МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук

«До захисту допущено»
В.о. завідувача кафедри

Ігор ШЕЛЕХОВ

_____
(підпис)

_____

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня Бакалавр

зі спеціальності 122 - Комп'ютерних наук,
освітньо-професійної програми «Інформатика»
на тему: «Навчальна інформаційна система формування практичних навичок роботи з 3d Unity»
здобувача групи ІН-02 Батюка Івана Миколайовича

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

Іван БАТЮК

_____
(підпис)

Керівник, Олег БЕРЕСТ

_____

старший викладач,
кандидат технічних наук

(підпис)

Суми – 2024

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук

«Затверджую»
В.о. завідувача кафедри

Ігор ШЕЛЕХОВ

_____
(підпис)

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня бакалавра

зі спеціальності 122 - Комп'ютерних наук, освітньо-професійної програми «Інформатика»
здобувача групи ІН-02 Батюка Івана Миколайовича

1. Тема роботи: «Навчальна інформаційна система отримання практичних навичок роботи з 3d Unity»
затверджую наказом по СумДУ від _____

2. Термін здачі здобувачем кваліфікаційної роботи _____

3. Вхідні дані до кваліфікаційної роботи _____

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)
1) Аналіз проблеми предметної області, постановка й формування завдань дослідження. 2) Аналіз об'єкту дослідження. 3) Розробка інформаційного забезпечення для створення заготовок на заняття. 4) Аналіз результатів.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) _____

6. Консультанти до проекту (роботи), із значенням розділів проекту, що стосується їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання «____» _____ 20 ___ р.

Завдання прийняв до виконання _____ Керівник _____

(підпис) (підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Термін виконання	Примітка
1	*Аналіз проблеми предметної області, постановка й формування завдань дослідження*		
2	*Аналіз об'єкту дослідження*		
3	*Розробка інформаційного та програмного забезпечення для створення заготовок на заняття*		
4	*Аналіз отриманих результатів*		
5	*Оформлення пояснювальної записки до кваліфікаційної роботи*		

Здобувач вищої освіти _____ Керівник _____

(підпис) (підпис)

АНОТАЦІЯ

Записка: 41 с., 11 рис., 2 табл., 1 додаток, 14 джерел.

Обґрунтування актуальності теми роботи – автоматизація проведення занять по розробці ігор на Unity, необхідно створити заготовки для проведення тематичних занять, підкреслення розроблюємих механік гри, залучення студентів до роботи і обгрунтування важливості тієї чи іншої механіки

Об’єкт дослідження — навчальний інформаційний процес отримання навичок роботи з 3d Unity

Мета роботи — створення інформаційної технології автоматизованого навчання, яке більш доречно використовувати в асинхронному форматі

Методи дослідження — методи побудови інформаційних систем.

Результати — було спроектовано та реалізовано інформаційну систему автоматизованого навчання групи людей програмуванню на C# та роботі з об'єктами Unity, створено методичні матеріали в PDF і заготовки до кожної пари в Unity, розпланованo теми та матеріал по принципу 80/20 (20% теорія, 80% практика)

UNITY, ІНФОРМАЦІЙНА СИСТЕМА, МЕТОДИЧНІ МАТЕРІАЛИ, 3D, C#, ПРАКТИЧНІ НАВИЧКИ, ДОКУМЕНТАЦІЯ, НАДБУДОВА, ФІЗИКА, ГРА, ДОДАТОК.

Зміст

Вступ 1
1 Аналітичний огляд 3
1.1 Про проект 3
1.2 Основні особливості 3
2 Аналіз існуючих систем для геймдев навчання 8
2.1. Переваги та недоліки існуючих рішень 8
2.2. Відсутність аналогічної системи на ринку 9
3 Вибір технологій для розробки інформаційної системи 10
3.1. Вибір мови програмування 10
3.2. Використання бібліотек та надбудов 11
3.3 Unity Engine – як основний інстурмент 11
3.4 Бібліотека InputSystem 13
3.5 Надбудова Cinemachine 14
4 Практична реалізація 16
4.1 Структура навчальної інформаційної системи отримання практичних навичок роботи з 3d Unity 16
4.2. Організація роботи з рушієм 16
4.3. Інтеграція зовнішніх сервісів 16
4.6. Реалізація методичних матеріалів 23
4.7. Взаємодія з користувачем через гугл клас та інтерактивні завдання 23
Висновок 26
Список літератури 27
Додаток А. Приклад сценарію методичних матеріалів 29

Вступ

У сучасному світі технології розвиваються стрімкими темпами, і все більше сфер діяльності стають невід'ємно пов'язаними з цифровими інноваціями. Однією з таких технологій, яка набуває дедалі більшого значення, є 3D візуалізація, зокрема за допомогою потужного і гнучкого інструменту Unity. Unity, як платформа для розробки інтерактивного 3D контенту, завоювала визнання у сфері створення відеоігор, архітектурних візуалізацій, анімацій, симуляцій і навіть у медичній галузі. Враховуючи цей феноменальний потенціал, навчання роботи з Unity стає ключовим елементом підготовки майбутніх фахівців у різних галузях.

З огляду на сучасні освітні тенденції, традиційні методи навчання, зосереджені на теоретичному викладанні, поступово поступаються місцем інтерактивним, практично орієнтованим підходам. Це особливо актуально для опанування технологій, де необхідно не лише засвоїти теоретичні знання, але й розвинути практичні навички, здатність творчо та ефективно застосовувати їх на практиці. Саме в цьому контексті виникає потреба у створенні навчальної інформаційної системи, що забезпечить ефективне отримання практичних навичок роботи з 3D Unity.

Розробка такої системи передбачає інтеграцію теоретичних матеріалів з практичними завданнями, що дозволить студентам не лише вивчати концепції та принципи 3D моделювання та програмування, але й застосовувати ці знання в реальних проектах. Важливою складовою навчальної системи є її адаптивність та інтерактивність, що забезпечують індивідуальний підхід до кожного студента, враховуючи його початковий рівень знань і темп засвоєння матеріалу.

Крім того, використання таких передових інструментів, як Unity, сприяє розвитку критичного мислення, креативності та навичок вирішення проблем. В процесі навчання студенти вчаться працювати в команді, що є незамінним у

сучасному робочому середовищі, де більшість проектів розробляються колективно.

Таким чином, навчальна інформаційна система, орієнтована на практичні аспекти роботи з 3D Unity, є не просто інструментом для передачі знань, але й платформою для формування висококваліфікованих, творчих і технічно підготовлених спеціалістів, здатних відповідати викликам сучасного технологічного світу. Впровадження такої системи в навчальний процес дозволить значно підвищити рівень підготовки студентів, зробивши їх конкурентоспроможними на ринку праці та готовими до інновацій у своїй професійній діяльності.

1 Аналітичний огляд

1.1 Про проект

Інформаційна система розроблена для формування практичних навичок роботи з платформою Unity для розробки ігор та інтерактивних додатків. Ця система надає користувачам можливість вивчати та вдосконалювати навички роботи в Unity шляхом виконання практичних завдань, проходження курсів та виконання проектів.

1.2 Основні особливості

До основних особливостей належать:

- Інтерактивні курси: пропонують широкий вибір інтерактивних курсів з різних аспектів розробки в Unity, включаючи програмування, створення графіки, анімації та інші.
- Практичні завдання: Система надає користувачам доступ до практичних завдань, які допомагають закріплювати отримані знання та навички шляхом їх застосування на практиці.
- Проекти: дозволяють користувачам створювати власні проекти в Unity, отримуючи можливість застосувати знання та навички, отримані під час навчання, у реальних проектах.

1.3 Постановка задачі

Метою роботи є створення інформаційної системи для автоматизованого навчання групи людей роботі з Unity та C#, розробка методичних матеріалів з покроковим мануалом створення власної 3д гри з урахуванням різного рівня навичок та досвіду

Для досягнення поставленої мети необхідно вирішити наступні задачі:

Таблиця 1. - Основні функціональні та нефункціональні вимоги

R_ID	Вимога
R1	***Функціональні вимоги***
R1.1	Матеріали мають бути по принципу 20% теорії і 80% практики для кращого засвоєння інформації.
R1.2	Матеріали мають бути адаптовані під викладача та студента, з метою покращити можливість їх самостійного опрацювання.
R1.3	Мають бути пояснення до кожного класу, функції, бібліотеки, додаткових модулів.
R1.4	Методичні матеріали повинні бути зрозумілими для цільової аудиторії. Інструкції та пояснення мають бути лаконічними та доступними.
R1.5	Матеріали повинні включати всю необхідну інформацію, яка допоможе користувачам досягти поставленої мети.
R1.6	Інформація в методичних матеріалах повинна бути організована в логічній послідовності, щоб користувачам було легко слідувати за ними.
R1.7	Матеріали мають відповідати конкретним цілям і завданням, які вони призначені підтримувати.
R1.8	Мають бути прикріплені скріншоти та посилання на додатковий відеоматеріал для візуалізації конкретної роботи і коректного розуміння задачі користувачами.
R1.9	У разі необхідності, методичні матеріали можуть містити елементи інтерактивності, такі як вправи, тести тощо.

R2	***Нефункціональні вимоги***
R2.1	Матеріали мають бути у форматі .pdf для надання змоги відкриття з будь якого пристрою
R2.2	Продукт, що розробляється має коректно працювати на будь якій версії Unity
R2.3	Система має бути локалізованою. Має бути створено українську та англійську версію кожного файлу
R2.4	Методичні матеріали повинні мати привабливий дизайн і оформлення, що полегшує сприйняття інформації.
R2.5	Методичні матеріали повинні бути перевірені на відсутність помилок і недоліків, які можуть призвести до неправильного розуміння або виконання інструкцій.
R2.6	Матеріали повинні завантажуватися та відображатися швидко, а також працювати ефективно під час використання.
R2.7	Методичні матеріали повинні бути придатні для використання в різних масштабах - від індивідуального використання до групового або масового.
R2.8	Матеріали мають бути доступні 24/7 для користувачів з різних регіонів
R2.9	При виникненні помилок або випадків втрати даних, методичні матеріали повинні бути легко відновлюваними без значного затримання користувачів.
R2.10	Методичні матеріали повинні відповідати вимогам щодо збереження конфіденційності, безпеки та інших правил, що регулюють їхнє використання.

R2.11	Методичні матеріали повинні бути адаптовані до потреб користувачів різного віку, рівня знань та досвіду.

