

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ
ОНТУ»

Спеціальність: 123 «Комп'ютерна інженерія»

Освітня програма: «Комп'ютерна графіка і Web-дизайн»

Група: 4КГ-05

Дипломний проект

здобувача освіти денної форми навчання
КГ.05.19.000.ДП

*Петрова Мирослава
Надеровича*

м. Одеса
2022 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОДЕСЬКОГО
НАЦІОНАЛЬНОГО ТЕХНОЛОГІЧНОГО УНІВЕРСИТЕТУ»

Спеціальність: **123 «Комп'ютерна інженерія»**

Освітня програма: **«Комп'ютерна графіка і Web-дизайн»**

Група: **4КГ-05**

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проекту (роботи) на тему:

Розробка крос-платформної гри на Unreal engine 4 з використанням Blue prints

Проектний матеріал складається з пояснювальної записки на _____ сторінках та
графічного (презентаційного) матеріалу на _____ аркушах (слайдах).

Дипломник _____ (Петров М. Н.)

Керівник _____ (Шевцов Ю.С.)

Консультанти:

з економічної частини _____ (Копайгородська Т.Г.)

з охорони праці _____ (Чорновол Н.І.)

з дотримання вимог ЄСКД _____ (Петрашова В.І.)

старший консультант _____ (Скорнякова О.В.)

До захисту допущений

Голова циклової комісії _____ (Скорнякова О.В.)

Завідувач відділення _____ (Суліма Ю.Ю.)

Захист « _____ » _____ 2022 р. Протокол ДКК № _____

Оцінка ДКК _____

Секретар ДКК _____

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНАХТ»

Відділення комп'ютерних систем Комісія КТ та ІІ
Спеціальність 123 «Комп'ютерна інженерія»
Освітня програма «Комп'ютерна графіка та web-дизайн»

ЗАТВЕРДЖУЮ:

Заст. дир. з НВР _____

“ _____ ” _____ 2022 р.

ЗАВДАННЯ

на дипломний проект (роботу)

Здобувачеві (здобувачці) освіти Петров Мирослав Надерович
(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Розробка крос-платформної гри на Unreal engine 4 з використанням Blue prints

затверджена наказом по коледжу від “ 30 ” грудня 202 1 р. № 306-A2-ОД

2. Термін здачі закінченого проекту (роботи) _____

3. Вихідні данні до проекту (роботи) Unreal Engine 4, Blender

4. Зміст розрахунково-пояснювальної записки (перелік питань, які необхідно розробити)

1. Розробка крос-платформної гри на Unreal Engine 4 з використанням Blueprints. 2. Економічний розрахунок. 3. Охорона праці.

5. Перелік графічного (презентаційного) матеріалу (з точним зазначенням обов'язкових креслень, кількості слайдів)

Презентація (10 слайдів)

6. Консультанти по проекту (роботі), із зазначенням розділів проекту, що їх стосується

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Основний	Шевцов Ю.С.		
Економічний	Копайгородська Т.Г.		
Охорона праці	Чорновол Н.І.		
Нормоконтроль	Петрашова В.І.		
Старший консультант	Скорнякова О.В.		

7. Дата видачі завдання _____

Керівник _____
(підпис)

Завдання прийняв до виконання _____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/р	Назва етапів дипломного проекту (роботи)	Термін виконання етапів дипломного проекту (роботи)	Відмітка про виконання
1	Розділ 1. Розробка крос-платформної гри на Unreal engine 4 з використанням Blue prints		
2	Розділ 2. Економічний розрахунок		
3	Розділ 3. Охорона праці		
4	Розробка презентації до дипломної роботи		
5	Чистове оформлення пояснювальної записки		
6	Підготовка доповіді до захисту		
7	Отримання рецензії, відповіді на зауваження		
8	Рецензента		
9	Захист роботи		

Дипломник _____
(підпис)

Керівник _____
(підпис)

[illegible]

АНОТАЦІЯ

Метою даної роботи є розробка кросплатформної гри за допомогою інструментів ігрового двигуну.

В даній випускній роботі молодшого спеціаліста розглянутий функціонал мови візуального скриптіngu Blueprints, процес розробки кросплатформної гри на ігровому двигуні Unreal Engine 4 та створення деяких асетів для гри.

ЗМІСТ

Вступ.....	7
1. Що таке кросплатформність	
1.1. Про кросплатформність в цілому.....	9
1.2. Кросплатформність у Unreal Engine.....	17
2. Ігровий двигун Unreal Engine	
2.1. Опис двигуна.....	18
2.2. Версії Unreal Engine та модифікації.....	19
3. Розробка простої гри на UE 4 з використанням Blueprints	
3.1. Опис майбутньої гри.....	29
3.2. Асети для гри (меші, текстури тощо).....	30
3.3. Процес розробки.....	31
4. Економічна частина	
4.1 Резюме.....	53
4.2. Визначення трудомісткості розробки програмного забезпечення.....	54
4.3 Розрахунок ціни програмного продукту.....	58
5. Охорона праці	
5.1 Аналіз небезпечних та шкідливих чинників, що впливають на працівника.....	61
5.2 Організація робочого місця.....	64
5.3 Пожежна безпека.....	65
Висновки.....	67
Перелік посилань.....	68

Вступ

У сучасному світі відеоігри стали одним з найпопулярніших видів розваг та, можна навіть сказати мистецтва. Це унікальне явище, яке може об'єднувати в собі цікавий сюжет і дивовижні світи як у відомих творах Толкієна, епічних героїв як у кінофільмах Марвел, захоплючу музику мабуть будь-яких жанрів, красиві пейзажі, від підводного чи підземного світу, до далеких галактик. Чи навпаки, гра може не мати в собі ні сюжету, ні красивого зображення, але все одно бути дуже захоплюючою. Ігри навіть зайняли своє місце у спорті – деякі онлайн проекти збирають тисячі і мільйони глядачів, які у прямому ефірі можуть спостерігати за своєю улюбленою командою на якомусь міжнародному турнірі з шутера Counter Strike чи улюбленим гравцем у стратегії Starcraft. Популярність відеоігор настільки велика, що по ним навіть знімають фільми (Оселя зла) чи навпаки, за фільмами або книгами створюють ігри (Alien: Isolation, Метро 2033).

Звісно ж не лише заради мистецтва створюється величезна кількість проектів різних жанрів на різних платформах. Створення ігор може мати і практичну користь у вигляді заробітку грошей.

За час розвитку відеоігор від не зрозумілих для більшості, розваг для «тіків» до масового явища, з'явилося багато великих компаній які займалися цим видом розваг, а також створення так званих «ігрових двигунів». Спочатку це були середовища розробки для внутрішнього використання компанії чи команди, але потім деякі розробники почали розповсюджувати свої двигуни для інших бажаючих.

Розробка гри це процес не простий, і готові ігрові двигуни допомагають іншим розробникам заощадити сили та час. Ігровий двигун дозволяє будь якому розробнику, будь то велика компанія чи одиночка,

					КГ 05.19.000 ДП ПЗ	Лист
						7
Изм.	Лист	№ докум.	Подпись	Дата		

маючи певні знання та навички розробляти свої ігри. Використовуючи інструменти ігрового двигуна, можна створити гру навіть не володіючи повноцінними мовами програмування.

За свої старання розробники двигуна беруть з клієнтів або відсоток від заробітку з їхніх проектів, або напряду продають ліцензію на використання свого продукту.

					КГ 05.19.000 ДП ПЗ	Лист
						8
Изм.	Лист	№ докум.	Подпись	Дата		

1. Ігровий двигун Unreal Engine

1.1. Про кросплатформність в цілому

Крос-платформний - це термін, який використовується для позначення частини програмного забезпечення, сумісного з більш ніж однією системою. Це стосується як ігор, так і програм. Наприклад, популярний медіаплеєр VLC сумісний із трьома основними настільними операційними системами: Microsoft, Mac OS та Linux. Крос-платформна підтримка також може поширюватися на мобільні пристрої, при цьому багато програм доступні як в Apple App Store, так і в Google Play Store.

Залежно від того, як запрограмовано конкретне програмне забезпечення, розробнику може знадобитися повністю перекодувати та переупакувати програмне забезпечення. Деякі фреймворки дозволяють розробникам більш плавно включати кросплатформену підтримку свого програмного забезпечення.

Розквіт кросплатформених ігор

Коли справа доходить до ігор, крос-сумісність – велика тема. У розпал консольних воєн з 1980-х до початку 2010-х кросплатформенна сумісність була великою проблемою. Основні виробники ігор, такі як Sony, Microsoft та Nintendo, виробляли та фінансували ігри. Ці ігри часто розроблялися власними студіями або ігровими студіями, з якими вони мали стратегічне партнерство.

Тому багато ігор були «ексклюзивними», їх можна було купити та грати лише на одному пристрої. Коли споживачі вирішують яку консоль купувати, вони повинні враховувати, які ігри доступні для конкретних консолей. Halo завжди буде на Xbox, Uncharted на Playstation, а Mario буде на консолях Nintendo.

					КГ 05.19.000 ДП ПЗ	Лист
						9
Изм.	Лист	№ докум.	Подпись	Дата		

У той час як Nintendo в основному дотримувалася ексклюзивних пропозицій, Microsoft і Sony почали пропонувати більш кроссплатформенну підтримку. Через зростання популярності комп'ютерних ігор в останнє десятиліття багато великих ексклюзивних ігор почали з'являтися на ПК та консолях. Через вищу стелю продуктивності на настільних комп'ютерах розробники також почали розробляти свої ігри з урахуванням потреб комп'ютерних геймерів і знижувати продуктивність при переході на консолі.

Багатокористувацька крос-гра

Крім можливості грати в різні ігри на одному пристрої, однією з найважливіших міркувань для тих, хто прагне сумісності на різних платформах, є кросплатформова гра, часто звана крос-грою. Це відноситься до функції, яка дозволяє людям, які грають в одну і ту ж гру на різних пристроях, грати разом в розрахованих на багато користувачів режимах. Наприклад, люди, які грають у Fortnite на Playstation 5, зможуть грати проти та в тандемі з користувачами-гравцями, які використовують ПК з Windows.

У міру того, як ігри стають менш ексклюзивними, останні кілька років зростає популярність кросплатформової гри. Незважаючи на те, що деякі супротивники все ще залишаються, наприклад, з Nintendo, більшість виробників та розробників ігор почали відкриватися для її реалізації. Люди з одних і тих же груп друзів, які володіють різними платформами, але все ж таки хочуть грати один з одним, отримують велику вигоду від ігор з цими функціями.

Крос гра також значно збільшує доступну базу користувачів конкретної гри, що добре як для гравців, так і для розробників. Активна велика спільнота скорочує час очікування, збільшує діапазон рівнів навичок, доступних для гри, і робить привабливішим для нових гравців початок гри.

Ось деякі з найвідоміших ігор із підтримкою крос-ігри станом на 2022 рік: PUBG, Apex Legends, Call of Duty, Fortnite, Minecraft, Genshin Impact.

До сьомого покоління ігрових консолей відеоігри зазвичай розроблялися для однієї консолі, і лише кілька ігор отримували

					КГ 05.19.000 ДП ПЗ	Лист
						10
Изм.	Лист	№ докум.	Подпись	Дата		

кроссплатформні версії на кількох консолях. Це було з унікальною архітектурою обробки кожної консолі, що робило розробку кожної закритої екосистеми і вимагало додаткових зусиль переносу інші системи. З появою сьомого покоління консолей, на якому Sony встановила стандартний процесор на базі x86 для PlayStation 3, кроссплатформенна розробка як для консолей, так і для персональних комп'ютерів стала набагато простіше з використанням стандартних програмних бібліотек та ігрових двигунів, а також мови сценаріїв, які відокремлюють специфічні для платформи деталі від конкретних елементів гри. Такі інструменти дозволяли випускати ігри одночасно для кількох платформ.

З появою Інтернету в ігри були включені розраховані на багато користувачів онлайн-компоненти, що дозволяють двом або більше користувачам одночасно грати на різних комп'ютерних системах. Ігри, випущені для платформи, можуть використовувати для цього переваги мережних бібліотек для конкретної платформи, наприклад, рівня Winsock для Microsoft Windows. У ці ігри не можна буде грати крос-платформно з іншими версіями, випущеними для інших систем. Натомість у більшості ігор з онлайн-компонентами, розробленими для декількох платформ, зазвичай використовуються стандартні функції типу TCP/IP для зв'язку між клієнтами гравців або між клієнтом та ігровим сервером, зводячи нанівець внутрішні відмінності між апаратними платформами.

Є деякі практичні обмеження для кроссплатформної гри. В іграх, де комп'ютер або консоль гравця виступає як сервер, можливості обладнання можуть накладати обмеження на кількість гравців, які може приймати цей сервер, і тим самим запобігати кроссплатформенній грі. Апаратне забезпечення також відіграє важливу роль при розгляді того, наскільки гравець може налаштувати гру на комп'ютері для роботи з високою частотою кадрів, тоді як консольні версії виправлені для оптимальної роботи на заданій конфігурації обладнання.

					КГ 05.19.000 ДП ПЗ	Лист
						11
Изм.	Лист	№ докум.	Подпись	Дата		

Найбільш частим обмеженням для підтримки кроссплатформенної гри з боку розробника є різниця у схемах управління між консолями та комп'ютерами. Зазвичай вважається, що комп'ютери з керуванням за допомогою клавіатури та миші на персональних комп'ютерах мають значну перевагу в іграх, що вимагають прицілювання, таких як шутери від першої особи, порівняно з аналоговими контролерами для консолей. Потім згодом розробляються консольні ігри з такими функціями як допомога в прицілюванні, щоб компенсувати відсутність точного управління. У 2010 році Рахул Суд, президент Voodoo PC, заявив, що Microsoft припинила кроссплатформенну гру між Xbox 360 і комп'ютерними гравцями для майбутньої гри, стверджуючи, що навіть досвідчені консольні гравці «щоразу знищувалися» у матчах з комп'ютерними гравцями із посередніми навичками через різницю між контролером і клавіатурою та мишею, і, таким чином, це буде розглядатися як утруднення для Xbox 360. Старший директор Microsoft з комп'ютерних та мобільних ігор Кевін Унангст заперечив на це, заявивши, що внутрішнє Тестування Microsoft показало, що багато проблем, пов'язаних з відмінностями у схемах управління, можуть бути усунені за рахунок дизайну та балансу гри. Blizzard Entertainment прагне реалізувати кроссплатформенну гру між версіями Overwatch для Xbox One і PlayStation 4, але не буде також включати платформу Windows, зокрема через перевагу, яку мають гравці з клавіатурою і мишею над контролерами, що сильно впливає на продуктивність у динамічній грі. Кліфф Блєжинські вважав, що кроссплатформенна гра в його гру LawBreakers була «нездійсненою мрією», оскільки він очікував, що, розмістивши такі інструменти, як допомога при прицілюванні, щоб допомогти консольним гравцям відповідати комп'ютерним гравцям, комп'ютерні гравці будуть засмучені і основа гравців відреагує на це негативно.

