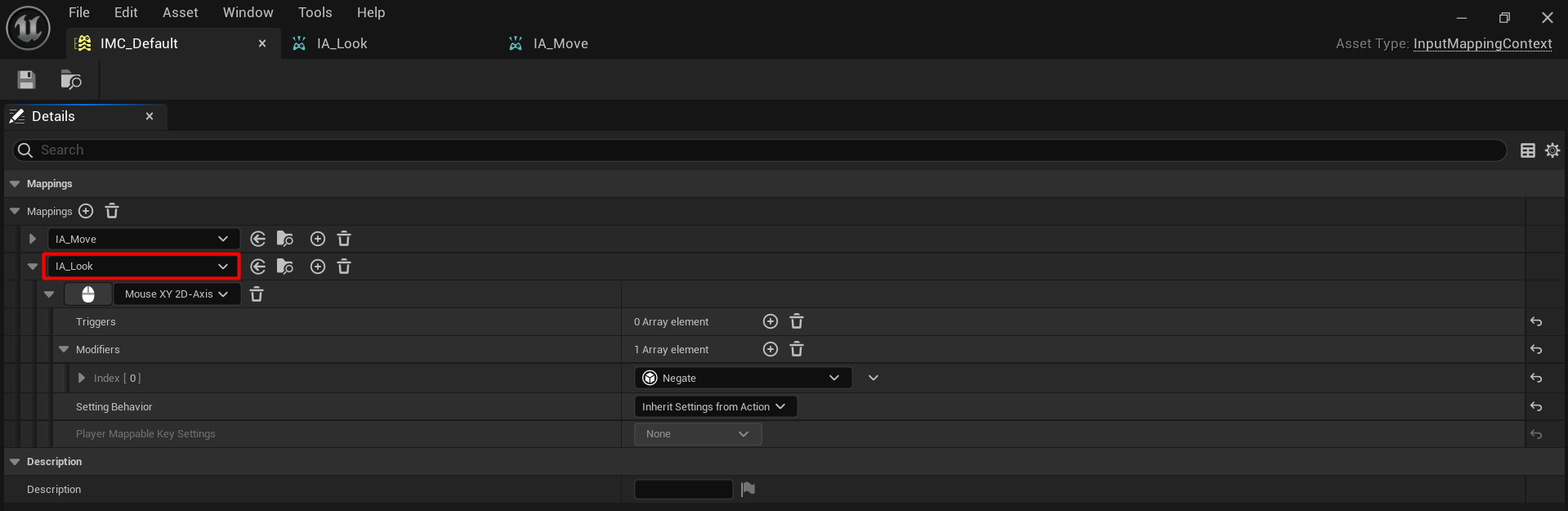
Далі необхідно скопіювати граф подій *VR Pawn* та відредагувати його для тестового персонажа (рис. 4).

  
Рисунок 4 – Копіювання графу подій

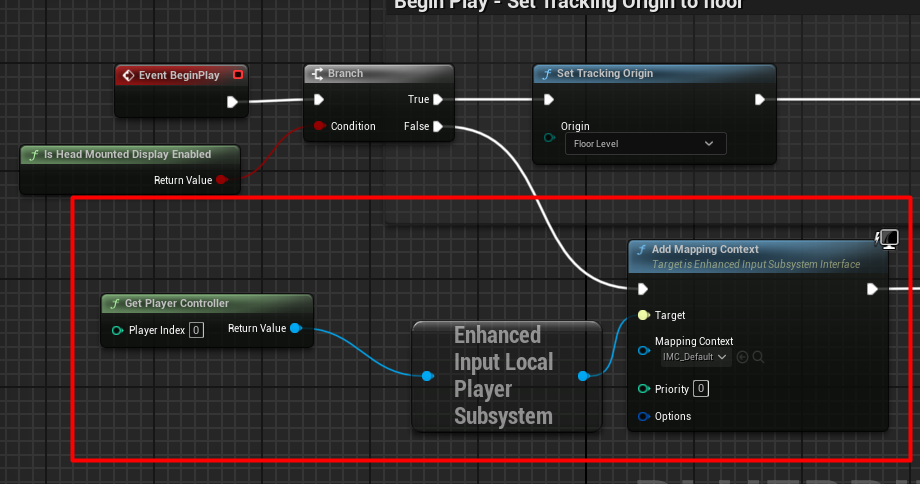
Для впровадження пересування клавіатурою та мишкою необхідно створити *Input Actions*. Пересування та погляд містять однакові налаштування *Input Actions* (рис. 5). Також необхідно створити *Input Mapping Context*, щоб задати клавіши. Для пересування задаємо стандартну карту *WASD* (рис. 6). А для погляду налаштовуємо Mouse XY 2D-Axis (рис. 7).