Таблиця 2. - Основні питання

R_ID	Question	Outcome
R1.1; R1.2; R1.3; R1.4; R1.5; R1.6; R1.7; R2.5; R2.7;	Відображення повної інформації про кожну дію	Кожна дія має бути розписана з чіткими вказівками роботи та поясненням в повній мірі важливості та функціонального сенсу кожної з них
R1.8; R1.9; R2.5;	Повинні бути прикріплені посилання на додаткові матеріали а також скриншоти до роботи	Посилання мають бути робочі, скриншоти мають бути корректними
R2.1; R2.6; R2.8; R2.9;	Відкриття незалежно від пристрою та ПЗ	Матеріали мають бути в форматі .pdf
R2.2	Умови працездатності матеріалів	Матеріали мають працювати з усіма актуальними версіями Unity
R2.3	Локалізація	Матеріали мають бути локалізовані під українську та англійську мову

R2.4; **R2.10;** **R2.11;**	Адаптація, безпека та дизайн	Повинен бути привабливий дизайн який допоможе користувачам різного віку краще розуміти інформацію, а також вони мають бути збережені всі авторські права

2 Аналіз існуючих систем для геймдев навчання

2.1. Переваги та недоліки існуючих рішень

Під час аналізу існуючих систем для навчання роботи в Unity було виявлено декілька переваг та недоліків. Деякі з переваг таких систем включають доступність з будь-якого пристрою з підключенням до Інтернету, інтерактивний інтерфейс, можливість навчатися у власному темпі. Однак, існуючі системи мають деякі недоліки, такі як обмежений функціонал, недостатній рівень взаємодії з користувачем, а також обмеження у виборі інструментів для навчання.

До основних переваг належать:

● Доступність: Можливість використання з будь-якого пристрою з Інтернетом, що робить навчання доступним і зручним.

● Інтерактивність: Інтерактивний інтерфейс, що сприяє залученню користувача та покращує засвоєння матеріалу.

● Темп навчання: Можливість навчатися у власному темпі, встановлюючи власний розклад та обсяг занять.

До основних недоліків належать:

● Обмежений функціонал: Деякі існуючі додатки мають обмежений функціонал, що може обмежувати можливості користувачів.

● Недостатній рівень взаємодії: Деякі додатки можуть мати недостатній рівень взаємодії з користувачем, що може впливати на зручність використання.

● Обмеження вибору інструментів: Можливі обмеження у виборі інструментів для навчання, що не дозволяє користувачам розширювати свої навички в широкому спектрі роботи з Unity.

2.2. Відсутність аналогічної системи на ринку

Під час аналізу популярності та пошукових запитів за допомогою Google Trends було виявлено, що існує велика відсутність запитів про інформаційну систему для формування практичних навичок роботи в Unity. Запити за рік не перевищували 100, що свідчить про низький рівень інтересу до цього напрямку. Дані пошуку веб-додатка також показали відсутність запитів на ринку.

Це відкриває можливості для успішного впровадження та розвитку інформаційної системи для навчання роботи в Unity. Відсутність аналогічної системи у конкурентів також може допомогти залучити увагу користувачів та стати лідером у даній області.

Ці аспекти дозволяють побачити переваги та можливості системи на тлі існуючих рішень і показують потенціал для успішного впровадження та розвитку у цій сфері.

3 Вибір технологій для розробки інформаційної системи

3.1. Вибір мови програмування

Unity використовує мову програмування C# (C Sharp), тоді як Java є мовою програмування, яка часто використовується для розробки Android-додатків та веб-додатків.

Обидві мови, C# і Java, мають свої переваги та недоліки для розробки у Unity, але насправді вибір між ними залежить від наших особистих вподобань, досвіду програмування та конкретних потреб нашого проекту.

Підтримка Unity: Оскільки Unity найбільш широко використовується мову програмування C#, багато матеріалів, уроків та документації доступні саме для нього. Це робить C# більш придатним для початківців або тих, хто швидко хоче навчитися розробці у Unity.

Продуктивність: C# може бути більш продуктивним для розробки у Unity, оскільки має ряд функцій, що полегшують створення складних ігор та додатків. Крім того, він має більш розвинену систему об'єктно-орієнтованого програмування, яка дозволяє краще організувати код.

Швидкодія: Швидкодія може бути важливою для деяких проектів. Хоча Java швидше за C# в деяких випадках, у контексті роботи з Unity різниця може бути незначною, оскільки обидві мови інтерпретуються або компілюються в інтерпретований код.

Екосистема: Вирішивши використовувати Java у Unity, нам може бути складніше знайти відповідні ресурси, бібліотеки або підтримку, оскільки C# є основною мовою програмування для цієї платформи.

В цілому, для новачків у розробці у Unity або для швидкого виходу на ринок з проектом, то C# може бути кращим вибором. Саме на цій мові буде створена вся екосистема ігор по методичним матеріалам в нашій системі.

3.2. Використання бібліотек та надбудов

Для реалізації взаємодії з користувачем використовуються наступні бібліотеки та надбудови:

UnityEngine[6]: Бібліотека UnityEngine використовується для керування вбудованими інструментами Unity, об’єктами, класами та їх методами.

Cinemachine: Надбудова для створення динамічної камери для слідкування за ігровими об’єктами.

Input system: Надбудова для створення кроссплатформового керування ігровими об’єктами.

Вибір цих бібліотек та надбудов дозволяє нам ефективно реалізувати поставлені завдання та надати користувачам високоякісний та захоплюючий досвід користування нашою інформаційною системою.

3.3 Unity Engine – як основний інстурмент

Unity Engine[6] - це потужний інтегрований набір інструментів для розробки ігор та інтерактивних додатків. Вона поєднує в собі інноваційні технології, широкий функціонал та простоту використання, що робить її однією з найпопулярніших платформ для створення візуально захоплюючих ігрових досвідів.

Основні складові бібліотеки Unity Engine включають:

Графічний движок: Unity використовує потужний графічний движок, який дозволяє створювати реалістичні та естетично привабливі візуальні ефекти. Цей движок підтримує високоякісне освітлення, тіні, текстури, анімацію та інші графічні ефекти.

Фізичний движок: Unity має вбудований фізичний движок, який дозволяє моделювати реалістичну фізику в іграх. Він підтримує різні типи тіл, колізії, сили тяжіння та інші фізичні властивості, що додають реалізму ігровим сценам.

Аудіо-система: Unity надає розширену аудіо-систему, яка дозволяє легко інтегрувати звуки, музику та інші аудіо-ефекти у ваші проекти. Вона підтримує різні формати аудіофайлів та дозволяє налаштовувати їх відтворення за допомогою скриптів.

Анімація: Unity надає широкі можливості для створення анімації об'єктів та персонажів. Вона підтримує анімаційні куратори, анімаційні кліпи, скелетну анімацію та інші методи створення анімації.

Інструменти розробки: Unity має інтегроване середовище розробки (IDE), яке включає в себе редактор сцен, графічний редактор, консоль розробника та інші інструменти для створення та налагодження ігор та додатків(див. рис. 3.1)

Система управління ресурсами: Unity має ефективну систему управління ресурсами, яка дозволяє ефективно керувати активами, текстурами, звуками та іншими ресурсами вашого проекту.

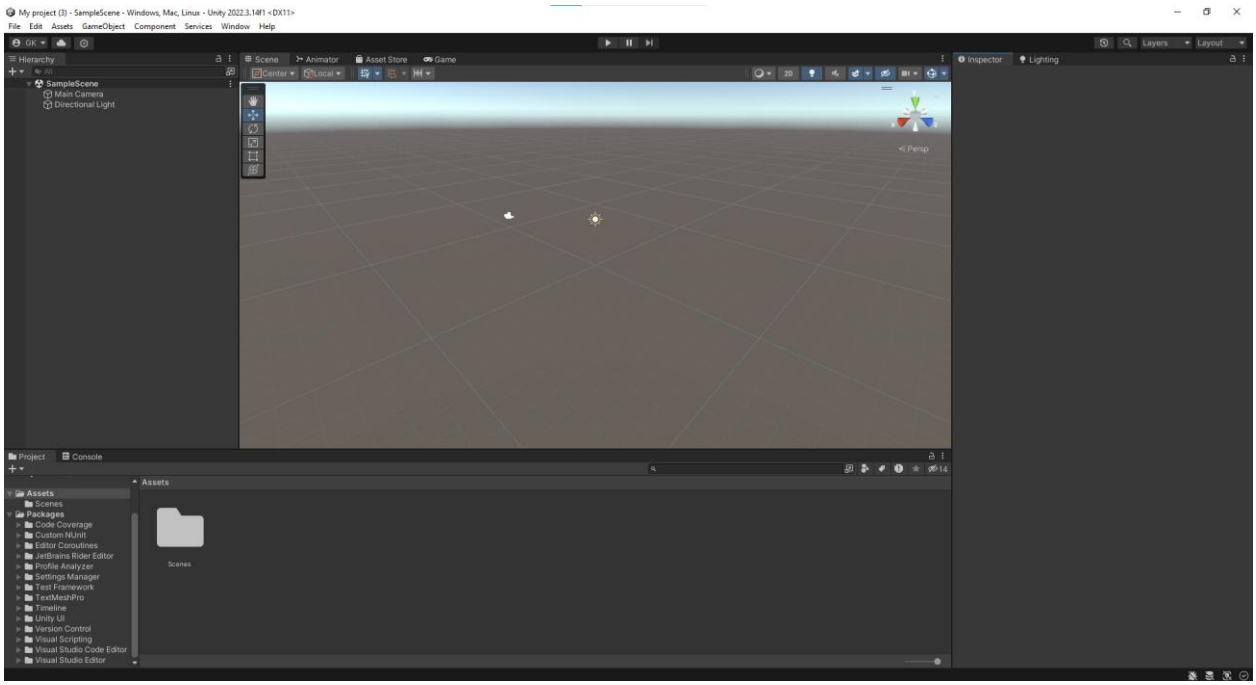


Рисунок 3.1 – Інтерфейс Unity

3.4 Бібліотека InputSystem

"InputSystem[12]" - це надбудова в Unity, яка надає потужні та розширені можливості для обробки введення від різних джерел, таких як клавіатура, миша, геймпади, сенсори та інші пристрої введення. Ця система є заміною старого Input Manager і була розроблена з метою поліпшення продуктивності, гнучкості та розширюваності обробки введення в Unity.

Основні переваги "InputSystem[12]" включають:

● Гнучкість: В InputSystem[12] існує можливість динамічно налаштовувати обробку введення за допомогою скриптів. Це дозволяє створювати різні сценарії обробки введення для різних пристроїв або ситуацій.

● Розширюваність: InputSystem[12] надає механізми розширення для підтримки нових типів пристроїв введення та додаткових функцій обробки введення.

● Універсальність: В InputSystem[12] інтегровано підтримку різних типів пристроїв введення, включаючи клавіатуру, мишу, геймпади, тачпади, сенсорні екрани та інші пристрої.

● Швидкодія: "InputSystem[12]" працює оптимально та ефективно, що дозволяє обробляти великий обсяг введення без втрат продуктивності.

● Мультиплатформеність: Ця надбудова підтримує різні платформи, включаючи ПК, консолі, мобільні пристрої та інші, що робить її ідеальним рішенням для розробки багатьох видів ігор та додатків.

● Сучасний API: InputSystem[12] пропонує сучасний API з можливостями асинхронної обробки введення та інших продуктивних функцій, що полегшують розробку складних інтерактивних додатків.