Забезпечення кроссплатформної гри розглядається як засіб збереження великої ігрової бази гравців навіть за кілька місяців після її випуску.

					КГ 05.19.000 ДП ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		12

Як правило, кросплатформова гра між персональними комп'ютерами з різними операційними системами легко дозволяється з використанням стандартних протоколів зв'язку та вимагає лише відповідного портування гри на інші системи; Комп'ютерна платформа у зв'язку з цим вважається дуже відкритою. Хоча цифрові онлайн-сервіси, що працюють на комп'ютері, стали популярними приблизно з 2010 року, ці системи, як правило, залишаються відкритими, надаючи розробнику інструменти, що дозволяють використовувати переваги кросплатформної гри. Наприклад, сервіс онлайн-ігор Valve, Steam, спочатку створювався для комп'ютерів з Windows, але в 2010 році розширився до систем OS X, а в 2013 - до Linux (включаючи налаштовану SteamOS від Valve). API Steamworks, запропонований розробникам через цю службу, дозволяє використовувати кросплатформенну гру в цих різних операційних системах, використовуючи при цьому можливості друзів, спілкування та підбору гравців, які пропонуються Steam.

Щодо консолей

До 2006 р. апаратні консолі зазвичай не мали вбудованих підключень до Інтернету, тому для підключення до Інтернету часто потрібно спеціальне обладнання. Це дозволило розгорнути деякі ігри як кросплатформні. Ранні спроби кросплатформної гри включають Sega Dreamcast 1998 з вбудованим модемом; Microsoft працювала з Sega, щоб надати розробникам версію Windows CE для створення кросплатформових ігор між Dreamcast та гравцями Windows для таких ігор, як 4x4 Evo, Maximum Pool, Quake 3 Arena та Phantasy Star Online. Sony випустить PlayStation 2 у 2000 році з підтримкою онлайн-ігри через зовнішній модем. 13 вересня 2001 Capcom vs. SNK 2 була випущена для Dreamcast та PlayStation 2 в Японії. Японські версії гри дозволяють гравцям з обох платформ змагатися між собою у мережі через службу KDDI Multi-Matching, що робить її першою грою, що дозволяє перехресну гру між ігровими консолями конкуруючих виробників. Square Enix представила онлайн-гру між користувачами PlayStation 2 і Windows для Final Fantasy XI в 2002 році.

					КГ 05.19.000 ДП ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		13

З появою ігрових консолей, готових до роботи в Інтернеті, таких як Sony PlayStation 3 та Microsoft Xbox, з'явилися онлайн-сервіси, які допомагають безпечно керувати обліковими даними гравця, покупками у цифровому магазині, списками друзів, обміном повідомленнями та іншими соціальними функціями, а також онлайн-підбором гравців для розрахованих на багато користувачів ігор. Незважаючи на те, що ці онлайн-сервіси надають перевагу гравцям, вони також допомагають керуючій компанії підтримувати узгоджений і привабливий досвід для своїх користувачів, гарантуючи, що ігри, оновлення та інший контент відповідають як бажаній якості, так і обмеженням контенту, щоб залучити нових гравців до цих консолей. Історично склалося так, що кроссплатформенна гра з консолями була дуже обмежена через ці послуги, і можливість створювати консольні ігри з кроссплатформенною грою вважається «святим Граалем» у секторі відеоігор.

Одна технічна проблема, з якою стикається кроссплатформенна гра на консолях, - це взаємодія між платформами, управління різними протоколами, використовуваними кожною службою. Тим не менш, технічні обмеження можна подолати, принаймні, три розробники заявили, що вони можуть увімкнути кроссплатформенну гру протягом дня, як тільки їм це буде дозволено. «Проблема зі конфігурацією» на короткий час дозволила гравцям комп'ютера, PlayStation 4 та Xbox One грати разом у спільній мережній грі Fortnite в середині вересня 2017 року. Це не було очікуваною особливістю гри, як зазначено у поточних специфікаціях гри. Хоча Epic Games виправила конфігурацію та заявила, що це помилка, короткі ситуації продемонстрували, що технічні перешкоди для кроссплатформної гри можуть бути подолані. Пізніше, у березні 2018 року, Epic оголосила про плани отримати Fortnite Battle Royale для Windows, PlayStation 4, Xbox One та мобільних пристроїв з можливістю кроссплатформної гри.

Як тільки ці технічні проблеми можуть бути подолані, основним обмежуючим фактором для кроссплатформної гри стануть умови обслуговування та політика допустимого використання, яких розробники та

					КГ 05.19.000 ДП ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		14

гравці повинні дотримуватись при використанні онлайн-сервісів консолей. Шухей Йошида з Sony, обговорюючи можливість кроссплатформної гри між платформами PlayStation і Xbox, зазначив, що «технічний аспект, можливо, буде найлегше» подолати порівняно з проблемами, пов'язаними з політикою та бізнесом. Деякі онлайн-сервіси мають обмеження на контент, пов'язаний з віком, які не дозволяють деяким іграм використовувати кроссплатформенну гру або відключати певні функції, щоб це дозволити; Дейв Хейджвуд, провідний розробник Rocket League, зазначив, що їм довелося запустити свою гру, яка підтримує кроссплатформенну гру між версіями Windows і PlayStation 4, без можливості для гравців спілкуватися між системами через вимоги до контенту, встановлених Sony. ; Пізніше вони змогли вставити фільтри, щоб дозволити цей зв'язок у рамках служби Sony. Valve довелося виключити PlayStation 3 і комп'ютерну кроссплатформенну гру з Counter-Strike: Global Offensive 2012 безпосередньо перед запуском, оскільки вони хотіли мати можливість регулярно оновлювати гру, що буде обмежено процесом сертифікації Sony на система PlayStation; Вони аналогічно намагалися перенести Steamworks на Xbox 360 для цієї гри, але також виявили, що політика сертифікації Microsoft занадто обмежує часті оновлення.

Протягом десятиліть виробники консолей активно працювали над захистом ексклюзивності гри у своїй системі від інших конкурентів консолей, для яких кроссплатформова гра може розглядатися як загроза. Деякі журналісти припустили, що кроссплатформова гра була обмежена виробниками консолей, щоб гарантувати, що гравці залишаться на їхній платформі для майбутніх ігор. Кайл Орланд з Ars Technica сказав, що якщо гравець хоче продовжувати грати в нові ігри з друзями, відсутність кроссплатформної гри вимагає від нього продовжувати купувати нові ігри для цієї консольної платформи, створюючи потужні мережеві ефекти. Алекс Перрі з Міс аналогічним чином зазначив, що відсутність кроссплатформної гри може призвести до того, що гравець спробує підштовхнути своїх друзів і

					КГ 05.19.000 ДП ПЗ	Лист
						15
Изм.	Лист	№ докум.	Подпись	Дата		

вплинути на них за допомогою однієї консолі, щоб вони могли грати разом, що збільшує продаж цього виробника консолей.

Щодо мобільних пристроїв

Як правило, ігри на мобільних пристроях, хоч і використовують операційні системи iOS, Android або Windows Mobile, не підтримують кросплатформенну гру. Мобільні ігри розробляються з урахуванням обмежень швидкості з'єднання стільникових мереж, тому більшість багатокористувацьких ігор часто являють собою покрокові стратегії, а не ігри в реальному часі. Багато розрахованих на багато користувачів ігри для мобільних пристроїв є асинхронними, коли гравці індивідуально виконують ходи або дії, ці дії відправляються в центральні служби і передаються іншим гравцям, на яких ці дії можуть вплинути.

Існують мобільні ігри, в яких є синхронна кросплатформова гра, зазвичай з використанням централізованих сервісів для нормалізації вибору платформи. Типовим прикладом є Hearthstone, який дозволяє мобільним гравцям кидати виклик гравцям на будь-якій іншій платформі, де випущена гра, включаючи комп'ютери. Microsoft представила серверні Realms у червні 2016 року, щоб дозволити гравцям Minecraft на пристроях Windows, iOS та Android грати разом, при цьому Xbox One був включений у 2017 році. Версія Minecraft "Bedrock" об'єднує можливості гри на Windows 10, мобільних пристроях, Xbox і Nintendo Switch, з кросплатформною грою для PlayStation, доданою в грудні 2019 року, а мобільні версії Fortnite Battle Royale були створені з кросплатформною грою з комп'ютерами та консолями.

Майбутнє кросплатформенності

Враховуючи, наскільки ця функція увійшла в ігри протягом останніх кількох років, ймовірно, в майбутньому ми побачимо більше ігор з кросплатформовою підтримкою. Ігрові студії почали усвідомлювати, наскільки крос-гра корисна для активності гравців.

					КГ 05.19.000 ДП ПЗ	Лист
						16
Изм.	Лист	№ докум.	Подпись	Дата		

1.2 Кроссплатформність у Unreal Engine

Двигун UE дозволяє розробляти як одиночні, так і багатокористувацькі кроссплатформні ігри. Підтримуються платформи Windows, OS X, Linux, PlayStation, Xbox, IOS, Android та інші.

У травні 2020 року Epic Games випустила безкоштовний SDK Epic Online Services, який дозволяє розробникам використовувати переваги своєї попередньої роботи у кроссплатформній багатокористувацькій грі для підтримки підбору гравців, списку друзів, досягнень та інших функцій у своїх іграх, підтримуючи персональні комп'ютери, PlayStation 4, Xbox One, Nintendo Switch, системи iOS та Android.

Epic Online Services SDK дозволяє розробникам легко інтегрувати кроссплатформенну гру в мобільні клієнти за допомогою встановленої серверної технології Epic.

					КГ 05.19.000 ДП ПЗ	Лист
						17
Изм.	Лист	№ докум.	Подпись	Дата		

2. Ігровий двигун Unreal Engine

2.1. Опис двигуна

Unreal Engine - ігровий двигун, що розробляється і підтримується американською компанією Epic Games. Першою грою на цьому двигуні був шутер від першої особи «Unreal», випущений у 1998 році. Хоча двигун спочатку був призначений для розробки шутерів від першої особи, його наступні версії успішно застосовувалися в іграх різних жанрів, у тому числі стелс-іграх, файтингах і масових багатокористувацьких рольових онлайн-іграх. У минулому двигун поширювався на умовах оплати щомісячної передплати; з

2015 року Unreal Engine безкоштовний, але розробники додатків, що використовують його, зобов'язані перераховувати 5 % роялті від загальносвітового доходу з деякими умовами.

Поточна політика ліцензування: Unreal Engine 4

2 березня 2015 року Unreal Engine 4 став безкоштовним. При цьому розробники ігор, як і раніше, повинні передавати 5% від виторгу з продажу гри (п. 5 ч. 10 ліцензійної угоди) компанії Epic Games, якщо щоквартальна виручка перевищує \$3000.

13 травня 2020 року Epic Games підняли поріг виторгу безкоштовного використання до \$1 000 000.

					КГ 05.19.000 ДП ПЗ	Лист
						18
Изм.	Лист	№ докум.	Подпись	Дата		

2.2. Версії Unreal Engine та модифікації

Unreal Engine 1

В 1998 році Unreal Engine 1 був одним з перших ігрових двигунів подібної універсальності; він поєднував у собі графічний двигун, фізичний двигун, штучний інтелект, управління файловою та мережевою системами та готове середовище розробки для ігор – UnrealEd. З огляду на рівень продуктивності більшості комп'ютерів на той час, розробники дещо спростили деякі елементи двигуна: систему виявлення зіткнень, мережевий код, код контролера для гравця.

Деякі технології двигуна Unreal Engine були революційно новими для того часу: наприклад, використання динамічного графа сцени. Ця технологія дозволяла додати ряд ефектів для накладання на поверхні:

Частково чи повністю дзеркальні поверхні.

Технологія варпінгу (warp) - можливість при промальовуванні замінювати зображення однієї поверхні проекцією зображення іншу, паралельну поверхню. Незважаючи на те, що ряд об'єктів міг безперешкодно переходити через варп-зони (наприклад, пущена гравцем ракета), була велика кількість обмежень на роботу таких зон. Дана технологія виглядала у грі як портали (через які можна було стріляти і проходити, але вони не пропускали звуки), які істотно випередили свій час. Дизайнери гри додавали до таких порталів чудовий динамічний ефект поверхні, що переливається і світиться, в результаті чого портали були однією з помітних особливостей Unreal 1.

Скайбокс (Skybox) — проектування на поверхні малювання з іншої точки (яка зазвичай поміщалася в невелику «коробку» з накладеною текстурою неба, звідси й назва Skybox). Таким чином, відображення не лицьової грані полігонів, а зворотної, із заздалегідь накладеною текстурою на об'єкт. У грі широко використовувався прийом масштабування текстур таким способом, що призводило до реалістичних видів довкілля. Для порівняння, у Quake 3, що використовує двигун-конкурент id Tech 3, також було динамічне

					КГ 05.19.000 ДП ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		19

небо, але на відміну від Unreal, воно не виглядало фотореалістичним (маючи крім того, помилку в декомпресії алгоритму S3TC, що призвела до появи градієнтів). Unreal був однією з перших ігор, що широко використовували мультитекстурування, що дозволяло (при правильному розміщенні джерел світла та орієнтації щодо пререндерених текстур) досягти ефекту тривимірності поверхонь (хоча справжніх технологій рельєфного текстурування на той час не було ще в жодній грі).

Двигун став одним із перших, у якому з'явився ефект гало навколо джерел світла, які плавно згасали, перекриті краями стін у міру руху гравця; лава та деякі текстури калюж води, в які падали краплі, зроблені динамічними завдяки використанню не шейдерів, як у пізніх іграх, а процедурних текстур; та фотореалістичне анімоване небо.