  
Рисунок 5 – Налаштування Input Action для пересування та погляду

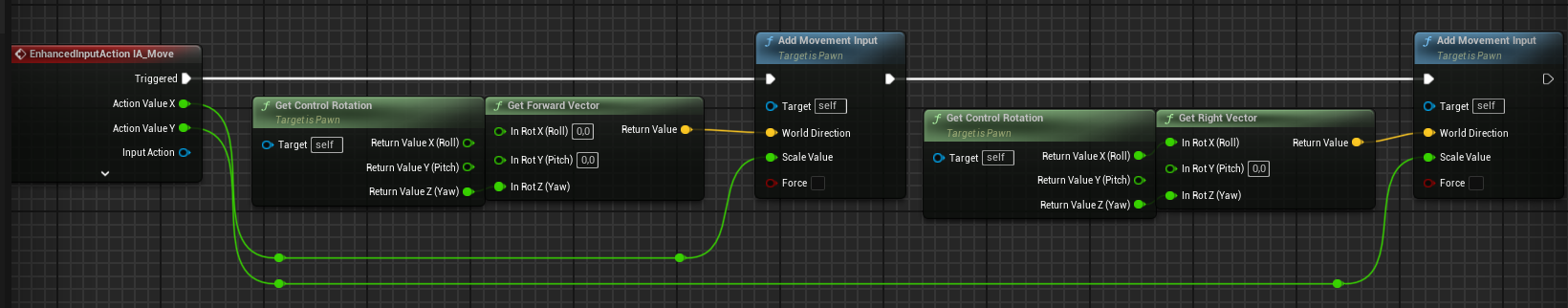
  
Рисунок 6 – Налаштування Input Mapping Context для пересування

  
Рисунок 7 – Налаштування Input Mapping Context для погляду

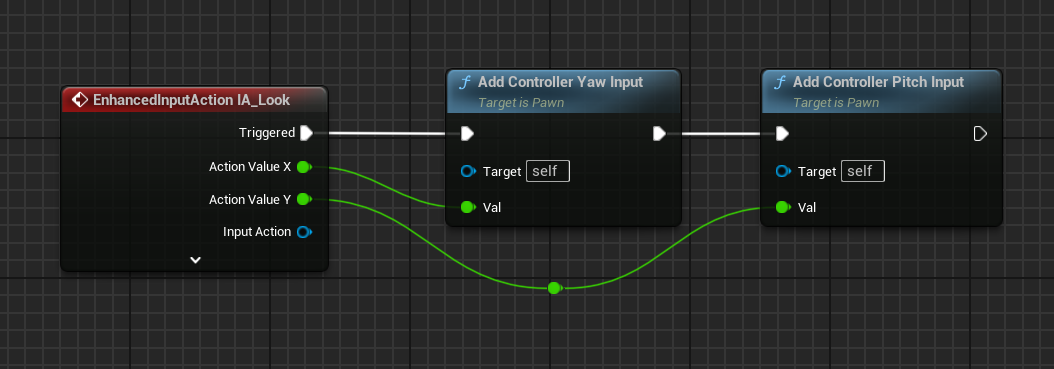
У графі подій тестового персонажу для події Begin Play створюємо умовну вітку при відключеному шоломі. Вона буде вмикати створений раніше контекст розкладки через *Add Mapping Context* (рис. 8).