Загалом, "InputSystem[12]" є потужним інструментом для обробки введення в Unity, який дозволяє розробникам створювати більш гнучкі, ефективні та масштабовані ігри та додатки(див. рис. 3.2)

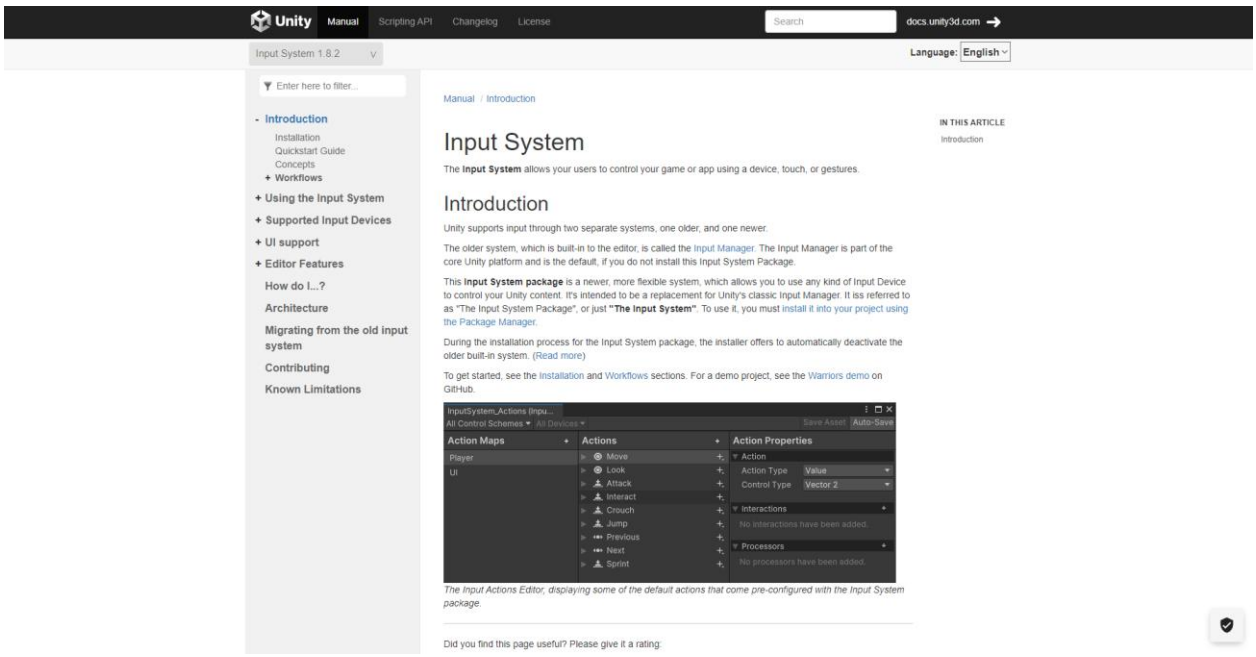


Рисунок 3.2 – Документація до бібліотеки InputSystem

3.5 Надбудова Cinemachine

"Cinemachine[11]" - це надбудова в Unity, яка призначена для створення кінематографічних ефектів та камерних рухів у проектах. Вона дозволяє розробникам створювати вражаючі та професійні кіносценарії, керуючи камерами, їхніми рухами, композицією та іншими аспектами зображення.

Основні функції та переваги Cinemachine[1] включають:

● Автоматичне вирішення задач камери: Cinemachine[11] автоматично робить багато рутинних задач, таких як слідування за об'єктами, стабілізація камери, зони колізії та інші, що дозволяє розробникам швидко створювати камерні ефекти без необхідності великої кількості програмування.

● Гнучкість налаштувань: Cinemachine[11] має велику кількість параметрів налаштувань, які дозволяють контролювати кожний аспект камерного руху та композиції, від розміщення та орієнтації до швидкості та плавності.

● Підтримка скриптів: Cinemachine[11] інтегровано з Unity та підтримує використання скриптів для програмного керування камерами та їхніми

параметрами. Це дозволяє розробникам створювати складні та динамічні камерні ефекти за допомогою коду.[8]

● Плавні переходи: Cinemachine надає можливість створювати плавні переходи між різними камерними позиціями та налаштуваннями, що дозволяє створювати більш природні та зручні камерні рухи.

● Підтримка VR: Cinemachine також підтримує використання віртуальної реальності (VR), дозволяючи розробникам створювати кінематографічні ефекти та камерні рухи для VR-проектів.[8]

Усі ці функції роблять Cinemachine потужним інструментом для створення кінематографічних ефектів та камерних рухів у проекті Unity. Вона дозволяє вам швидко та легко створювати вражаючі камерні сцени, які покращують візуальний досвід гравців та роблять гру або додаток більш привабливими(див. рис. 3.3)

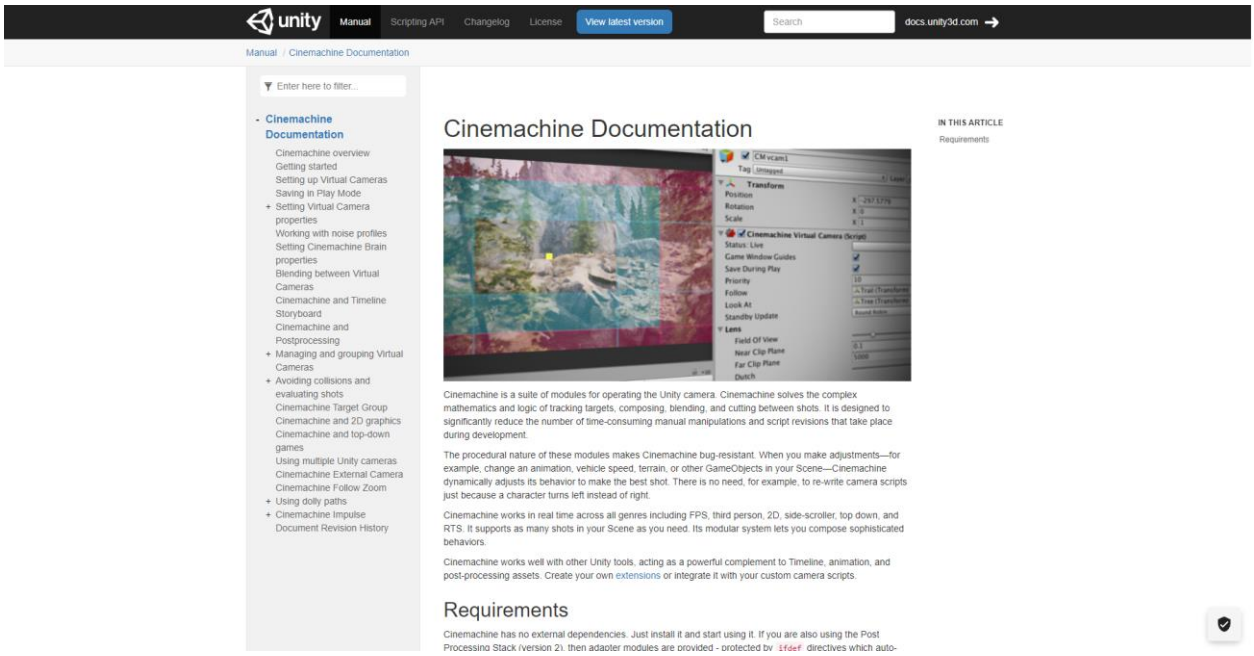


Рисунок 3.3 – Документація до надбудови Cinemachine

4 Практична реалізація

4.1 Структура навчальної інформаційної системи отримання практичних навичок роботи з 3d Unity

Ігри - заготовки будуть побудовані на наступних базових компонентах[3], які забезпечать їх функціональність та ефективну роботу:

- Інтерфейс користувача: Основний інтерфейс, який надає зручний доступ до функцій додатка та взаємодію з користувачем. Робота з UI у іграх та додатках для відображення інформації про стан персонажа, або елементи контролю для мобільних пристроїв.[10]

- Система керування: Будь яка система керування створена за допомогою Input System для надання змоги керувати персонажем як з клавіатури так і з геймпада.[7]

- Аніматор та його функції: Жодна гра не обходиться без анімацій, за допомогою вбудованої системи редактора анімацій надаємо нашому персонажу різні дерева анімацій і за потреби коректуємо їх.[7]

4.2. Організація роботи з рушієм

Для реалізації роботи з матеріалами для вивчення базовою потребою буде встановлення Unity разом з білдом під андроід, та створення акаунту в Unity для коректної роботи з Asset store[9]

4.3. Інтеграція зовнішніх сервісів

Система використовуватиме зовнішні сервіси для поліпшення роботи та додавання нових механік. Приклади сервісів: Mixamo[1], Asset store[2], Photon.(див. рис. 4.1-4.3)

Рисунок 4.1 - Photon - онлайн сервіс для створення і підключення онлайн серверу до гри