Спочатку двигун був випущений за допомогою двох платформ: Windows і Macintosh. Завдяки модульній системі двигуна була заявлена можливість портування двигуна на приставки "нового покоління" того часу і пізніше був успішно використаний на таких платформах як GameCube, PlayStation 2 і Xbox. Також підтримувалася незалежність мережевого коду від платформи клієнта — користувачі Mac OS могли грати в режимі Unreal з користувачами Windows.

Unreal Engine 1.5

У 1999 році вийшла вдосконалена версія двигуна Unreal, призначена для сучасних (на той час) комп'ютерів і консолей Dreamcast і PlayStation 2. Були внесені значні доповнення серед яких: підтримка лицьової анімації для персонажів; максимальна роздільна здатність текстур збільшено до 1024x1024, система «часток», що розширюється, технологія S3TC. Також було інтегровано другу версію редактора UnrealEd.

Ця версія двигуна використовується в розрахованому на багато користувачів шутері Unreal Tournament, і в пригодницькій грі «Гаррі Поттер і Філософський камінь».

					КГ 05.19.000 ДП ПЗ	Лист
						20
Изм.	Лист	№ докум.	Подпись	Дата		

Через кілька років група незалежних розробників UTPG звернулася до Epic Games за доступом до вихідних кодів виконуваних файлів Unreal Tournament, щоб продовжити підтримку гри. Epic Games прийняли пропозицію та деякий час команда розробників випускала неофіційні патчі для гри.

Unreal Engine 2

Друга версія Unreal Engine побачила світ у 2002 році з появою Unreal Tournament 2003, America's Army: Operations та Unreal Championship. У ній були практично повністю переписані ядро та механізм рендерингу, а також інтегрована нова (третя) версія редактора UnrealEd. Крім цього, у нього була інтегрована фізична підсистема Karma, що підтримує фізику лялькової ляльки та інші нововведення, що дозволили досягти більш реалістичної поведінки персонажів та взаємодії з навколишнім світом. Інші частини двигуна були також покращені або змінені для кращої сумісності з PlayStation 2, GameCube та Xbox.

З'явилися такі технології:

- Рідка поверхня (fluid surface) - плоский об'єкт, що складається з великої кількості полігонів, що імітує поверхню рідини. Такий об'єкт можна помістити на кордон з об'ємом води і він буде спотворюватися під дією об'єктів, що проходять через нього (наприклад, пірнаючого гравця).
- Листя (foliage) - об'єкти, що генеруються на льоту, які декорують ландшафт (наприклад, трава).
- Підтримка VoIP – можливість переговорів через мікрофон із іншими гравцями під час гри.
- Розпізнавання мовлення - переклад голосу в текст і обробка команди (наприклад, можливість віддавати ботам команди голосом). Ця технологія використовувала Microsoft Speech API і тому підтримувалась лише у 32-бітній версії Windows.

					КГ 05.19.000 ДП ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		21

- Більш складна фізика тіл, технологія «Карма» - обробка впливу на тіла не як вплив на один об'єкт, а як вплив на скелет (Ragdoll) з прив'язкою до нього частин тіла. У кожного суглоба персонажа з'явилися обмежувачі (обертання), ці обмежувачі бувають двох видів: шарнірні та лінійні. Лінійні можуть обертатися лише навколо однієї осі, а шарнірні по всіх трьох. Також сама дія на деякі об'єкти визначається не за коробками колізій, а за складними моделями, які також прив'язані до скелета.
- Фізика Ragdoll (як наслідок переходу на Karma) — незважаючи на те, що «скелетна» структура об'єктів була реалізована ще в першій версії двигуна, можливість використання фізики «ганчір'яної ляльки» з'явилася лише разом із переходом на нову фізичну підсистему.
- Транспортні засоби (vehicles) – можливість обробляти події від стороннього актора (актора). У Unreal Tournament 2003 реалізовані не повністю, є недопрацьований "актор" BullDog. Однак є реалізація рівня для UT2003, на якому знаходиться модифікований "актор", який виконує всі функції транспортного засобу. Транспортні засоби діляться за типами реалізацій - нерухомі (стаціонарні оборонні установки), складені з кількох об'єктів (Karma vehicle), з використанням скелетної структури та транспортні засоби, що прикріплюються до інших транспортних засобів (наприклад, кулемет на вежі танка). Деякі транспортні засоби можуть діяти без пілота, керовані власним штучним інтелектом.
- EAX 3.0 - двигун тривимірного звуку, розроблений Creative Labs.

У зв'язку з переходом на 32-бітну графіку та використання технології Karma, використання варп-зон стало утрудненим.

Unreal Engine 2.5

У цій версії був вкотре покращений і оптимізований графічний двигун - з'явилася підтримка Direct3D 9, OpenGL 2 і Pixomatic (остання система рендерингу служить для комп'ютерів зі слабкою відеокартою, але потужним

					КГ 05.19.000 ДП ПЗ	Лист
						22
Изм.	Лист	№ докум.	Подпись	Дата		

центрального процесора). Додано підтримку 64-бітних операційних систем Windows NT та GNU/Linux. Найбільша можлива роздільна здатність текстур піднята до 4096x4096 пікселів, додано повноцінну підтримку юнікоду (16-біт), що дозволило створювати повністю локалізовані ігри азіатськими мовами.

Крім того, сам двигун був значно оптимізований, що дозволило досягти більшої продуктивності за тих же системних вимог. З'явилася можливість відтворення відео у форматі DivX та Bink. У пізніших версіях впроваджено систему створення дерев SpeedTree.

Ця версія двигуна була використана в таких комп'ютерних іграх, як Unreal Tournament 2004, Pariah, Killing Floor та багатьох інших.

Unreal Engine 2 Runtime

Unreal Engine 2 Runtime є спеціальною версією двигуна Unreal Engine 2.0 (заснованої на білді 2226) з обмеженою ліцензією. Пізніше двигун був замінений на UDK - здешевлену версію двигуна Unreal Engine 3.

Користувач міг завантажити дистрибутив Windows-версії двигуна з офіційного сайту (для інших операційних систем потрібно було придбати ліцензію). У комплекті поставлявся редактор карт UnrealEd, утиліта usc, а також тестовий рівень та невеликий набір моделей та текстур, що демонструють можливості технології. Двигун безкоштовний для некомерційних проектів, а також для використання в освітніх цілях (наприклад створення 3D-презентацій).

Для використання у комерційних цілях потрібно придбати ліцензію.

Unreal Engine 2X

Спеціальна версія движка для консолі Xbox, яка базується на версії 2.0. Крім оптимізації коду, були введені нові візуальні ефекти, такі як depth of field (глибина зору), динамічна гамма-корекція, bloom та різні варіації blur. Формат текстур був змінений для більш реалістичного відображення тіней у високій роздільній здатності, доданий Memory Tracking, підтримка голосового чату, сервісу Xbox Live, а також функція розділення екрану

					КГ 05.19.000 ДП ПЗ	Лист
						23
Изм.	Лист	№ докум.	Подпись	Дата		

Був використаний лише у Unreal Championship 2.

Unreal Engine 3

Скріншоти Unreal Engine 3 були представлені в липні 2004 року, коли двигун перебував у розробці вже понад 18 місяців. Версія була розроблена з урахуванням персональних комп'ютерів, що використовують сучасні системи рендерингу (DirectX 9/10 та OpenGL 2/3), та консолей наступного покоління (PlayStation 3 та Xbox 360). У зв'язку з широким поширенням багатопроцесорних систем двигун використовує два паралельні основні потоки - основний потік (що відповідає переважно за ігровий процес) і потік рендерингу. Крім двох головних потоків можуть бути викликані другорядні, які виконують разові задачі. З'явилася підтримка багатопотокового динамічного завантаження даних (streaming), наприклад, завантаження «локації» безпосередньо при переміщенні по ній з метою економії ресурсів.

Оновлений графічний двигун підтримує більшість сучасних технологій, включаючи HDR, попиксельне освітлення, динамічні тіні, шейдерну модель четвертої версії, геометричні шейдери. Сам графічний конвеєр було переведено під управління шейдерів.

Від фізичної підсистеми Karma відмовилися, замінивши її на PhysX від компанії AGEIA. Пізніше AGEIA випустила набір додаткових бібліотек до гри, що дозволяють використовувати всі можливості фізичної системи (такі як ефект «рідини» або тканини).

За анімацію осіб персонажів відповідає вбудований анімаційний двигун FaceFX розробки OC3 Entertainment. Оновлено версію EAX до 5-ї. Додано підтримку технології SpeedTree для генерації дерев. Наголос був зроблений на additive-геометрії, проте від підтримки вичитувальної геометрії не відмовилися. Недоліком вичитувальної геометрії насамперед є набагато триваліший розрахунок освітлення.

Представлений новий редактор UnrealEd, переписаний з використанням wxWidgets.

					КГ 05.19.000 ДП ПЗ	Лист
						24
Изм.	Лист	№ докум.	Подпись	Дата		

Спочатку Unreal Engine 3 підтримував тільки платформи Windows, PlayStation 3 і Xbox 360, тоді як iOS (перша демонстрація - Epic Citadel) і Android були додані пізніше в 2010, з першими іграми Infinity Blade на iOS і Dungeon Defenders на Android. На заході Adobe Max в Лос-Анджелесі була продемонстрована робота двигуна за допомогою платформи Adobe Flash 11.

Першими іграми на Unreal Engine 3 були Gears of War для Xbox 360 і RoboBlitz для Windows, що вийшли 7 листопада 2006 року.

Unreal Development Kit (UDK)

Незважаючи на те, що Unreal Engine 3 був відкритий для створення модифікацій, можливість продавати ігри на базі UE3 надавали лише власникам ліцензії на ігровий двигун.

Однак, у листопаді 2009 року, компанія Epic Games випустила безкоштовну версію на базі Unreal Engine 3, що отримала назву Unreal Development Kit (UDK), яка стала доступною для кожного розробника-початківця. (Рух оновлювався паралельно з розвитком комерційної версії і згодом почав базуватися на Unreal Engine 3.5).

Відповідно до угоди (EULA), розробники ігор можуть продавати свої ігри зробивши одноразову виплату компанії Epic Games в сумі \$99. Якщо комерційний продукт на базі UDK набере в загальній сумі \$50,000 прибутку, то розробник буде змушений виплачувати 25% роялті компанії Epic Games. У грудні 2010 року була випущена версія UDK, яка підтримує створення ігор для платформи iOS.

Unreal Engine 3.5

У цій версії відбулася чергова еволюція графічної складової. Так, крім іншого, було додано фільтр пост-обробки Ambient occlusion, що покращує тіні та освітлення. Було збільшено кількість оброблюваних персонажів у кадрі. Перероблена на новому рівні технологія динамічної водної поверхні, фізика м'яких тіл і оточення, що руйнується, також значно покращені.

У березні 2010 року був опублікований анонс нових можливостей двигуна, які були представлені на Game Developers Conference 2010. Одним з

					КГ 05.19.000 ДП ПЗ	Лист
						25
Изм.	Лист	№ докум.	Подпись	Дата		

основних нововведень є нова система розрахунку освітлення Unreal Lightmass, що використовує глобальне освітлення без втрат функцій, які були реалізовані у старіших системах освітлення. Покращено роботу з багатопроцесорними системами — розподіл завдань між багатопроцесорними системами за допомогою Unreal Swarm, а також прискорення компіляції коду C++ та обробка Unreal Script за рахунок Unreal Build Tool (останнє збільшення продуктивності торкнеться лише розробників або творців аматорських модифікацій). Додано власний засіб поширення контенту користувача - Unreal Content Browser, з'явився Unreal Master Control Program - новий майстер-сервер, для оновлення клієнтів і ведення глобальної статистики, який вже використовується в Gears of War 2. У зв'язку з нововведеннями Epic Games China планувала продемонструвати використання двигуна для масових розрахованих на багато користувачів онлайн ігор.

У 2011 році компанією Epic Games був продемонстрований відеоролик про останні оновлення, внесені в двигун - Samaritan («Самаритянин»); преса та гравці стали називати його «Unreal Engine 3.9». У черговому оновленні двигун третього покоління отримав підтримку DirectX 11, а також безліч супутніх технологій.

Unreal Engine 4

- 18 серпня 2005 року віце-президент Epic Games Марк Рейн повідомляв, що Unreal Engine 4 вже два роки перебуває в розробці і його цільовими платформами є персональні комп'ютери та консолі 8-го покоління, а єдиною людиною, яка працювала над двигуном, є Тім Суїні. Сам Суїні на GDC 2006 оголосив, що розробка четвертого Unreal Engine не почнеться раніше 2008 року, оскільки UE3 буде актуальним як мінімум до 2010 року.
- 9 жовтня 2008 року на TGS група розробників Square Enix повідомила, що вони "бачили наступний Unreal Engine (Unreal Engine 4)" і "не

					КГ 05.19.000 ДП ПЗ	Лист
						26
Изм.	Лист	№ докум.	Подпись	Дата		

можуть дочекатися можливості попрацювати з ним". Раніше Square Enix випустила The Last Remnant і "не зустрічалася з проблемами при роботі з Unreal Engine 3".

- Презентацію технічної демонстрації Unreal Engine 4 було проведено на E3 2012 7 червня.
- 19 березня 2014 року Unreal Engine 4 розпочав своє вільне поширення для всіх бажаючих з підпискою в 19 \$ на місяць. Вихідні коди також викладено на репозиторій GitHub.
- 2 березня 2015 року Epic Games зробила заяву про зміну системи розповсюдження: двигун став безкоштовним для всіх розробників, за умови, що прибуток від додатків, створених на основі двигуна, не перевищує \$3000 за квартал.

Раніше Epic Games оголошувала про конкурс, під час якого розробники могли отримати 5 мільйонів доларів; для цього потрібно подати робочий прототип своєї гри.

Unreal Engine 5

13 травня 2020 року Epic Games офіційно анонсували Unreal Engine 5 і продемонстрували демо, у роздільній здатності QuadHD з частотою кадрів 30 FPS на консолі PlayStation 5. У демо була показана робота нових технологій: Lumen та Nanite. Lumen – нова технологія динамічного глобального освітлення. Nanite - одна з ключових технологій в основі двигуна Unreal Engine 5. Вона дозволяє показувати в кадрі стільки геометрії, скільки бачить око, і залежить від роздільної здатності - чим вона більша, тим вище стає деталізація. Вихід оновленої версії двигуна запланований на початок 2022 року. Компанія «Epic Games» 5 квітня 2022 відкрила доступ до ігрового двигуна «Unreal Engine 5» для всіх розробників.