  
Рисунок 8 – Підключення розкладки для персонажа

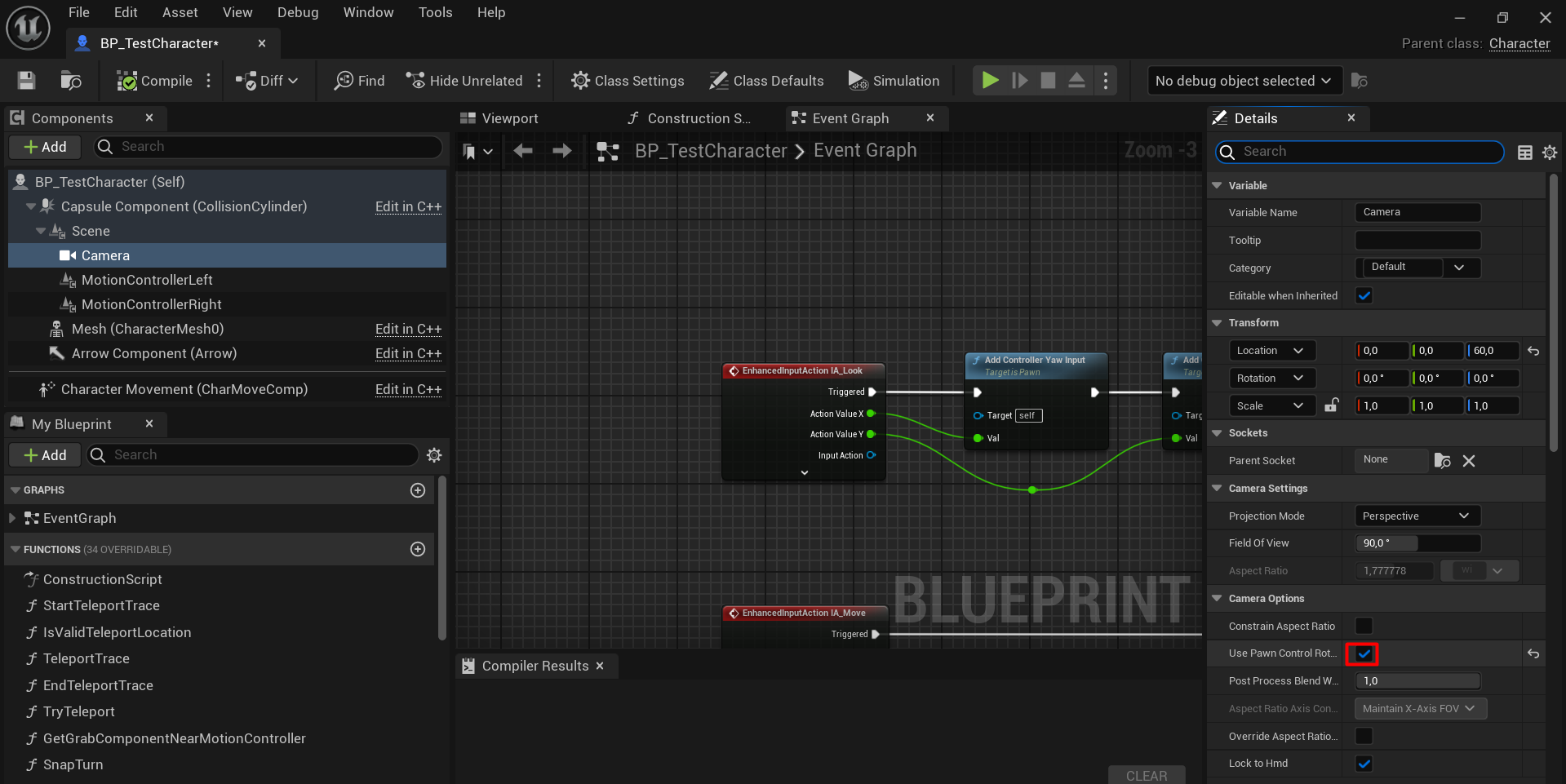
Щоб активувати пересування додамо наступний блок для події *EnchancedInputAction IA\_Move* (рис. 9).

  
Рисунок 9 – Створення логіки пересування

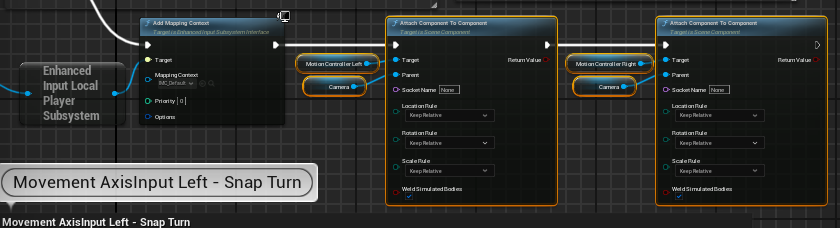
Щоб активувати погляд додамо наступний блок для події *EnchancedInputAction IA\_Look* (рис. 10).

  
Рисунок 10 – Створення логіки погляду

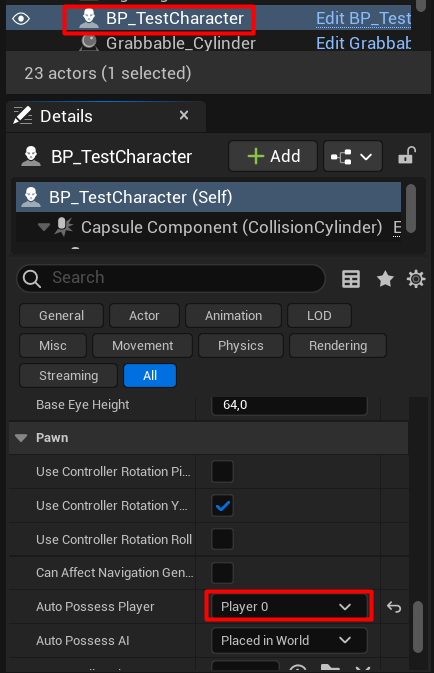
Для коректної роботи камери вмикаємо параметр *UsePawnControllRotation* (рис. 11).