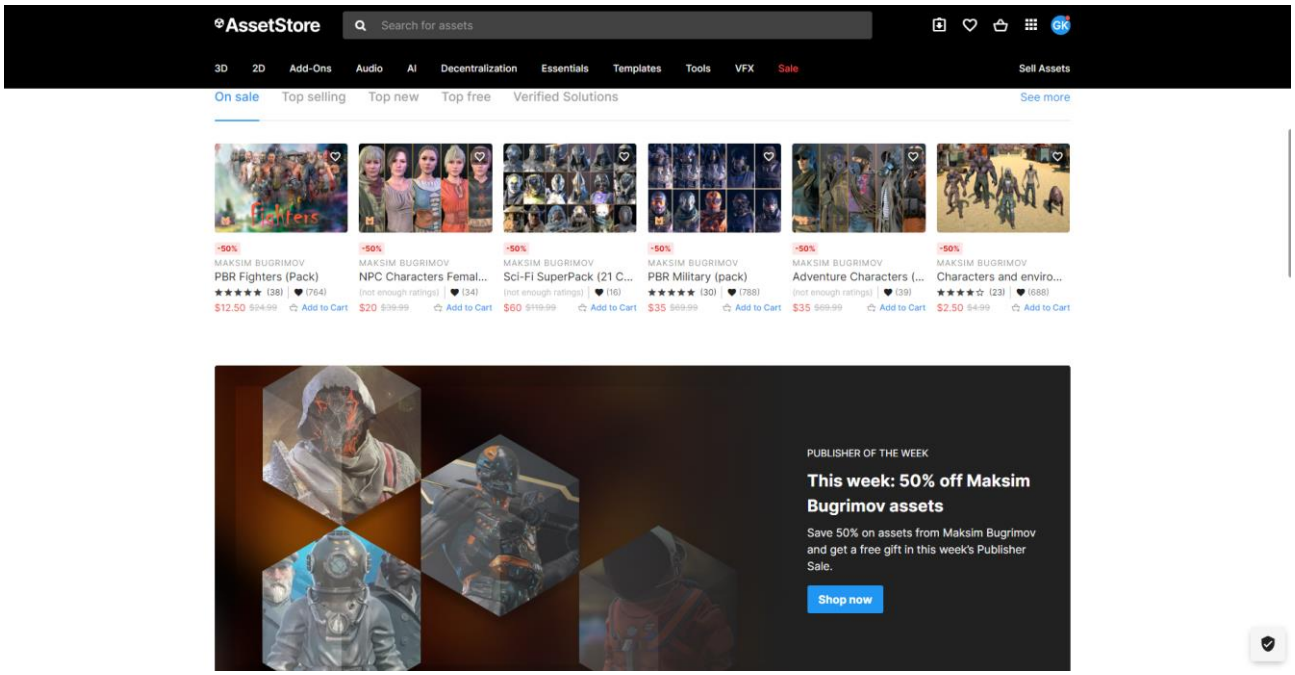


Рисунок 4.2 - AssetStore - онлайн сервіс для отримання асетів для ігор

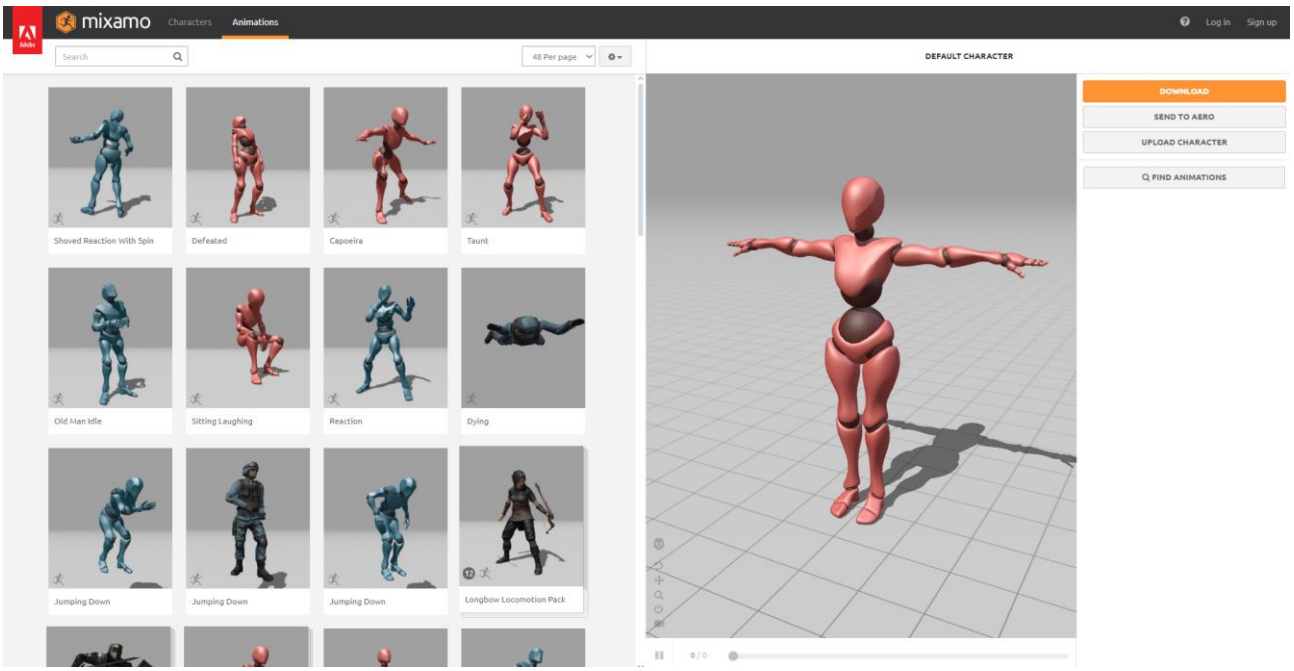


Рисунок 4.3 - Mixamo - онлайн сервіс для отримання персонажів та анімацій

4.4. Розробка методичних матеріалів та заготовок

Основною метою розробки методичних матеріалів для навчальної інформаційної системи є забезпечення ефективного та інтегрованого процесу навчання студентів роботи з 3D Unity. Методичні матеріали мають допомогти студентам зрозуміти основні принципи 3D моделювання та програмування, розвинути практичні навички створення інтерактивних 3D додатків, а також сприяти творчому мисленню та вирішенню практичних задач.

Завдання методичних матеріалів включають:

- Створення систематизованого та послідовного навчального контенту.
- Розробку практичних завдань і проектів, що відповідають різним рівням підготовки студентів.
- Забезпечення доступу до навчальних матеріалів через інтерактивну платформу.
- Підтримку самостійного навчання та оцінювання знань студентів.

Методичні матеріали поділяються на декілька основних компонентів:

1. Теоретичні модулі:
   - Вступ до Unity: Огляд платформи, інтерфейс користувача, основні функції.
   - Основи 3D моделювання: Геометрія, матеріали, текстури.
   - Програмування в Unity: Основи C#, сценарії, обробка подій.
   - Фізика і анімація в Unity: Динаміка, колізії, створення анімацій.
   - Розробка ігор та інтерактивних додатків: Проектний підхід, інтеграція звуку, створення інтерфейсу.
2. Практичні завдання:
   - Лабораторні роботи: Серія вправ для засвоєння основних інструментів та функцій Unity.

- Міні-проекти: Невеликі проекти для закріплення практичних навичок, наприклад, створення простої гри або 3D сцени.[13](див. рис. 4.4)
- Курсові роботи: Більш складні та тривалі проекти, що потребують застосування усіх набутих знань.

3. Методичні рекомендації для викладачів:

- Плани занять: Детальний розклад занять з рекомендаціями щодо їх проведення.
- Критерії оцінювання: Рубрики для оцінювання практичних завдань і проектів.
- Ресурси для додаткового навчання: Література, онлайн-курси, спільноти розробників

4. Навчальні матеріали для самостійного вивчення:

- Відеоуроки: Послідовні відеоуроки з поясненням теоретичних аспектів і демонстрацією практичних вправ.
- Посібники і підручники: Електронні книги та статті, що покривають всі теми курсу.
- Форуми і чати для студентів: Платформи для обміну досвідом та взаємодопомоги між студентами.

Розробка теоретичного контенту:

- Теоретичні модулі повинні бути розроблені на основі сучасних навчальних програм та відповідати останнім версіям Unity.
- Включення прикладів з реального життя для кращого розуміння матеріалу.

Підготовка практичних завдань:

- Завдання мають бути різнорівневими, від базових до більш складних, щоб забезпечити поступове зростання навичок.

- Кожне завдання повинно мати детальний опис, інструкції по виконанню та очікувані результати(див. рис 4.5)

Розробка методичних рекомендацій для викладачів:

- Викладачам потрібно надати чіткі інструкції щодо проведення занять та оцінювання робіт студентів.
- Рекомендації повинні включати поради щодо вирішення типових проблем, з якими можуть стикатися студенти.

Створення мультимедійного контенту:

- Запис відеоуроків та створення інтерактивних презентацій для пояснення складних концепцій.
- Використання графіки та анімації для візуалізації процесів.

Інтеграція навчальних матеріалів у платформу:

- Навчальна інформаційна система повинна забезпечити зручний доступ до всіх матеріалів.
- Розробка інтерфейсу користувача, що дозволить легко орієнтуватись в контенті.