Модифіковані Unreal Engine

Деякі компанії-розробники вважають за краще використовувати технології Unreal як основу для розробки власного двигуна.

					КГ 05.19.000 ДП ПЗ	Лист
						27
Изм.	Лист	№ докум.	Подпись	Дата		

Наприклад, Unreal Engine 2 був спеціально доопрацьований для ігор Deus Ex: Invisible War та Thief: Deadly Shadows, отримавши внутрішнє ім'я Flesh. Irrational Games однією з перших впровадила у свій Vengeance Engine (заснований на Unreal Engine 2.5) технологію HDR, а також фізичний двигун Havok 2.0 та інші функції, які нині широко використовуються в іграх. Цей двигун був використаний в Tribes: Vengeance і SWAT 4. Гра BioShock від Irrational спочатку також використовувала Vengeance Engine, але потім розробники частково замінили його на більш актуальний Unreal Engine 3. Модифікована версія стала називатися Vengeance Engine 2. Ubisoft використовують для своїх проєктів двигуни YETI engine та LEAD engine, створені на основі Unreal Engine 2.5.

					КГ 05.19.000 ДП ПЗ	Лист
						28
Изм.	Лист	№ докум.	Подпись	Дата		

3. Процес розробки простої гри на UE 4 з використанням Blueprints

3.1. Опис майбутньої гри

Створення великих ігор з великою кількістю механік, красивою графікою та іншим потребує дуже багато часу, та часто команду з великої кількості людей. На розробку таких ігор уходять роки, тому я створю невелику просту гру з простими механіками.

Суть гри буде полягати в тому, щоб летячи по нескінченному тунелю уникати перешкод якомога довше.

Для створення гри буде використовуватись Unreal Engine 4 та його система візуального скриптіngu Blueprints.

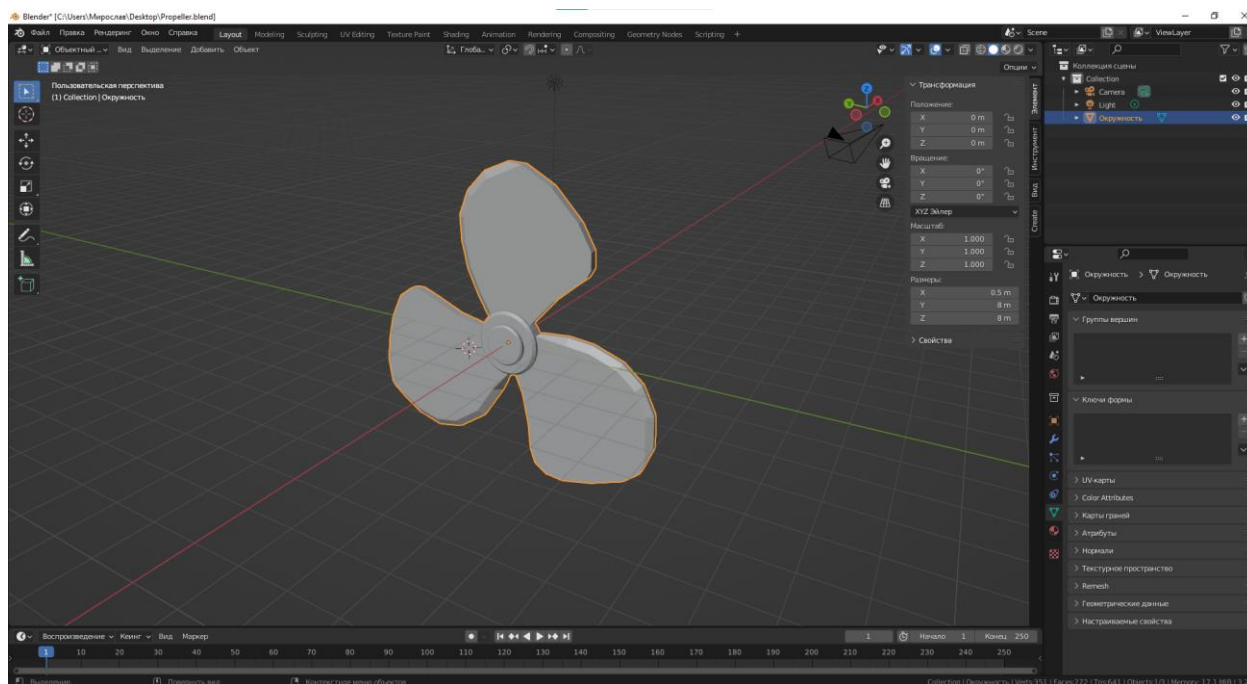
Гра розрахована на операційну систему Windows, але за допомогою інструментів двигуна, може працювати і на інших платформах, тобто може бути кросплатформеною.

					КГ 05.19.000 ДП ПЗ	Лист
						29
Изм.	Лист	№ докум.	Подпись	Дата		

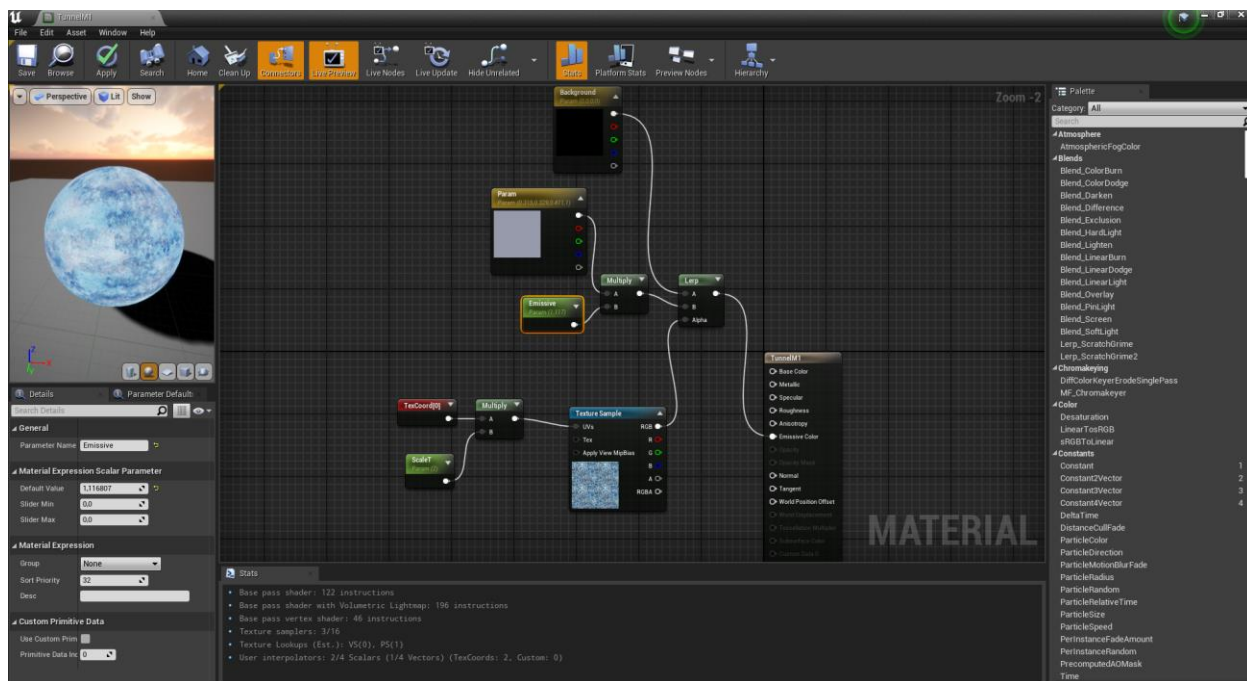
3.2. Асети для гри (меші, текстури тощо)

Перед початком розробки було підготовлено деякі асети, такі як моделі, текстури та блю прінти.

Деякі асети було взято з інтернету, деякі створено самому

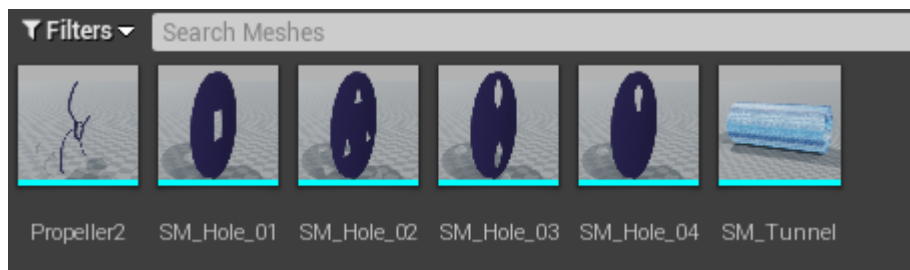


Створена у 3D редакторі Blender, модель пропелера яка буде використовуватись у грі як одна з перешкод



Матеріал, створений за допомогою текстури та нодів

					КГ 05.19.000 ДП ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		30



Меші, які будуть використовуватися у грі



Blueprints, які будуть використовуватись у грі.

- [BP_Player](#) – прінт, у якому задані налаштування гравця
- [BP_Tunnel](#) – прінт, у якому задані налаштування тунеля, в якому відбуваються події гри. Генерація перешкод
- [BP_Tunnel Spawner](#) – прінт, у якому задані налаштування безкінечного спавну тунелів
- [GM_Tutorial](#) – налаштування основних правил гри

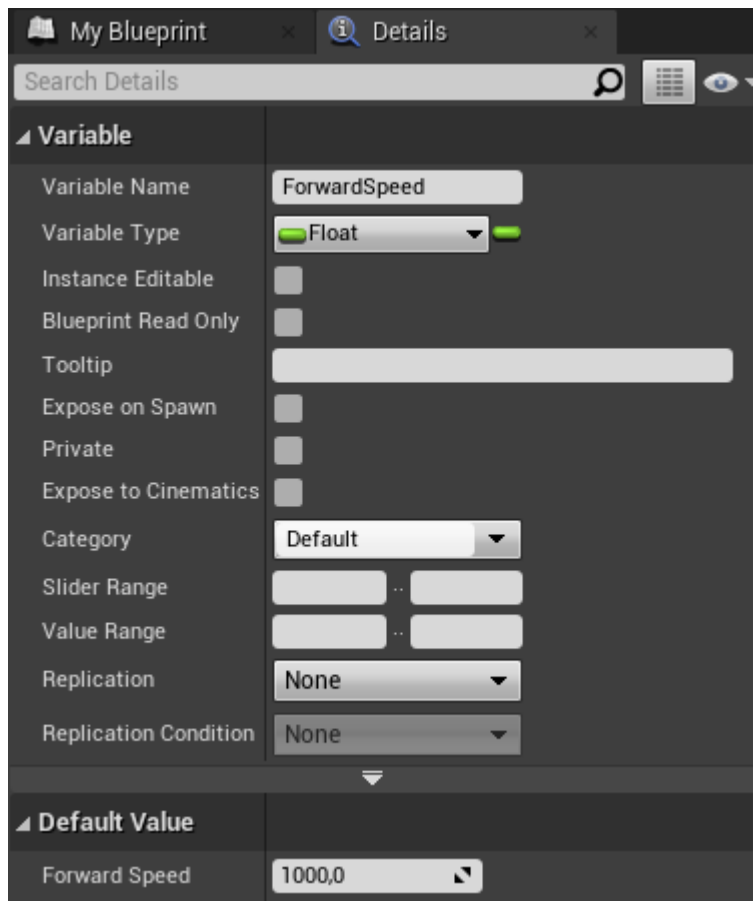
3.3. Процес розробки

Одне з першого що потрібно зробити, це рух гравця вперед.

Прінт [BP_Player](#)

Щоб рухати гравця вперед, потрібно додавати у кожному кадрі до розташування гравця зміщення.

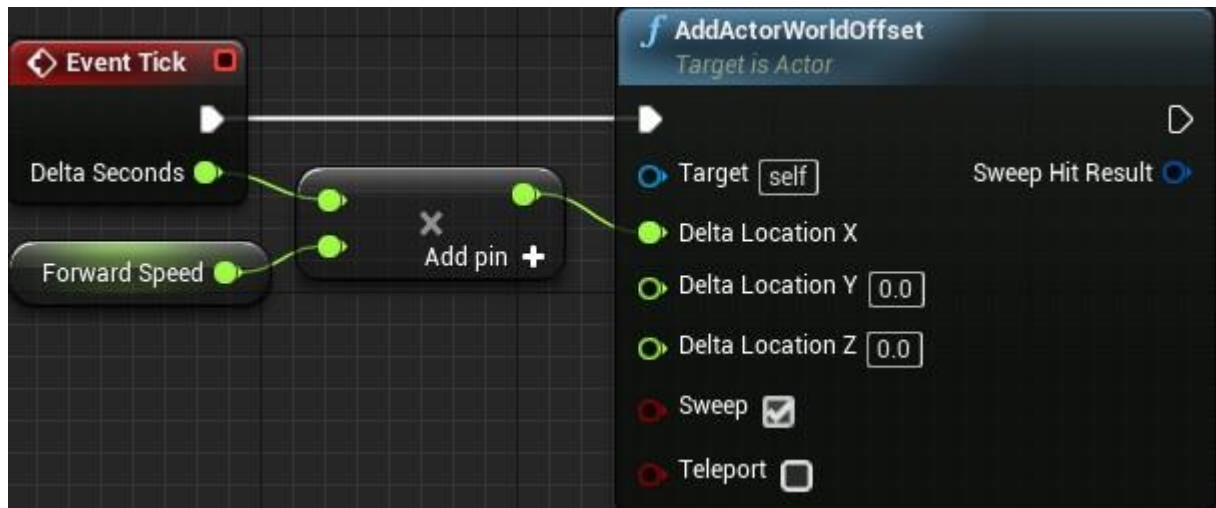
По-перше, нам потрібно створити змінну, яка задає швидкість руху гравця вперед. Створюємо змінну Float за назвою ForwardSpeed і встановлюємо початкову швидкість. Я встановив 1000



Щоб перемістити гравця, створюємо нод `AddActorWorldOffset`. Змінюємо значення `Sweep` на `true`, натиснувши прапорець.