  
Рисунок 11 – Вмикання обертання персонажа

Для прив’язки контролерів до камери додамо вузол *Attach Component To Component*, у якості вхідних параметрів якого виступають камера та контролери (лівий та правий) (рис. 12).

  
Рисунок 12 – Прив’язка контролерів до камери

Щоб увімкнути розкладку, переходимо до вікна гри, додаємо персонажа на рівень та встановлюємо значення параметра *Auto Possess Player* на *Player 0* (рис. 13).

  
Рисунок 13 – Вмикання розкладки для персонажа

На рисунках 13 – 14 подано результати виконання роботи.

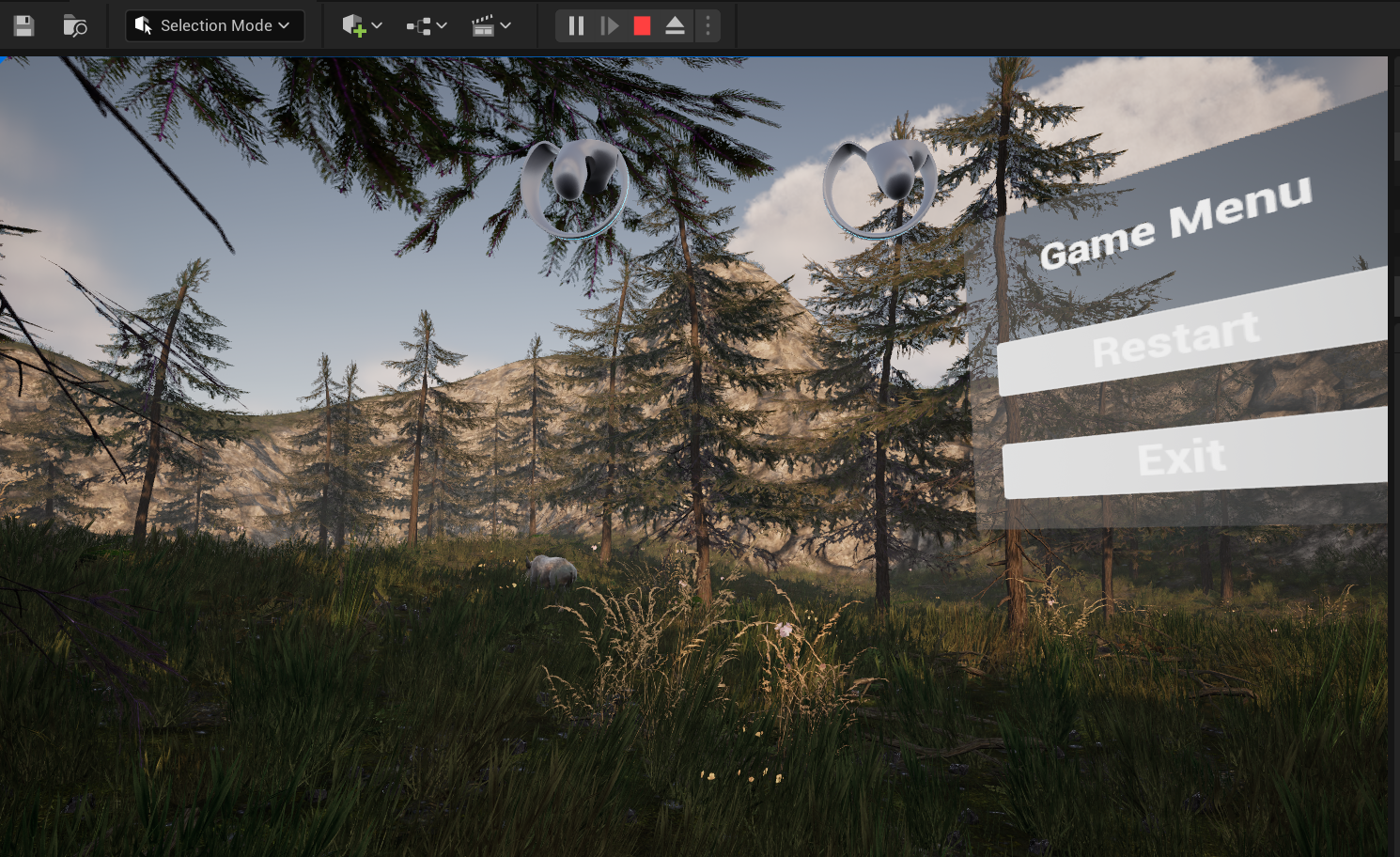
  
Рисунок 13 – Результат виконання роботи

  
Рисунок 14 – Результат виконання роботи

**Висновки**

При виконанні лабораторної роботи набуто практичні навички роботи із режимом тестування проекту без VR-шолому. Створено персонажа для тестування VR сцени без використання шолому.