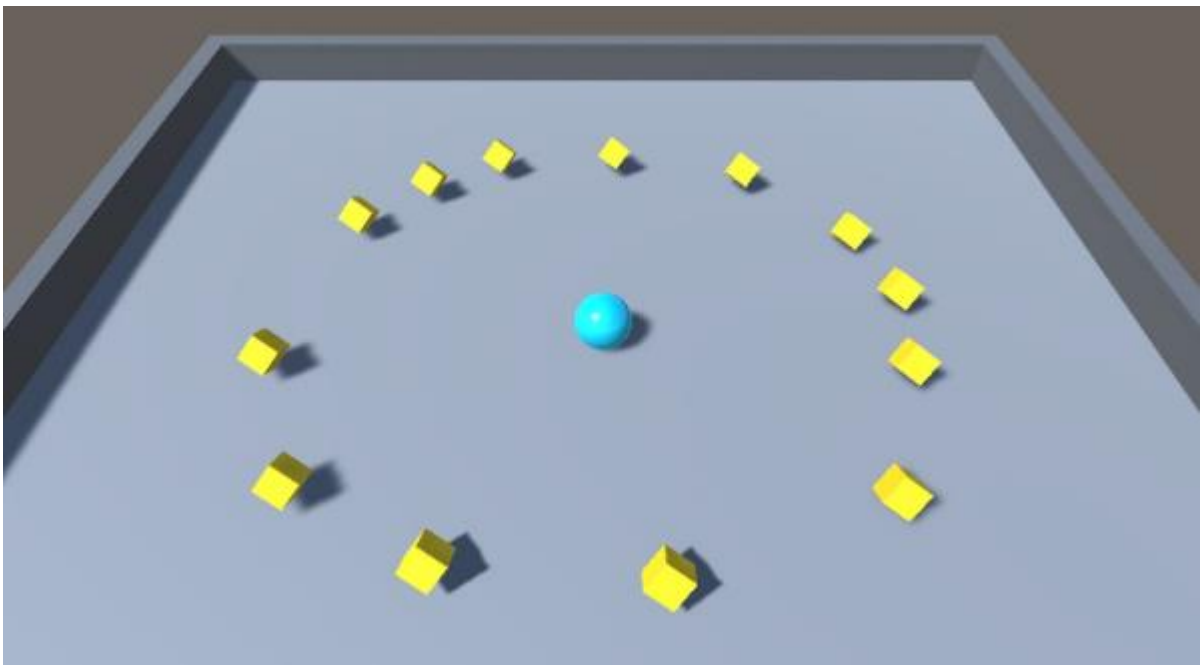


Рисунок 4.4 - Міні-гра з демонстрацією можливостей рушія

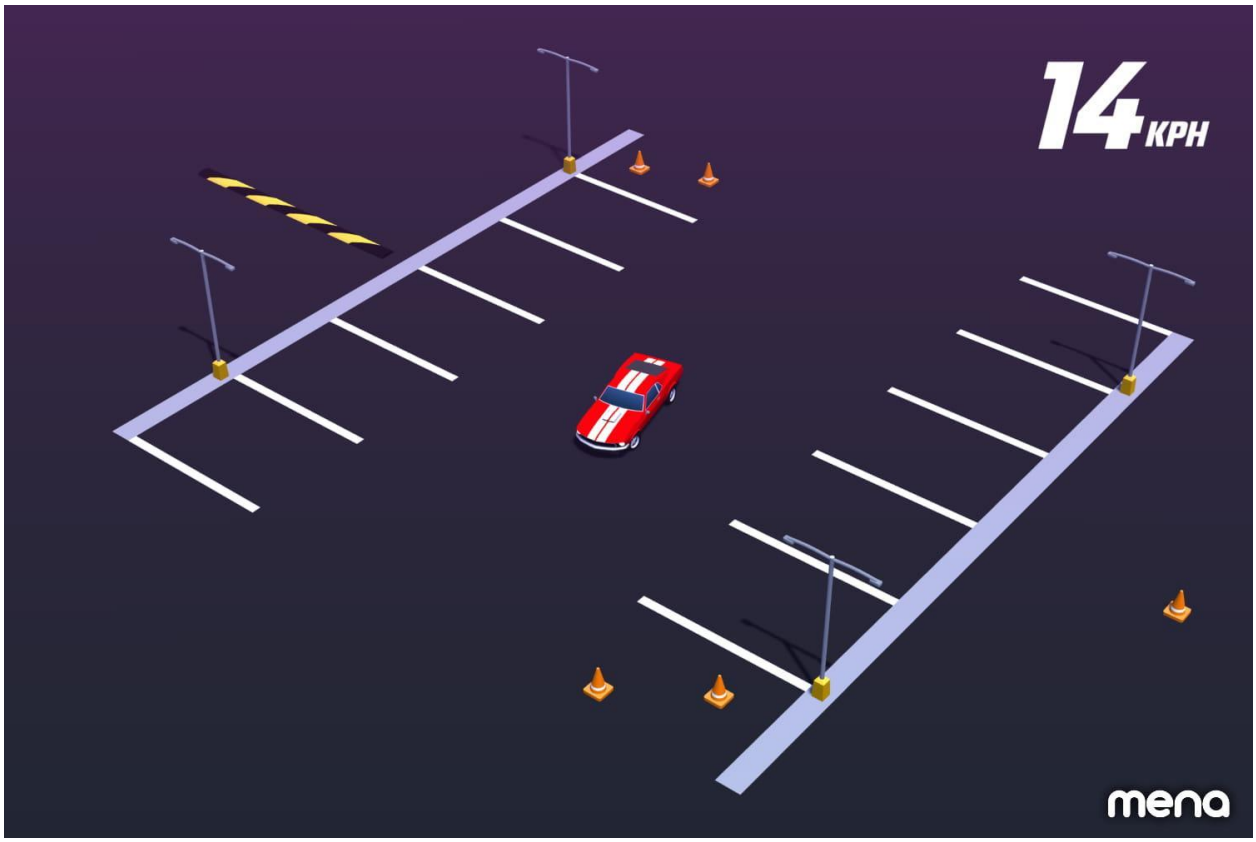


Рисунок 4.5 - Один з результатів заняття

4.5. Тестування та вдосконалення методичних матеріалів

Пілотне тестування:

- Вибір групи студентів для тестування навчальних матеріалів.
- Збір зворотного зв'язку від студентів та викладачів для оцінки ефективності.

Аналіз результатів:

- Аналіз успішності студентів, виявлення слабких місць у матеріалах.
- Коригування і доповнення матеріалів на основі зібраного зворотного зв'язку.

Постійне оновлення:

- Регулярне оновлення матеріалів відповідно до нових версій Unity та змін у галузі.
- Врахування нових технологій і методик навчання для покращення системи.

4.6. Реалізація методичних матеріалів

Будуть розроблені методичні матеріали в форматі .pdf з покроковою інструкцією та планом дій, також створені заготовки з готовими матеріалами до кожного уроку.[4]

4.7. Взаємодія з користувачем через гугл клас та інтерактивні завдання

Створення гугл класу[5] для надання методичних матеріалів та завдань.

Додавання інтерактивних завдань за допомогою онлайн платформ: learningapps.org[4], app.genial.ly[14](див. рис. 4.6-4.8)

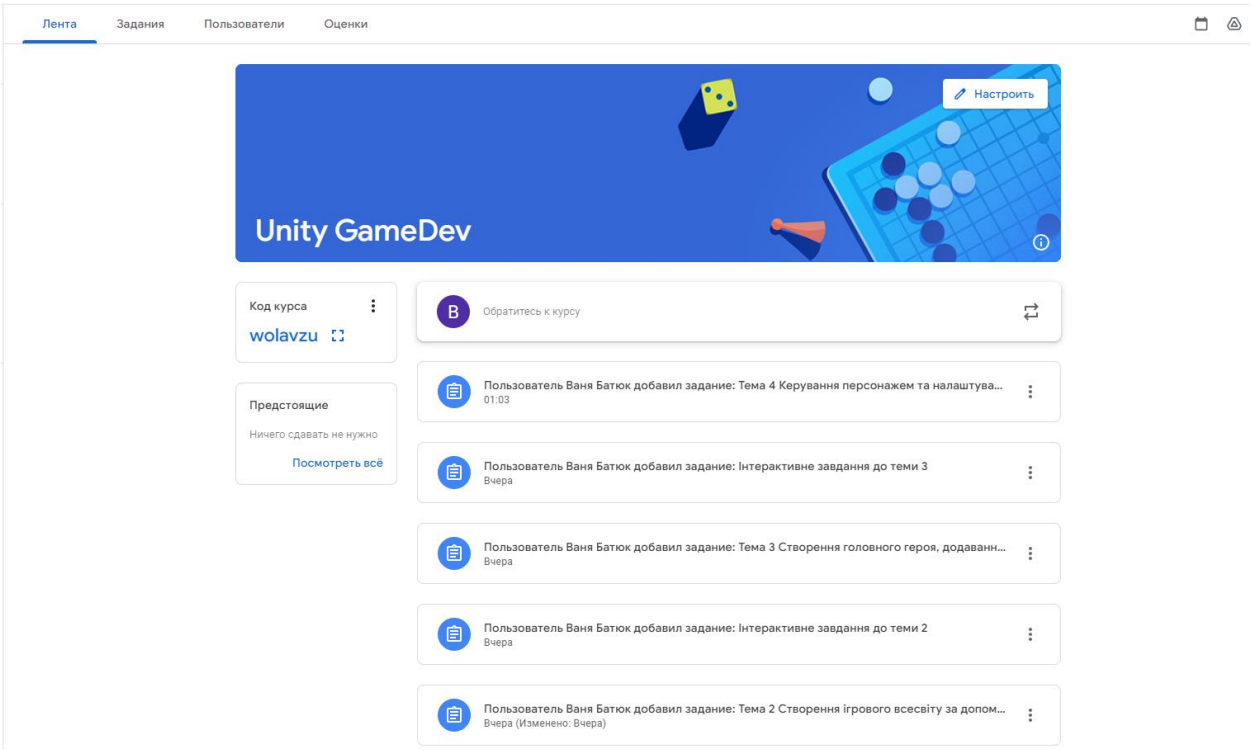


Рисунок 4.6 - Google classroom

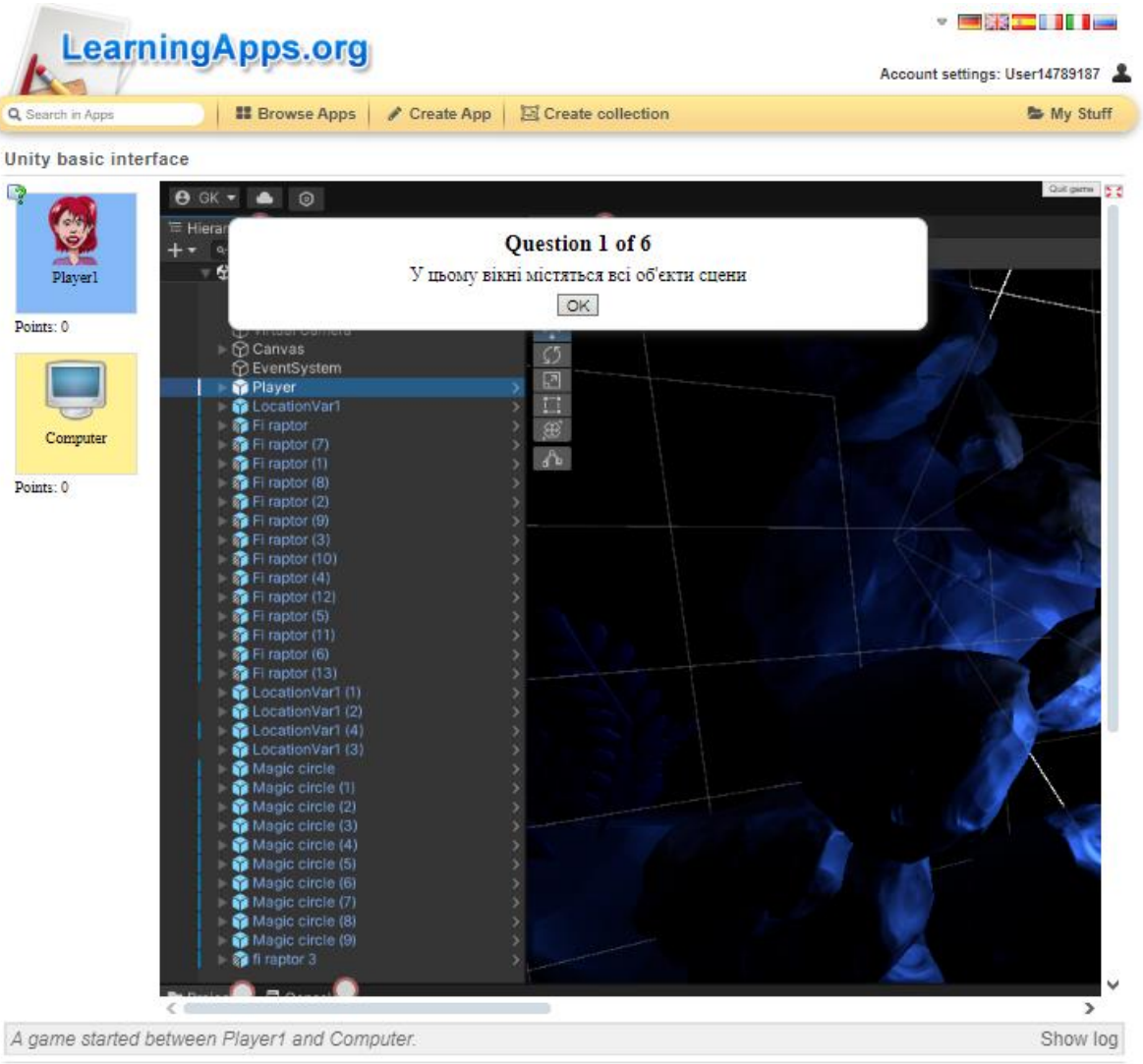


Рисунок 4.7 – Інтерактивне завдання

Рисунок 4.8 - Квіз створений на genially.com

Висновок

У данному проекті великий акцент був зроблений на створенні інтерактивної інформаційної системи для практичного вивчення Unity. Розроблені методичні матеріали, які доповнюються заготовками для занять та інтерактивними завданнями, сприяють ефективному засвоєнню матеріалу. Планується подальше розширення системи за допомогою відеоінструкцій та більш складних завдань, що дозволить студентам вдосконалити свої навички. Після освоєння базових концепцій, учні будуть вивчати більш продуктивні аспекти розробки, такі як мультиплеєр та монетизація проектів, що готуватиме їх до викликів сучасної індустрії розробки програмного забезпечення.

Отже, цей проект має потенціал стати важливим інструментом для навчання та підготовки молодих розробників до успішної кар'єри у сфері розробки ігор та програм. Покладаючи акцент на практичні навички та розвиток креативності, ця інформаційна система створює мостик між теоретичними знаннями та практичною діяльністю. Її модулярна структура дозволяє студентам зосередитися на конкретних аспектах розробки у межах їхніх індивідуальних інтересів та потреб. Розширення функціональності системи забезпечить їм можливість експериментувати з різноманітними ідеями та проектами, що відкриє нові горизонти та стимулюватиме творчість. Отже, ця інформаційна система не лише сприяє засвоєнню ключових принципів розробки в Unity, але й створює умови для творчого росту та самореалізації студентів у світі високотехнологічних інновацій.