Створюємо наступну схему:



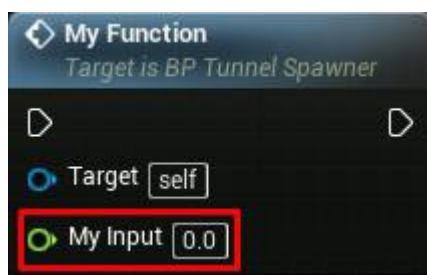
1. У кожному кадрі гра множитиме ForwardSpeed і Delta Seconds, щоб отримувати незалежний від частоти кадрів результат
2. AddActorWorldOffset буде використовувати цей результат для переміщення гравця вздовж осі X
3. Оскільки Sweep включений, гравець перериватиме рух уперед, коли його щось блокує

Створення системи спауна тунелів

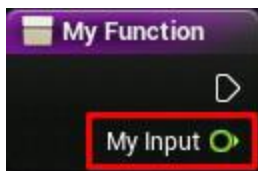
BP_TunnelSpawner

Гра створюватиме тунелі постійно, тому непогано було б створити функцію для спауна. У панелі My Blueprint створюємо нову функцію SpawnTunnel. Завданням функції буде спаун тунелю у вказаному місці.

Щоб надіслати функції розташування, потрібний вхідний параметр. Вони будуть відображатися під час виклику функції, як вхідні контакти.



Також вони будуть відображатися як вихідні контакти



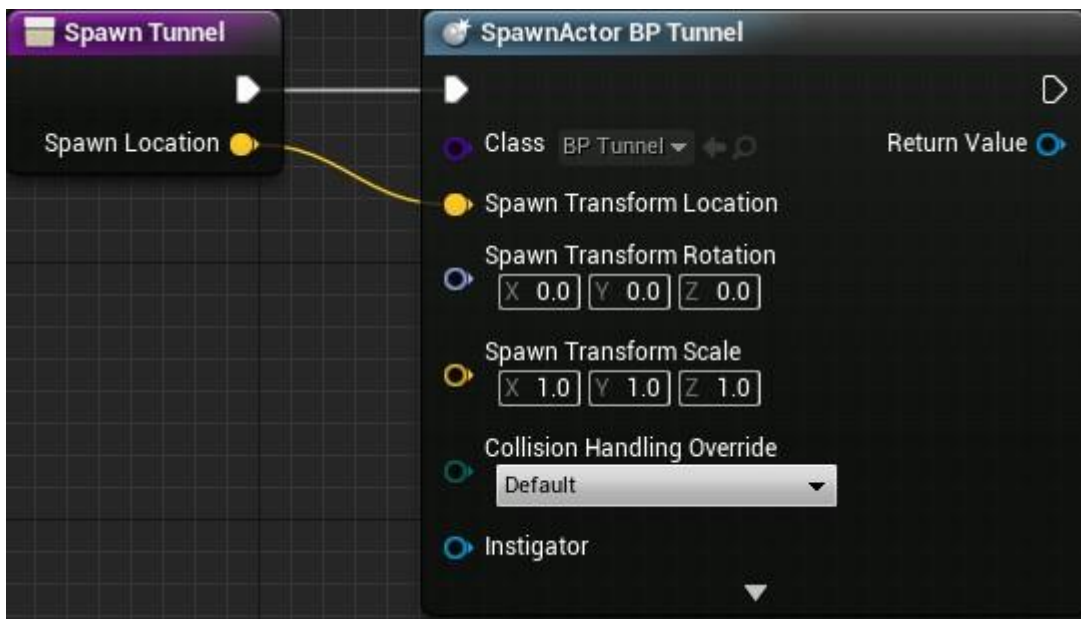
Створимо вхідний параметр. Переходимо до графа функції SpawnTunnel. Виберіть нод Entry та перейдіть до панелі Details.

Перейменуємо вхідний параметр у SpawnLocation та змінюємо його тип на Vector.

Щоб створити тунель, додаємо нод Spawn Actor From Class. Вибраємо Class BP_Tunnel



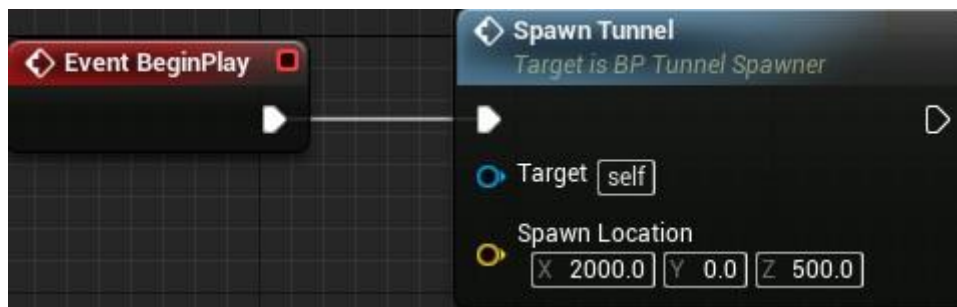
Щоб встановити розташування спауна, натискаємо правою клавішею на контакт Spawn Transform і вибираємо Split Struct Pin. Потім з'єднуємо нод Spawn Actor From Class з нодом Entry:



Тепер при виклику функції SpawnTunnel вона спаунить екземпляр BP_Tunnel в переданому їй місцезнаходженні.

У розділі Event Graph і знаходимо нод Event BeginPlay. Додаємо нод SpawnTunnel і з'єднуємо його з нодом Event BeginPlay.

У ноді SpawnTunnel задаймо Spawn Location значення (2000, 0, 500):



Тепер при запуску гри він спаунить тунель зверху і перед гравцем. Натискаємо Compile і повертаємось до основного редактора.

Спочатку видаляємо BP_Tunnel із рівня

Потім переходимо до Content Browser. Перетягніть BP_TunnelSpawner в Viewport. Завдяки цьому його екземпляр з'явиться на рівні.

Якщо натиснути на Play, то гра спаунить тунель зверху і перед гравцем

Завершивши перевірку, повертаємось до BP_TunnelSpawner. Скидаємо значення Spawn Location нода SpawnTunnel на (0, 0, 0).

Після цього натискаємо Compile і повертаємось в основний редактор.

Налаштування Blueprint тунелю

BP_Tunnel виконуватиме два завдання. По-перше, він визначатиме, коли гра має створити новий тунель. Для цього потрібно створити зону тригера. Після спрацювання тригера BP_Tunnel повідомлятиме BP_TunnelSpawner, що потрібно створити новий тунель. Завдяки цьому можна створити ілюзію нескінченного тунелю.

По-друге, він ставитиме точку спауна. Після цього BP_TunnelSpawner буде використовувати цю точку як наступне розташування спауна.

Створення зони тригера

Відкриваємо [BP_Tunnel](#) та переходимо до панелі Components. Додаємо компонент Box Collision та називаємо його TriggerZone.

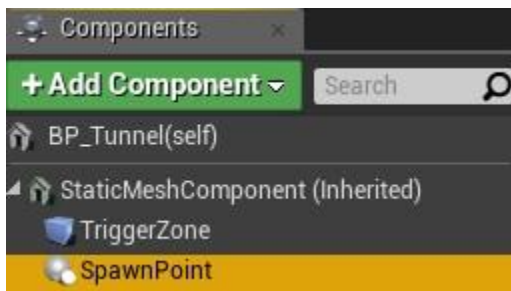
У панелі Details розділ Shape вказуємо властивість Box Extent значення (32, 500, 500). Це буде розмір зони тригера.

Тепер задаємо властивості Location значення (2532, 0, 0). Це розмістить TriggerZone прямо в кінець меша тунелю. Це означає, що новий тунель повинен буде створюватися тільки тоді, коли гравець досягне кінця тунелю.

Створення точки спауна

Для встановлення розташування точки спауна можна використовувати компонент Scene.

Переходимо до панелі Components та переконаймося, що нічого не вибрано. Додаємо компонент Scene і перейменовуємо його на SpawnPoint.



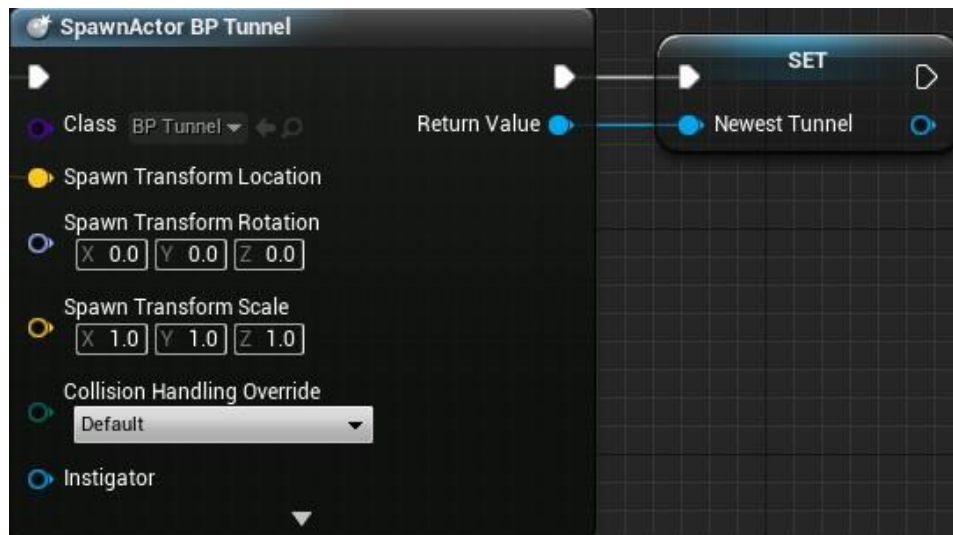
Меш тунелю має довжину 2500 одиниць по осі X, тому саме тут має бути точка приєднання. В панель Details і вказуємо властивість Location значення (2500, 0, 0).

Створення тунелів у точці спауна

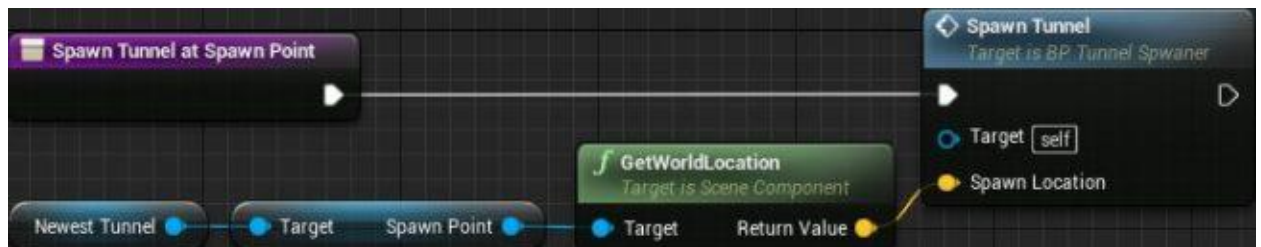
[BP_TunnelSpawner](#)

Наступний BP_Tunnel повинен спаунитися в SpawnPoint найдальшого тунелю. Завдяки цьому тунель завжди продовжуватиметься.

Відкриваємо граф SpawnTunnel. Натискаємо правою клавішею миші на контакті Return Value нода Spawn Actor From Class. Вибираємо Promote to Variable та перейменовуємо змінну на NewestTunnel.



Далі створюємо нову функцію під назвою `SpawnTunnelAtSpawnPoint`. Створюємо наступний граф:



Ця схема буде отримувати новий тунель і місце розташування його компонента `SpawnPoint`, після чого спавнити новий тунель в цьому місці.

Щоб `BP_Tunnel` міг обмінюватися даними з `BP_TunnelSpawner`, потрібне посилання. Без передачі даних `BP_TunnelSpawner` не знатиме, де спавнити наступний тунель.

Створення посилання на спаунер тунелів

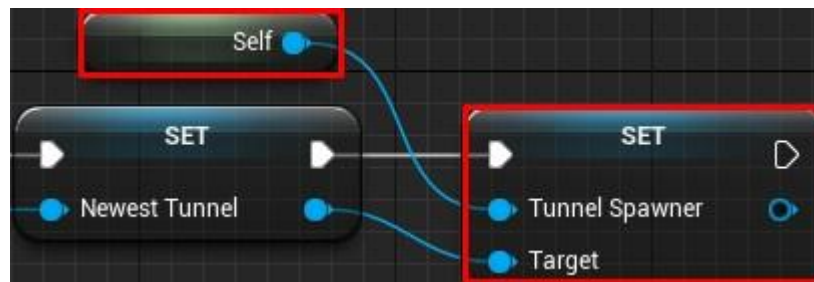
`BP_Tunnel`

Додаємо нову змінну `TunnelSpawner`. В якості `Variable Type` вибираємо `BP_TunnelSpawner\Object Reference`.



`BP_TunnelSpawner`

Відкриваємо граф `SpawnTunnel` і додаємо ноди:



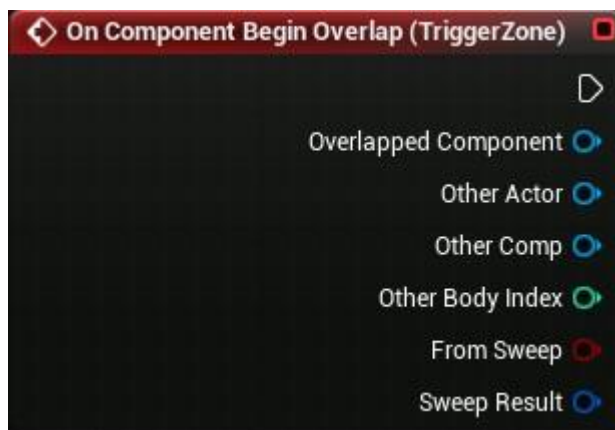
Тепер у кожного тунелю буде посилення BP_TunnelSpawner.

Далі потрібно повідомити BP_TunnelSpawner про необхідність спауна наступного тунелю, коли гравець входить до TriggerZone.