Список літератури

1. «mixamo.com» [Електронний ресурс]: [Інтернет-портал]. Отримання 3д моделей та анімацій – Режим доступу: https://www.mixamo.com

2. «AssetStore» [Електронний ресурс]: [Інтернет-портал]. Отримання ассетів для створення ігор – Режим доступу: https://assetstore.unity.com

3. «Unity documentation» [Електронний ресурс]: [Інтернет-портал]. Документація по роботі з компонентами та рушієм Unity – Режим доступу: https://docs.unity3d.com/Manual/index.html

4. «LearningApps» [Електронний ресурс]: [Інтернет-портал]. Створення інтерактивних завдань для опрацювання матеріалу – Режим доступу: https://learningapps.org

5. «Google Classroom» [Електронний ресурс]: [Інтернет-портал]. Створення класу для надання завдань, методичних матеріалів та фідбеку по ним – Режим доступу: https://classroom.google.com

6. «unity.com» [Електронний ресурс]: [Інтернет-портал]. Отримання ігрового рушія – Режим доступу: https://unity.com

7. «Unity blog» [Електронний ресурс]: [Інтернет-портал]. Блог розробників рушія – Режим доступу: https://blog.unity.com

8. «Visual effects» [Електронний ресурс]: [Інтернет-пост]. 12 Recipes for popular visual effects using the Universal Render Pipeline– Режим доступу: https://blog.unity.com/engine-platform/12-recipes-for-popular-visual-effects-using-universal-render-pipeline

9. «Level design» [Електронний ресурс]: [Інтернет-пост]. Our first-ever e-book for level designers is here – Режим доступу: https://blog.unity.com/games/e-book-for-level-designers

10. «UI» [Електронний ресурс]: [Інтернет-пост]. It’s all in here: The ultimate guide to creating UI interfaces in Unity – Режим доступу: https://blog.unity.com/games/ultimate-guide-to-creating-ui-interfaces

11. «Cinemachine» [Електронний ресурс]: [Інтернет-портал]. Документація до надбудови Cinemachine– Режим доступу: https://docs.unity3d.com/Packages/com.unity.cinemachine@2.3/manual/index.html

12. «InputSystem» [Електронний ресурс]: [Інтернет-портал]. Документація до бібліотеки InputSystem– Режим доступу: https://docs.unity3d.com/Packages/com.unity.cinemachine@2.3/manual/index.html

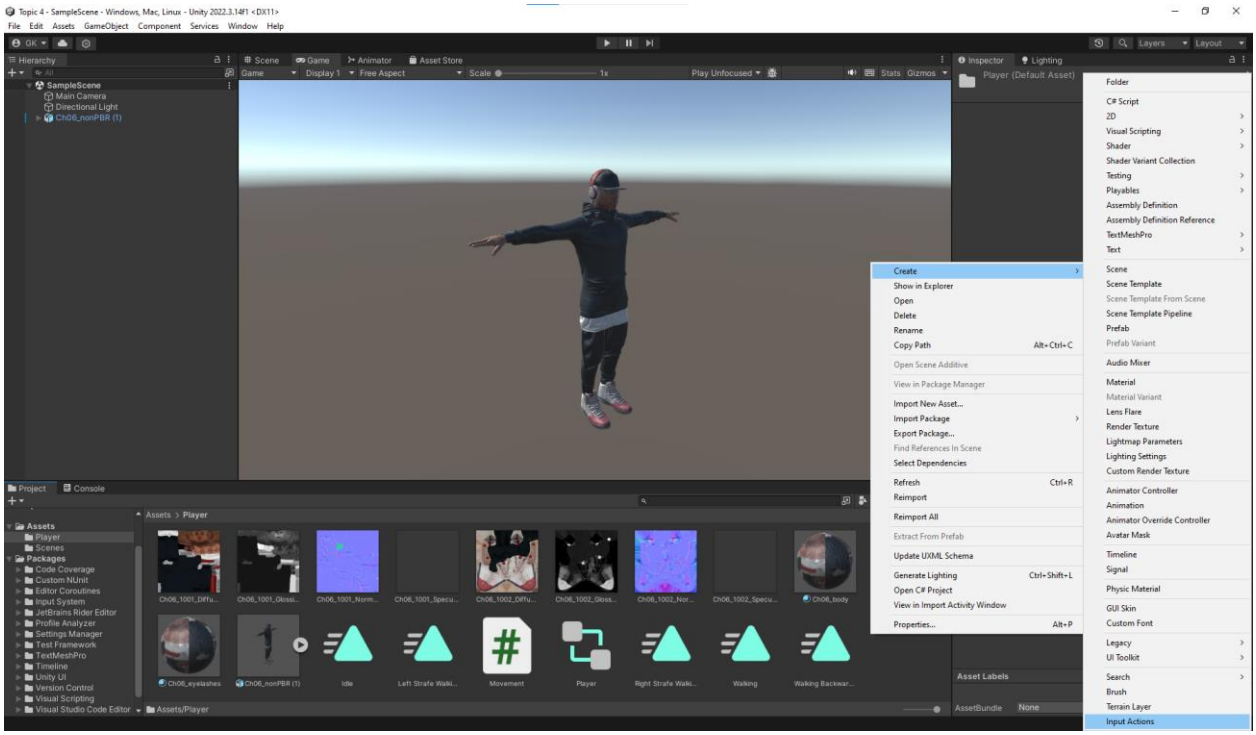
13. «roll a ball» [Електронний ресурс]: [Інтернет-портал]. Туторіал по створенню простих 3Д механік– Режим доступу: https://learn.unity.com/project/roll-a-ball

14. «Genially.com» [Електронний ресурс]: [Інтернет-портал]. Сервіс для створення квізів та карт знань– Режим доступу: https://app.genially.com

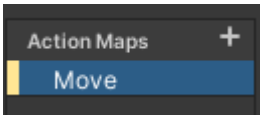
Додаток А. Приклад сценарію методичних матеріалів

Налаштування InputSystem

Після встановлення всіх необхідних надбудов нам необхідно створити керування персонажем. У вікні “Project” натискаємо правою кнопкою миші на пустий простір і створюємо Input actions:

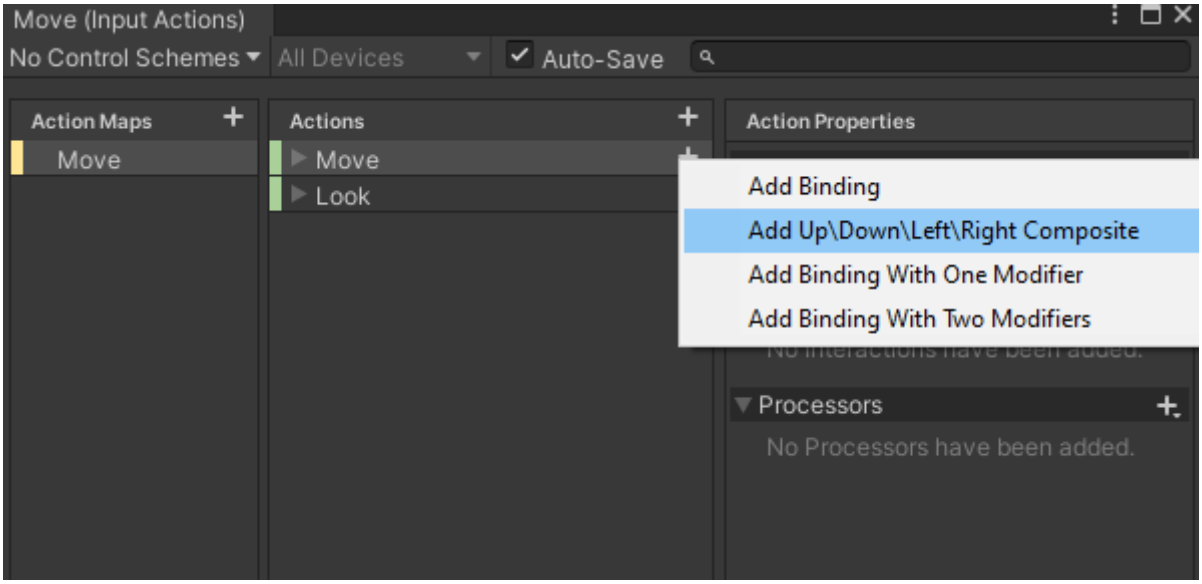


Після чого відкриваємо створений компонент створюємо **Action map** і даємо йому назву **Move**

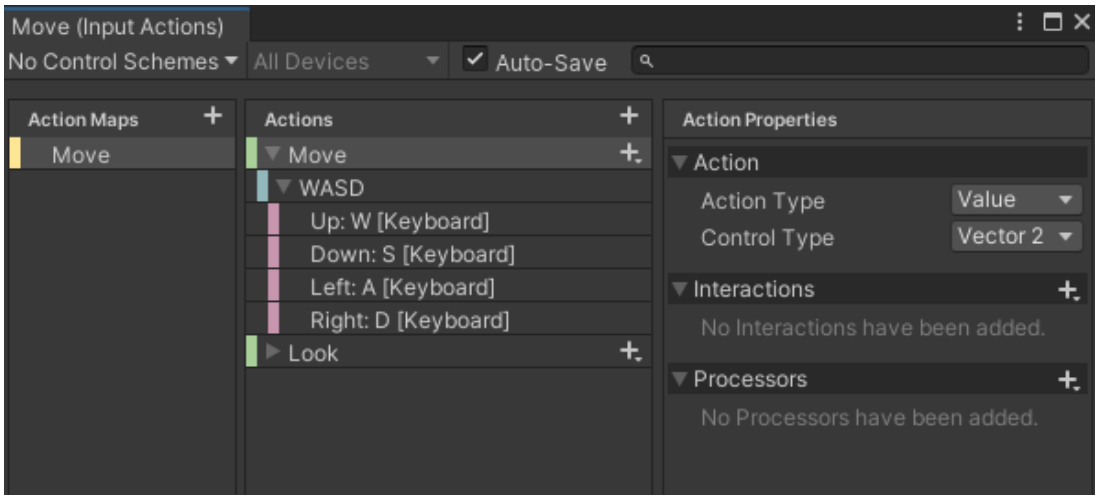


Далі створюємо 2 компонента **Action** і даємо їм відповідну назву, також важливо поставити на обидва компонента параметри як на скріншоті