Скриптинг зони тригера

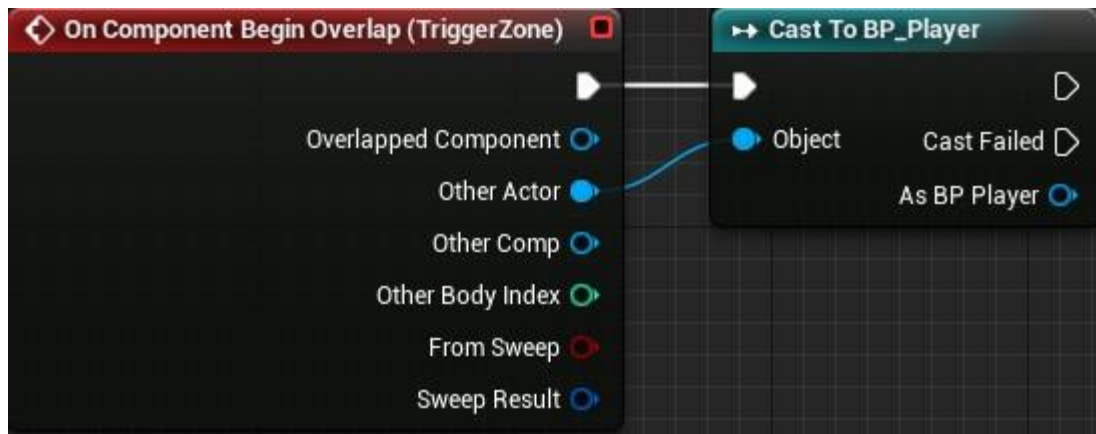
BP_Tunnel

У панелі Components натискаємо правою клавішею на TriggerZone. Вибираємо Add Event\Add OnComponentBeginOverlap. Це додасть у Event Graph наступний нод:



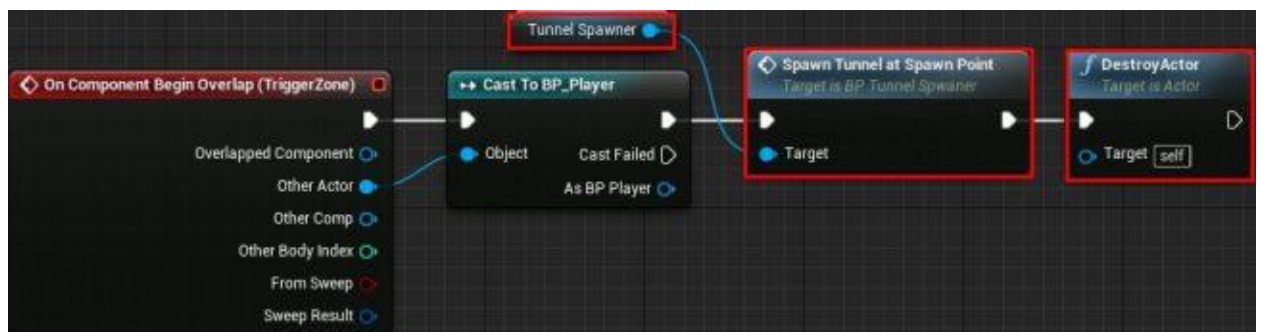
Цей нод буде виконуватися, коли інший Actor стосується TriggerZone.

Спочатку нам потрібно перевірити, чи Actor, що стосується TriggerZone, гравцем.



Тунель спауниться в кінці іншого тунелю, тому він призведе до спрацювання тригера TriggerZone цього тунелю. Cast to BP_Player запобігає виконанню всіх подальших нодів, якщо Other Actor є тунелем.

Тепер додамо позначені ноди після ноду Cast to BP_Player:



1. Коли Actor касається TriggerZone, виконується нод On Component Begin Overlap (TriggerZone)
2. Нод Cast to BP_Player перевіряє, чи є гравцем те, що торкається актора
3. Якщо це гравець, BP_TunnelSpawner створить новий тунель. Його розташування буде в компоненті SpawnPoint останнього створеного тунелю.
4. Оскільки старий тунель вже не потрібен, гра видаляє його за допомогою ноду DestroyActor.

Створення кількох тунелів

Хоча гра нескінченно створює тунелі, вони не виглядають нескінченними. виправити це можна, зробивши видимими кілька тунелів.

Перше, що потрібно зробити, - створити функцію, що створює певну кількість тунелів.

У принті `BP_TunnelSpawner` створюємо нову функцію під назвою `SpawnInitialTunnels`.

Щоб спаунити певну кількість тунелів, можна використовувати нод `ForLoop`. Цей нод виконуватиме з'єднані з ним ноди вказану кількість разів. Додаємо нод `ForLoop` та з'єднуємо його з нодом `Entry`.



Щоб нод `ForLoop` виконувався n разів, необхідно встановити для `Last Index` значення.

Ми будемо спаунити три тунелі. Для виконання трьох циклів задаємо `Last Index` значення 2 (бо нуль теж враховується).

Створення першого тунелю

Коли гра починається, гравець завжди повинен перебувати у тунелі. Для цього можна створити перший тунель у розташування гравця.

Щоб визначити, чи було створено перший тунель, ми можемо перевірити, чи встановлено `NewestTunnel`. Якщо ні, це означає, що перший тунель ще не створений, тому що `NewestTunnel` задається тільки після того, як гра створить тунель.

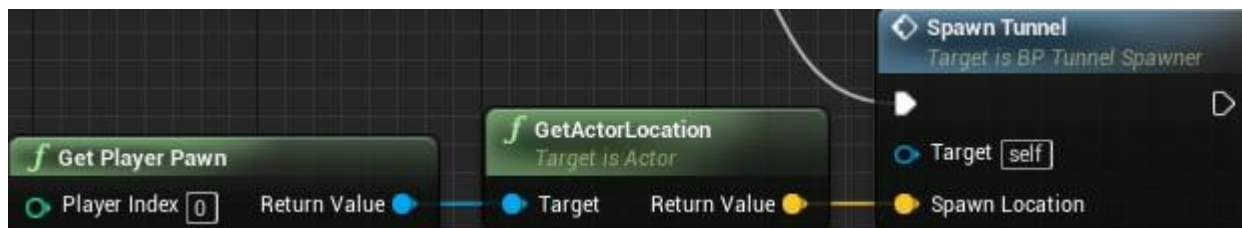
Щоб виконати цю перевірку, додамо нод `IsValid` (на якому є питання) після нода `ForLoop`.

Потім отримаємо посилання на `NewestTunnel` і з'єднаємо його з контактом `Input Object` нода `IsValid`.



Якщо `NewestTunnel` не заданий, буде виконано нод `Is Not Valid`, і навпаки.

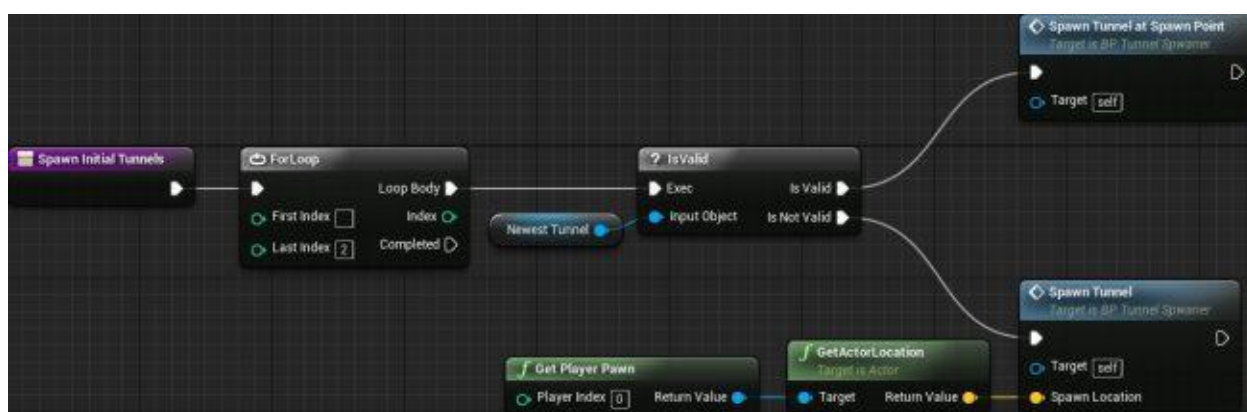
Ця схема буде створювати тунель у положенні Pawn гравця:



Потім нам потрібно створити наступні тунелі.

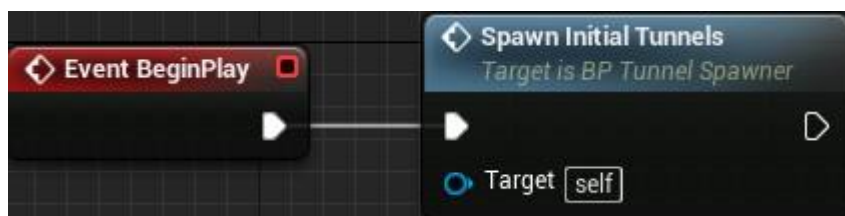
Створення наступних тунелів

Додаємо нод SpawnTunnelAtAttachPoint і з'єднуємо його з контактом Is Valid нода IsValid.



1. Нод ForLoop виконується три рази
2. У першому циклі він спаунить тунель у точці розташування гравця
3. У наступних циклах він спаунить тунель у точці SpawnPoint найновішого тунелю.

Тепер перейдемо в Event Graph та видалимо нод SpawnTunnel. Додамо нод SpawnInitialTunnels після Event BeginPlay.



Тиснемо Compile, та повертаємось в головний редактор.

Створення перешкод

Заготовлені раніше меші будемо спавнити у тунелі.

Відкриваємо BP_Tunnel та переходимо до панелі Components. Додайте компонент Static Mesh та назвемо його WallMesh.

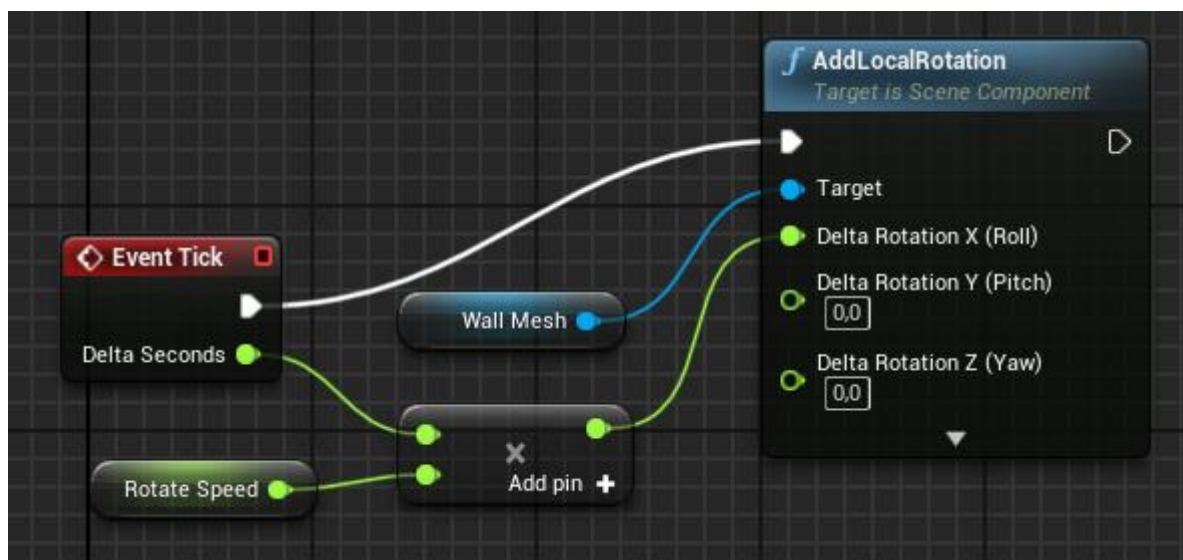
Перейдемо до панелі Details та змінюємо його властивість Static Mesh на SM_Hole_01.

Потім задамо його властивості Location значення (2470, 0, 0), розташувавши його таким чином у кінці тунелю.

Зробимо перешкодам можливість обертатися.

Додамо нову змінну Float та назвемо її RotateSpeed. Задамо Default Value значення 30.

Перейдемо в Event Graph і знайдемо нод Event Tick. Створемо таку схему:

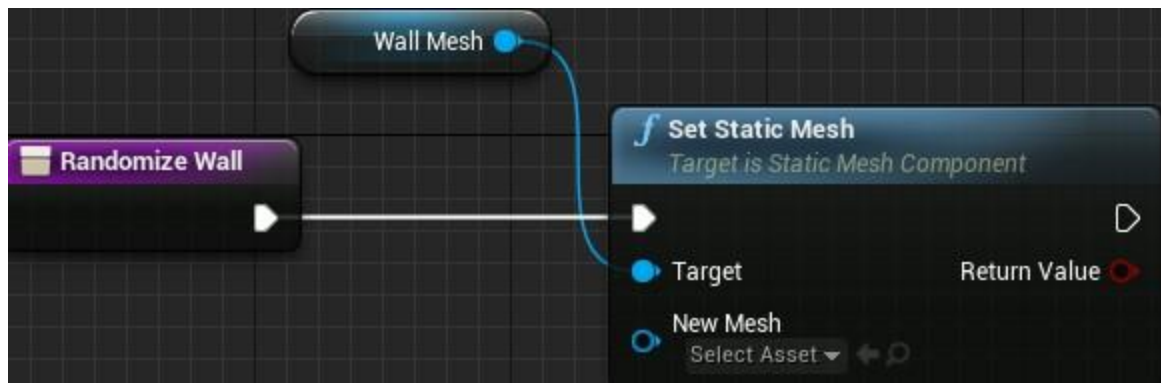


Це змусить WallMesh повертатися кожен кадр на задану величину.

Створення варіацій стін

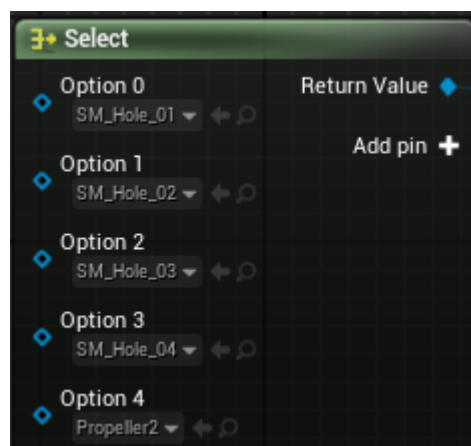
Замість створення нового Blueprint для кожної варіації ми можемо просто рандомізувати WallMesh.