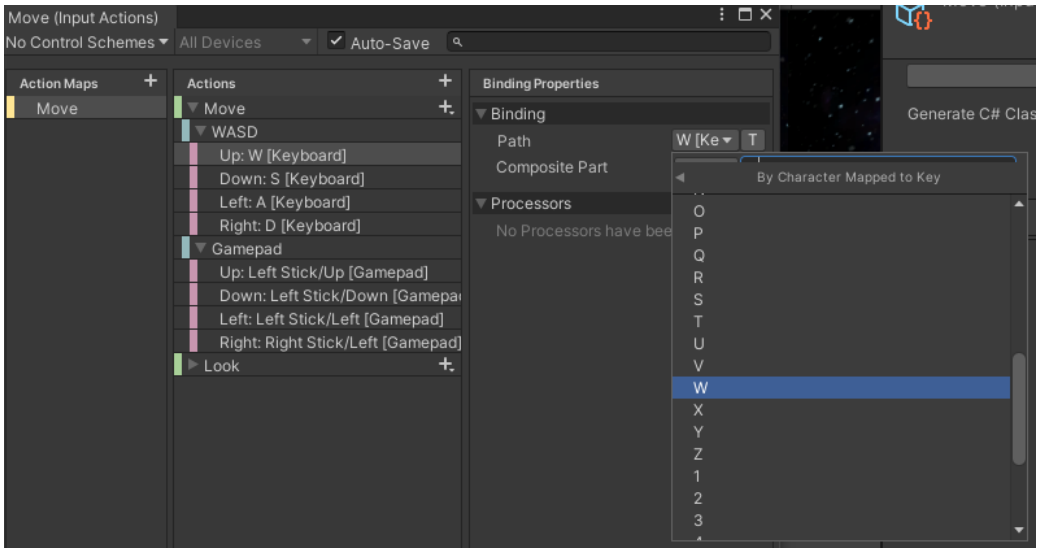
Після виставлення необхідних налаштувань необхідно натиснути на + і створити **Add Up\Down\Left\Right Composite**



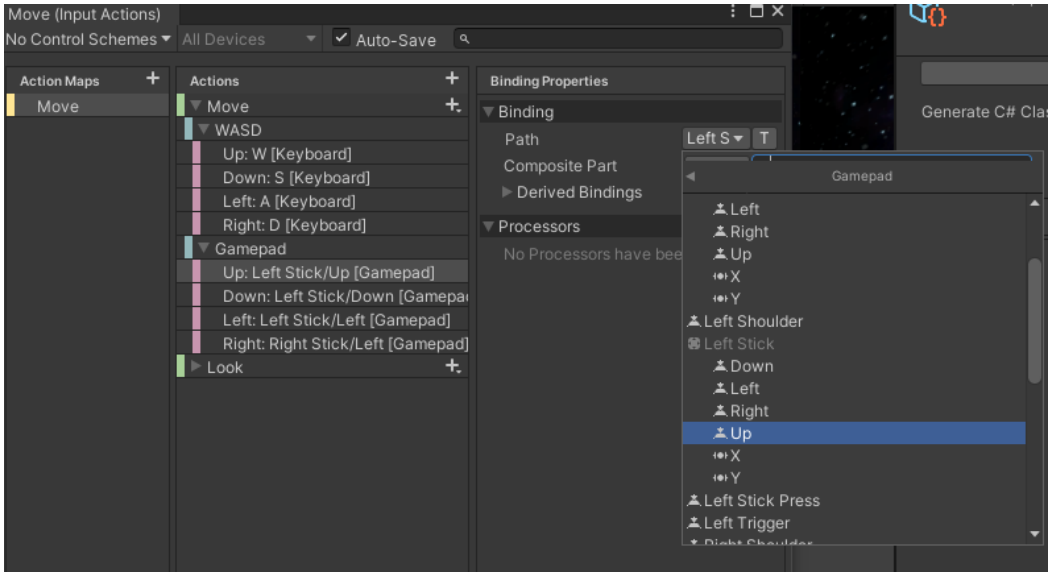
Підписуємо відповідно:



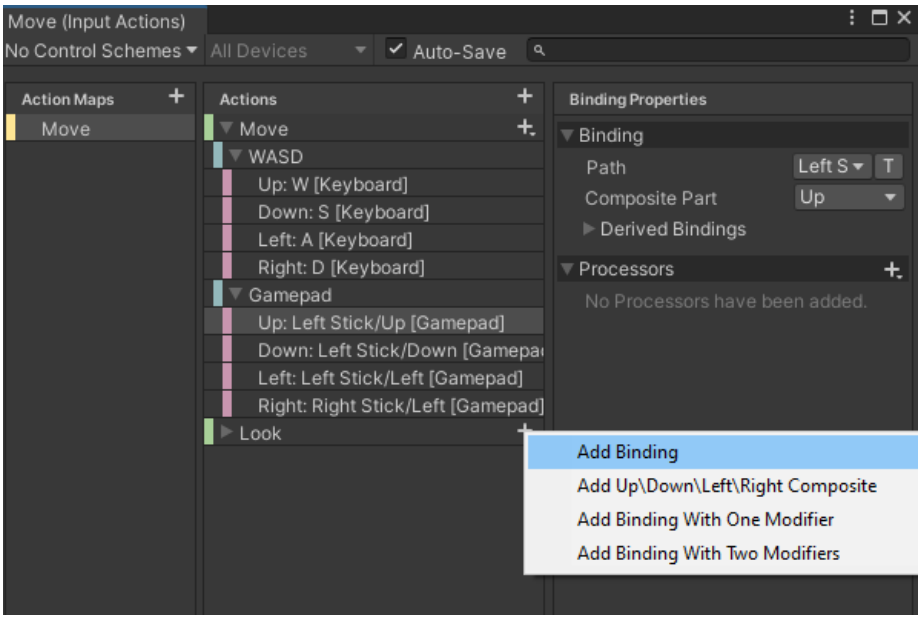
Обираємо кнопки для контролю:



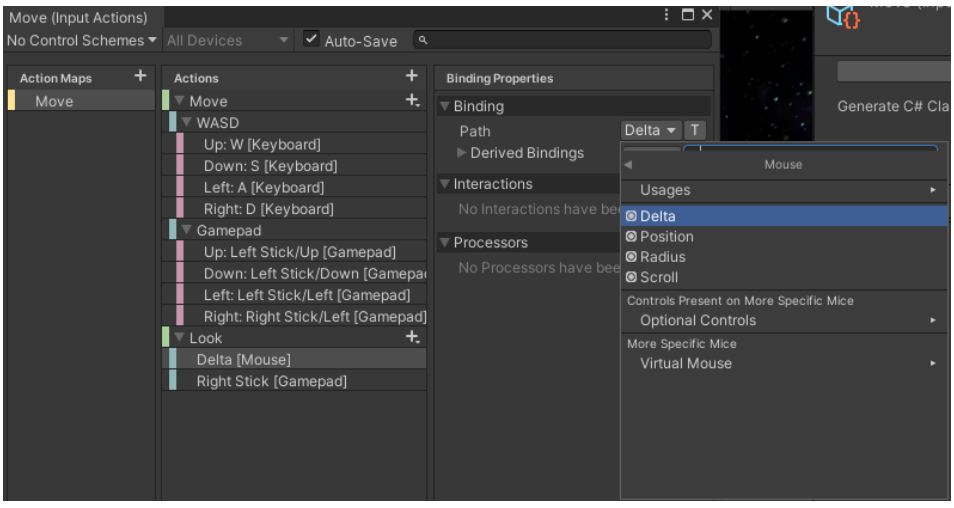
Повторюємо ті самі дії з геймпадом:



Також нам треба буде озиратися мишею, створимо звичайний біндінг



Відповідно назначаємо Mouse Delta та Right Stick:



Створіть скрипт C#, та надайте йому ім’я «Movement» для керування рухами персонажа. Напишіть код:

```
using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.InputSystem;


public class Movement : MonoBehaviour

{

    // Параметри швидкості гравця та швидкості повороту

    public float playerSpeed;

    public float rotationSpeed;


    // Компоненти для керування персонажем

    private CharacterController controller;

    private PlayerInput playerInput;

    private InputAction moveAction;

    private Transform cameraTransform;


    // Метод Start викликається один раз під час ініціалізації

    void Start()

    {

        // Блокування курсора миші

        Cursor.lockState = CursorLockMode.Locked;


        // Ініціалізація компонентів

        controller = GetComponent<CharacterController>();

        playerInput = GetComponent<PlayerInput>();

        moveAction = playerInput.actions["Move"];

        cameraTransform = Camera.main.transform;
```

```
    }

    // Метод Update викликається один раз за кадр
    void Update()
    {
        // Отримання вхідних даних від гравця
        Vector2 input = moveAction.ReadValue<Vector2>();

        // Розрахунок вектору руху
        Vector3 move = new Vector3(input.x, 0, input.y);
        move = move.x * cameraTransform.right.normalized + move.z * cameraTransform.forward.normalized;
        move.y = 0f;

        // Рух персонажа
        controller.Move(move * Time.deltaTime * playerSpeed);

        // Розрахунок повороту персонажа відповідно до напрямку камери
        Quaternion targetRotation = Quaternion.Euler(0, cameraTransform.eulerAngles.y, 0);
        transform.rotation = Quaternion.Lerp(transform.rotation, targetRotation, rotationSpeed * Time.deltaTime);
    }
}
```

Пояснення коду:

Оголошення змінних:

playerSpeed: швидкість переміщення гравця.

rotationSpeed: швидкість повороту гравця.

controller: компонент для фізичного керування персонажем.

playerInput: компонент для обробки вводу гравця.

moveAction: дії вводу для переміщення.

cameraTransform: трансформ головної камери для орієнтації руху.

Метод Start:

Блокує курсор миші для кращого контролю камери.

Ініціалізує компоненти, такі як CharacterController, PlayerInput, і встановлює moveAction для отримання вводу гравця.

Отримує трансформ головної камери для використання в розрахунках руху.

Метод Update:

Викликається один раз за кадр і обробляє ввід гравця.

Отримує вхідні дані руху від PlayerInput.

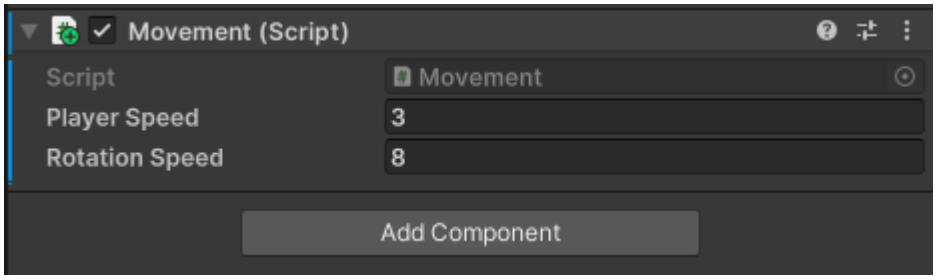
Розраховує вектор руху з урахуванням напрямку камери, щоб забезпечити правильний рух відносно положення камери.

Виконує переміщення персонажа за допомогою CharacterController.

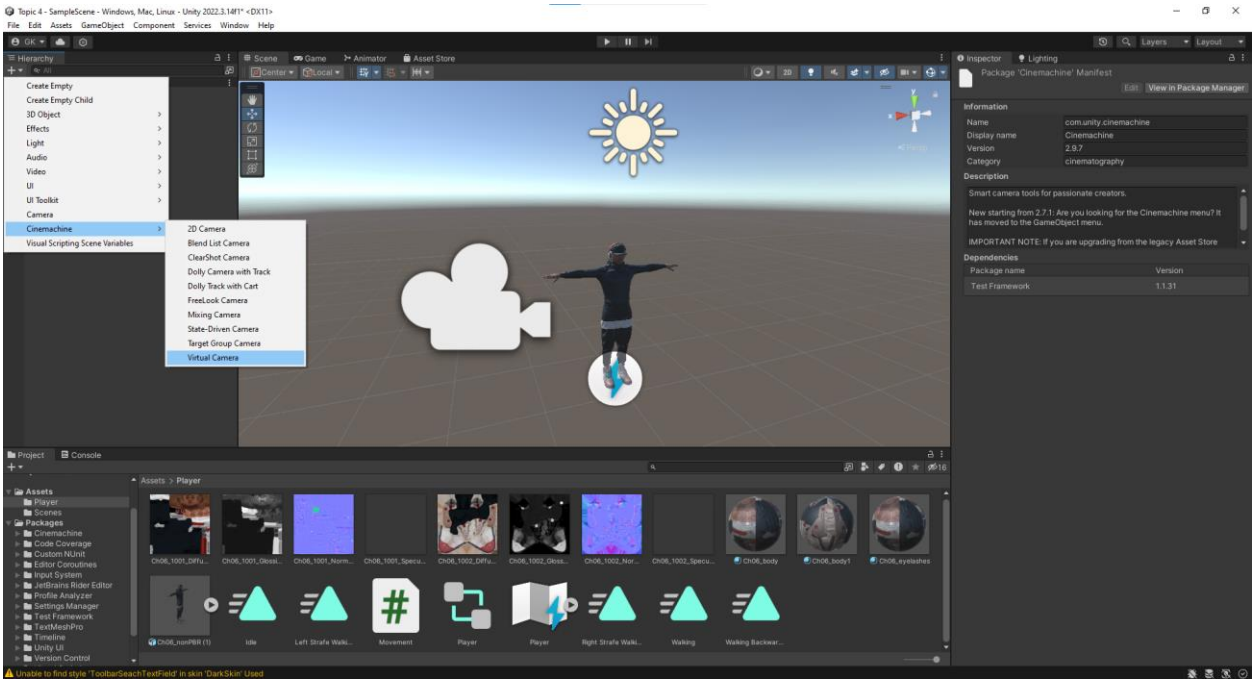
Розраховує і виконує плавний поворот персонажа відповідно до напрямку камери за допомогою інтерполяції Quaternion.Lerp.

Цей код забезпечує плавний та інтуїтивний контроль над персонажем у тривимірному просторі гри Unity, враховуючи напрямок камери та обробляючи ввід гравця для руху та поворотів.

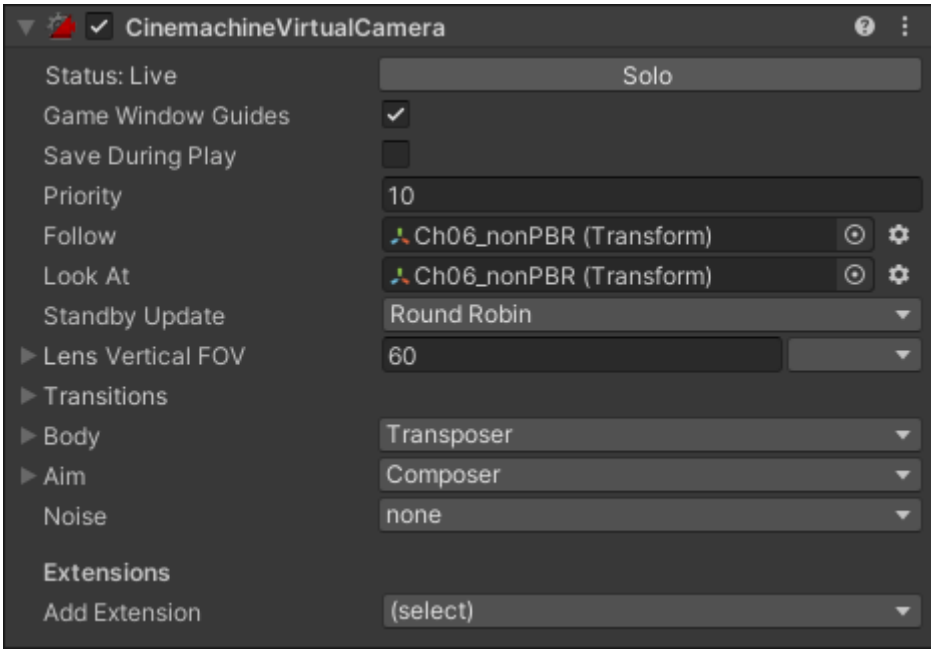
Додамо цей скрипт на персонажа і дамо стартові параметри швидкостей:



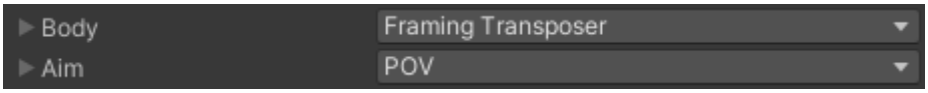
Останній момент це камера, створюємо віртуальну камеру як показано нижче:



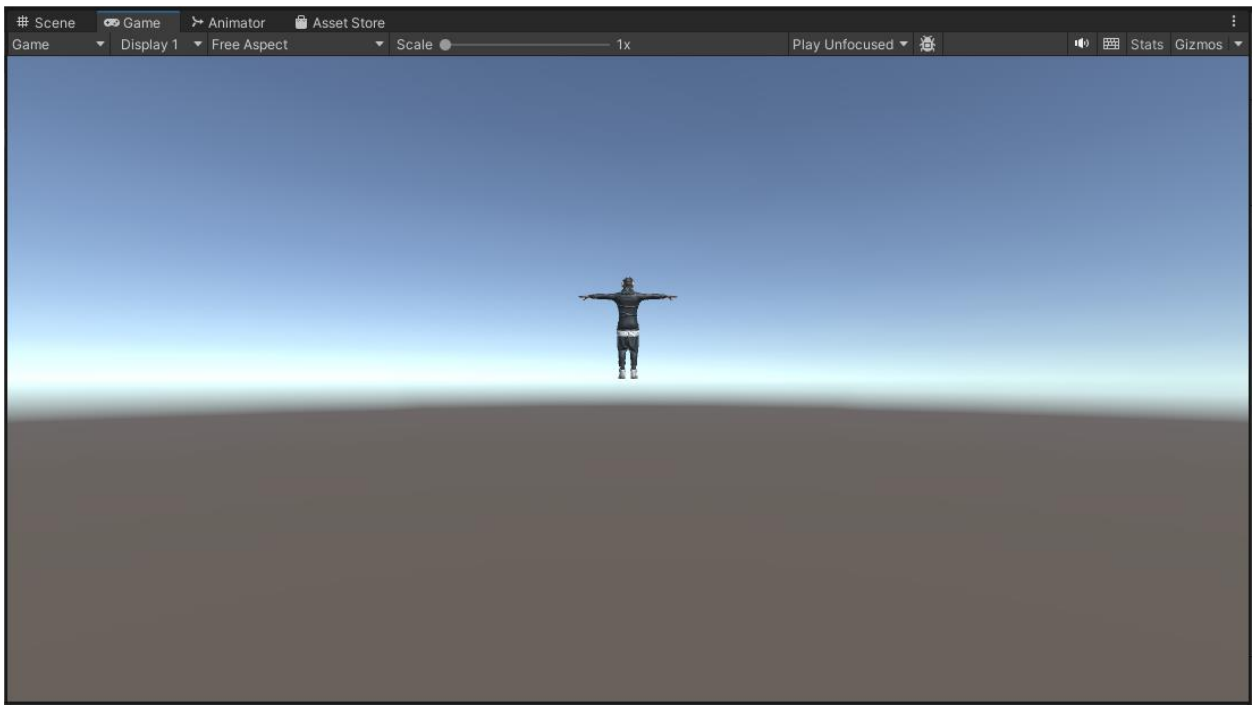
Встановлюємо персонажа на параметри Follow та Look at



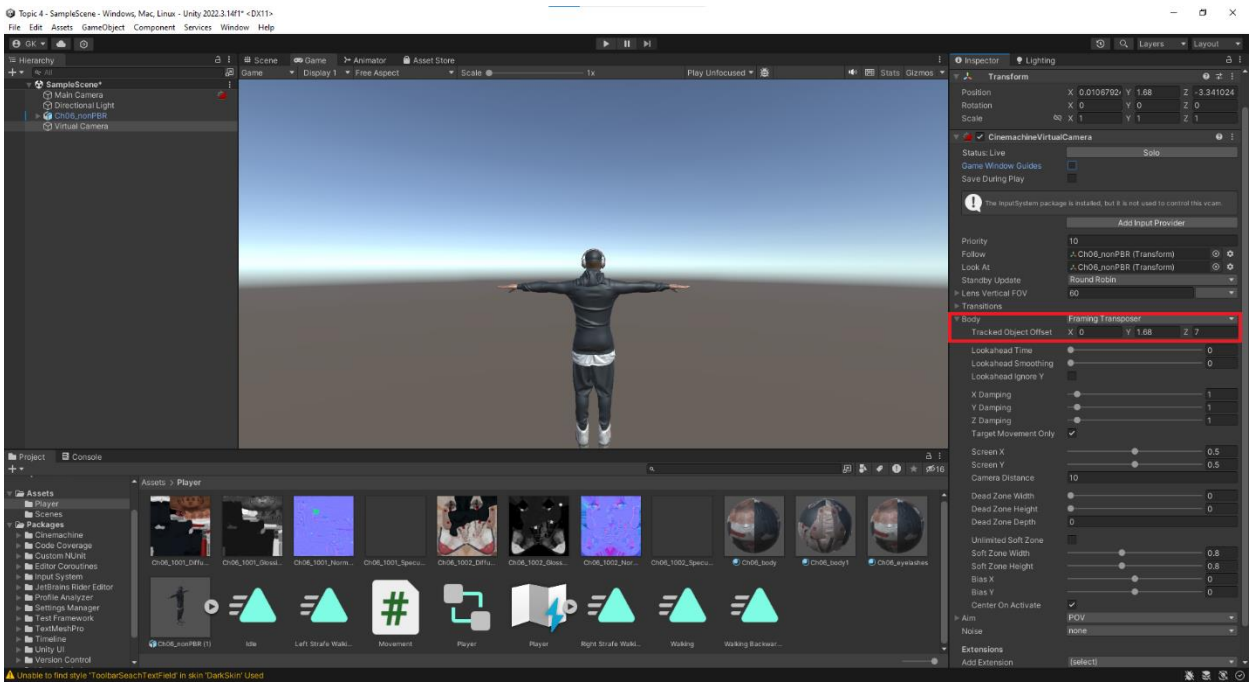
Після чого встановлюємо дані параметри як показано на скріні



У вікні Game спостерігаємо такий вигляд камери:



Оперуючи параметрами Offset виконуємо кінцеві налаштування



Додамо локацію згідно Теми 2, гра готова!