У BP_Tunnel створемо нову функцію RandomizeWall. Потім створемо наступний граф:



- Set Static Mesh задаватиме WallMesh переданий меш.
- Для створення списку статичних мешей можна використовувати нод Select.
- Нод Select дозволяє задати список варіантів. Вхід Index визначає, який варіант виводитиме нод Select.

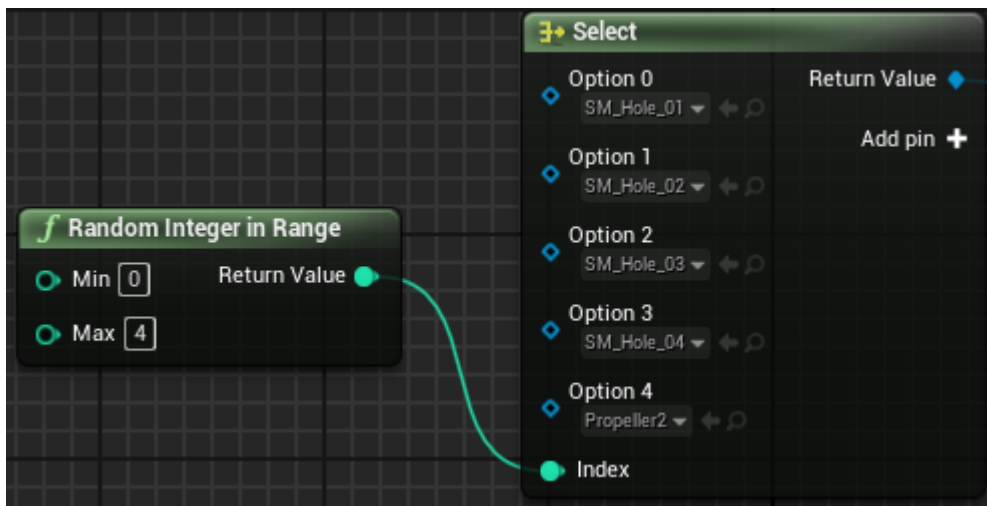
У нас є п'ять мешей перешкод. Задамо їх у ноді



Рандомізація перешкод

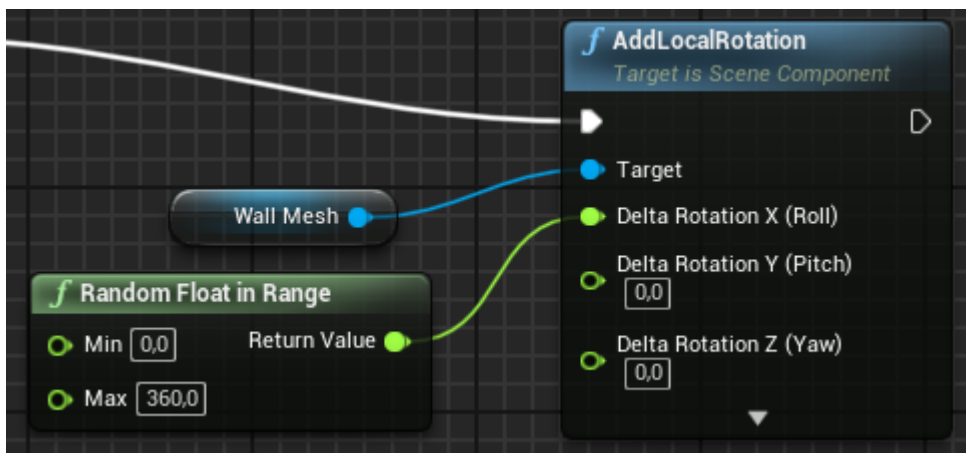
Тепер зробимо так, щоб перешкоди спавнились у рандомному варіанті.

Для отримання випадкового числа можна використовувати нод Random Integer in Range.



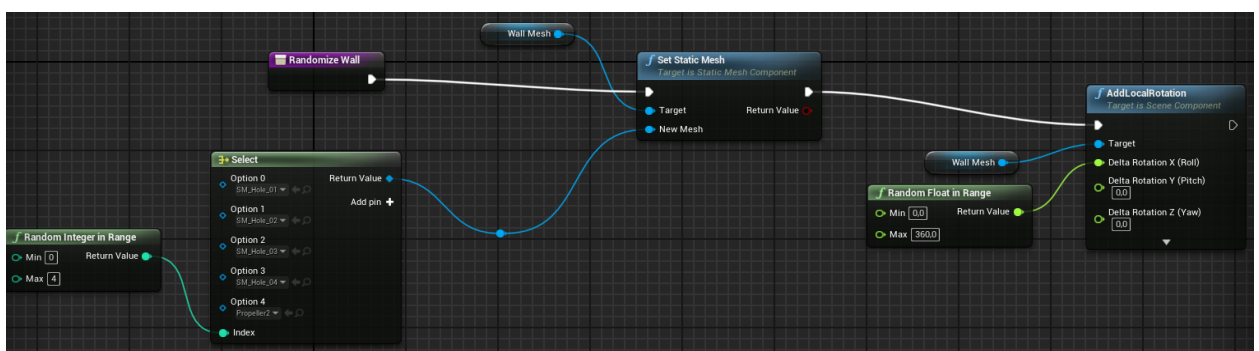
Задамо Max значення 4, це дасть нам п'ять можливих чисел – 0, 1, 2, 3, 4

Щоб додати випадковість, додамо WallMesh випадковий поворот.
Додамо наступне після ноду Set Static Mesh:



Це додасть WallMesh випадковий поворот в інтервалі від 0 до 360 градусів.

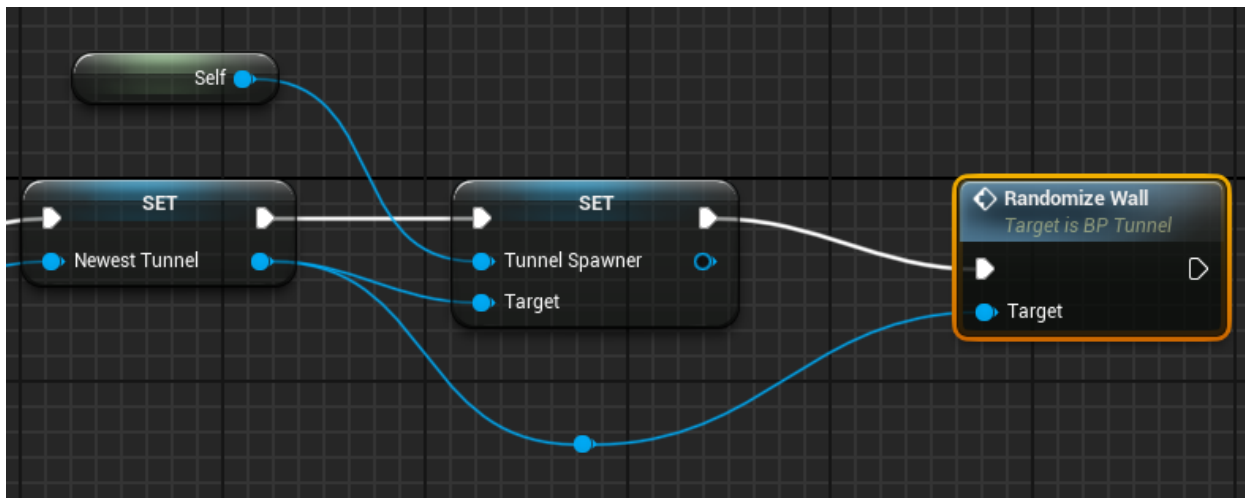
Ось як виглядає готовий граф:



1. Нод Select передає список мешей
2. За допомогою ноду Random Integer in Range вибирається випадковий меш
3. Нод Set Static Mesh задає WallMesh вибраний меш

4. AddLocalRotation додає зсув випадкового повороту WallMesh
Тиснемо Compile і закриваємо граф RandomizeWall.

Перейдемо на BP_TunnelSpawner і відкриємо граф SpawnTunnel.
Додамо виділений нод:



Тепер при спауні тунелю у нього буде випадковий міш стіни.

Натиснемо Compile та повернемося до основного редактора.

Обробка колізій зі стінами

Якщо гравець зіткнеться зі стіною, то перестане рухатися вперед.
Однак якщо продовжити рухатися і пройти крізь отвір, знову почне рухатися вперед.

Наступним кроком буде відключення руху вперед, після зіткнення гравця зі стіною.

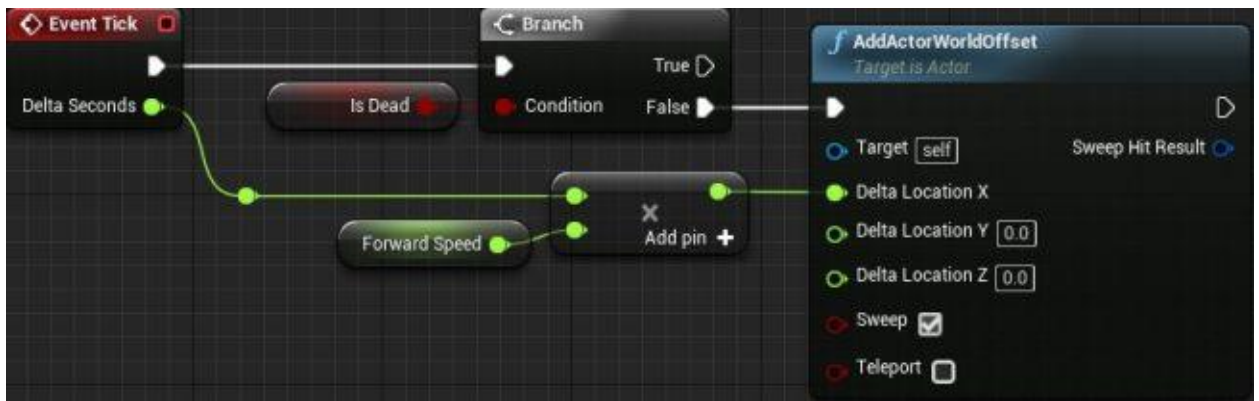
Для увімкнення та відключення руху вперед можна використовувати змінну Boolean. Вона має лише два стани: true та false.

Відкриємо BP_Player та створимо нову змінну Boolean під назвою IsDead.

Перейдемо до ноду Event Tick та створимо нод Branch.



З'єднаємо нод Event Tick із нодом Branch. Потім з'єднаємо контакт False нода Branch з нодом AddActorWorldOffset.



Тепер, коли IsDead дорівнює true, гравець зупинятиметься.

Задання змінної IsDead

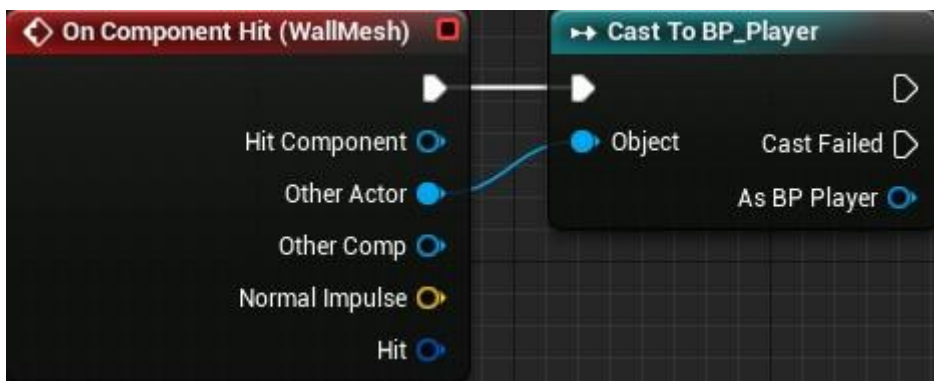
Натиснемо Compile і перейдемо на [BP_Tunnel](#). На панелі Components натиснемо правою клавішею миші на WallMesh і виберемо Add Event\Add OnComponentHit. Це додасть до Event Graph наступний нод:



Цей нод буде виконуватись при зіткненні іншого Actor з WallMesh.

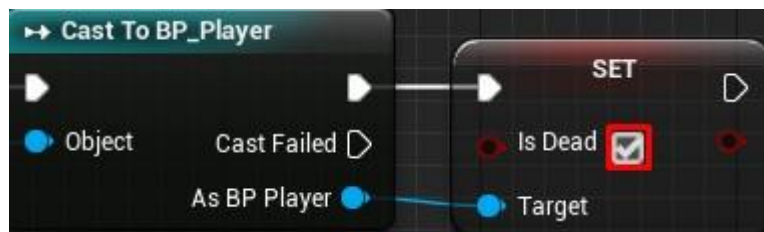
По-перше, нам потрібно перевірити, чи є гравцем Actor, який зіштовхнувся з WallMesh.

Створемо нод Cast to BP_Player.



Перетягнемо контакт BP_Player нода Cast to BP_Player та додаймо нод Set Is Dead.

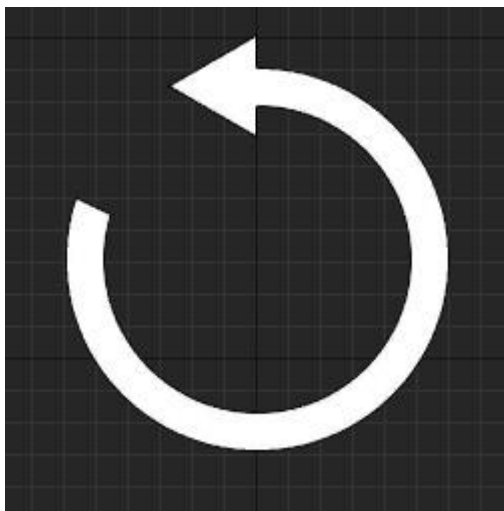
Встановимо IsDead значення true, поставивши прапорець.



Тиснемо Compile і повернемося до головного редактора.

Відображення кнопки перезапуску

Для того щоб у разі зіткнення зі стіною, гравець міг перезапустити гру, потрібно додати кнопку перезапуску. Для цього ми будемо використовувати віджет

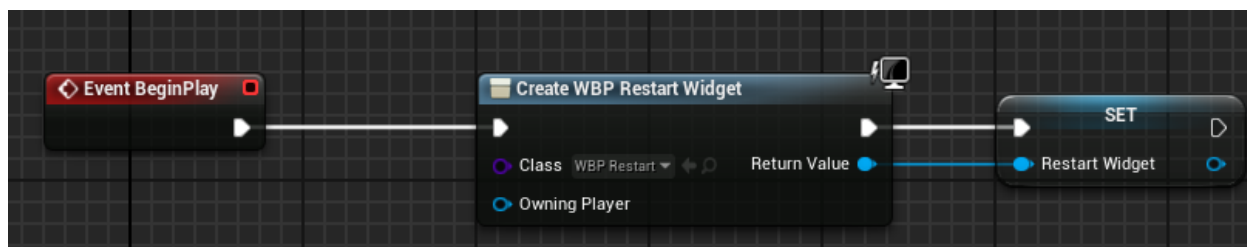


Щоб відобразити або приховати віджет, нам потрібне посилання на нього. Відкриємо BP_Player та створимо нову змінну RestartWidget. Змінимо Variable Type на WBP_Restart\Object Reference.

Потім перейдемо в Event Graph і знайдемо нод Event BeginPlay.

Додамо нод Create Widget і задамо Class значення WBP_Restart.

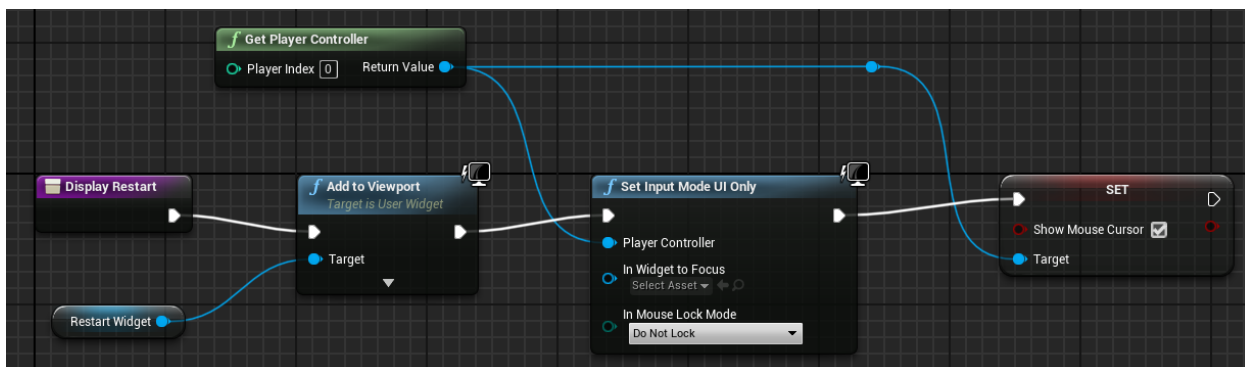
Тепер додамо нод Set Restart Widget і з'єднаємо все так:



Тепер при спауні гравця ми створюватимемо екземпляр WBP_Restart. Наступним кроком буде створення функції, яка відображатиме цей екземпляр.

Створення функції відображення

Створимо нову функцію DisplayRestart. Та створимо наступний граф:



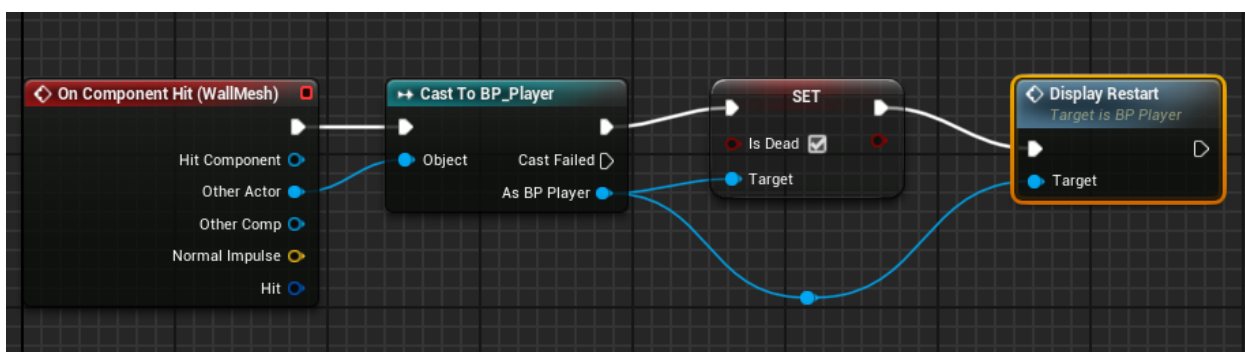
1. Add to Viewport відображає RestartWidget на екрані
2. Set Input Mode UI Only дозволяє гравцеві взаємодіяти лише з UI. Ми робимо так, щоб гравець не міг пересуватися, доки він мертвий.
3. Set Show Mouse Cursor відображає курсор миші

Виклик функції відображення

Закриємо граф DisplayRestart і натиснемо Compile.

Перейдемо на BP_Tunnel і знайдемо нод On Component Hit (WallMesh).

Додамо нод DisplayRestart до кінця ланцюжка нодів.



Натиснемо Compile і закриємо BP_Tunnel. Повернемося до головного редактора. Тепер коли гравець стикається зі стіною, з'являється кнопка перезапуску.

Перезапуск гри та скидання гравця

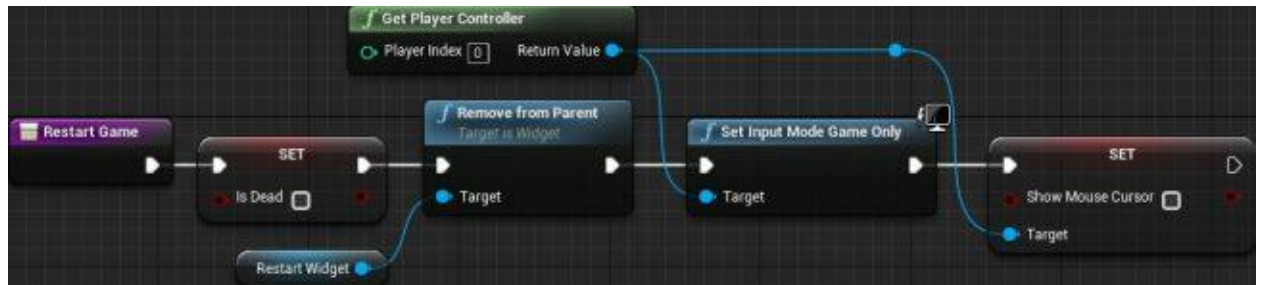
При перезапуску гра повинна виконувати дві дії:

1. Скидати стан гравця. Це включає і приховування кнопки перезапуску з екрана.
2. Повторно створювати тунелі. Таким чином, гравець стартує з початку тунелю.

Скидання гравця

Відкриємо BP_Player та створимо нову функцію RestartGame.

Створимо наступний граф:



1. Set Is Dead надає IsDead значення false. Це знову включає можливість руху вперед.
2. Remove From Parent видаляє з екрана RestartWidget
3. Set Input Mode Game Only знову включає можливість керування у грі, щоб гравець міг переміщатися
4. Set Show Mouse Cursor приховує курсор миші

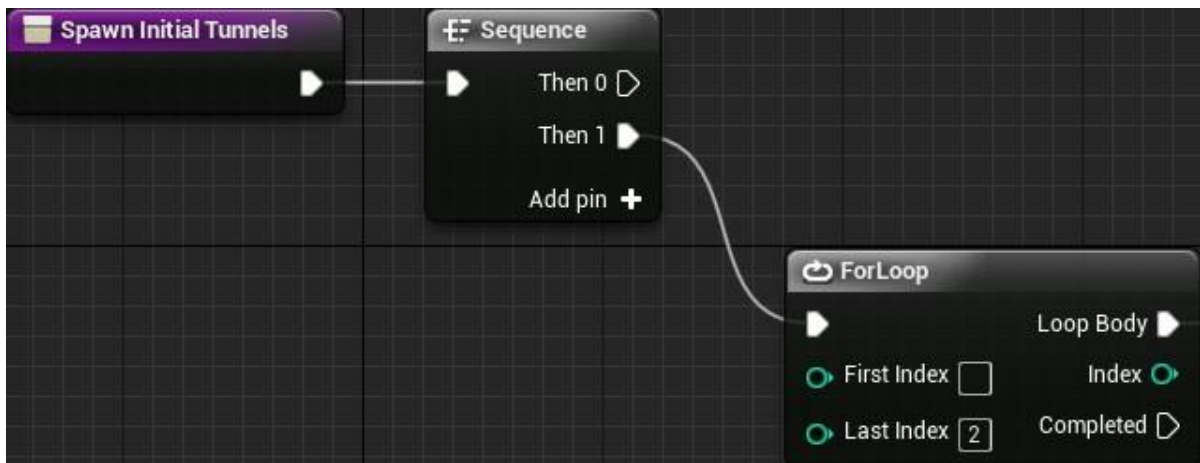
Повторне створення тунелів

Тиснемо Compile і закриваємо BP_Player.

Відкриваємо BP_TunnelSpawner та переходимо у граф SpawnInitialTunnels.

По-перше, перед створенням нових тунелів нам потрібно видалити вже існуючі.

Додамо нод Sequence після нода Entry. З'єднаємо контакт Then 1 із нодом ForLoop.



Нод Sequence виконує свої виходи у послідовному порядку. Це відмінний спосіб упорядкувати граф вертикально.

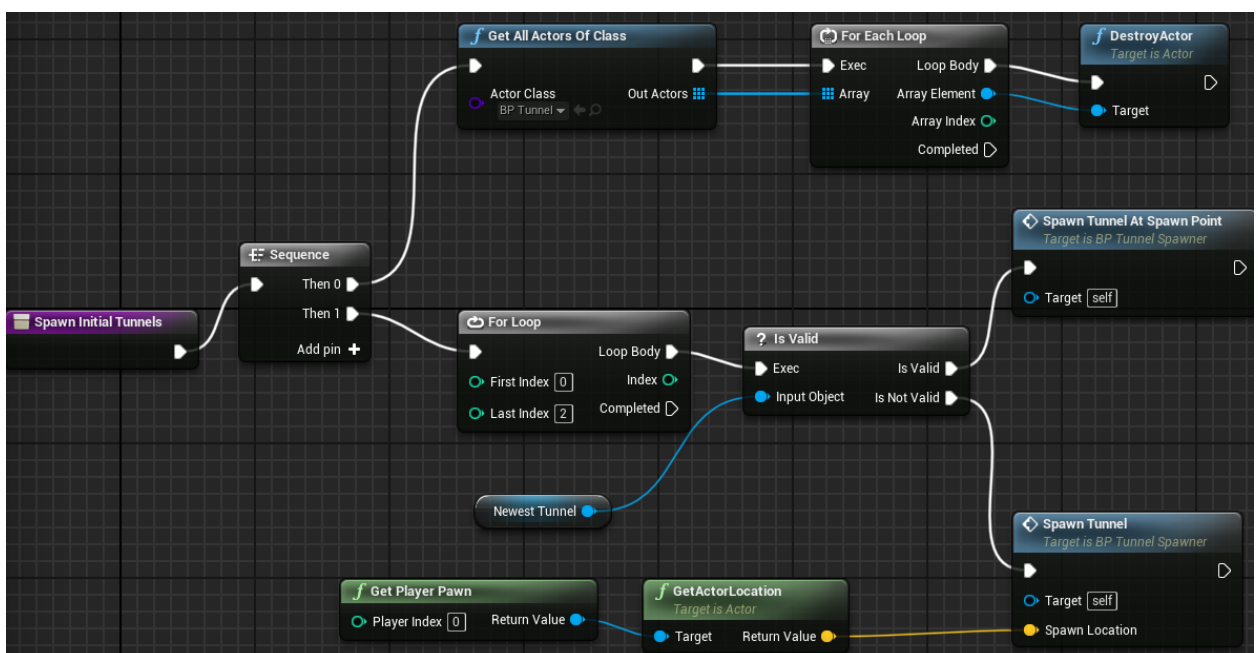
Потім створимо такі ноди:



Ця схема отримає всі існуючі тунелі та видалить їх із гри.

Нарешті, з'єднаємо контакт Then 0 нода Sequence з нодом Get All Actors of Class. Завдяки цьому тунелі видаляться до процесу створення.

Ось як виглядає кінцевий граф:

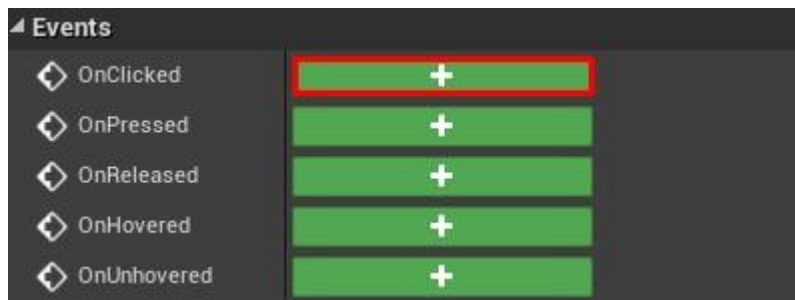


Обробка натискань на кнопку

Тиснемо Compile і закриваємо [BP_TunnelSpawner](#).

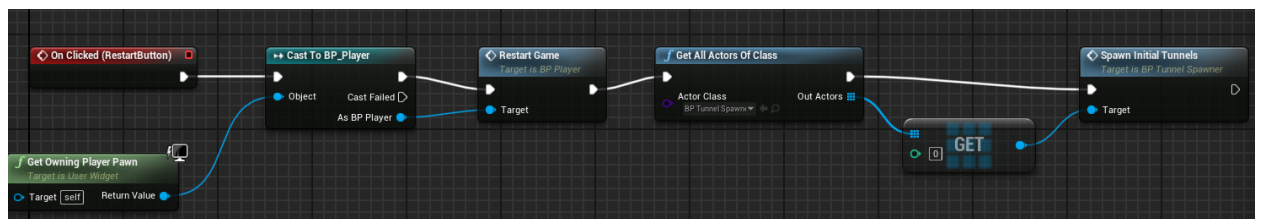
Перейдемо до Content Browser і знайдіть папку UI. Відкриємо WBP_Restart.

Виберемо RestartButton та перейдемо до панелі Details. Перейдемо до розділу Events і натиснемо кнопку поряд з OnClicked



Це створить нод On Clicked (RestartButton). Цей нод виконується, коли гравець натискає RestartButton.

Створюємо наступну схему:



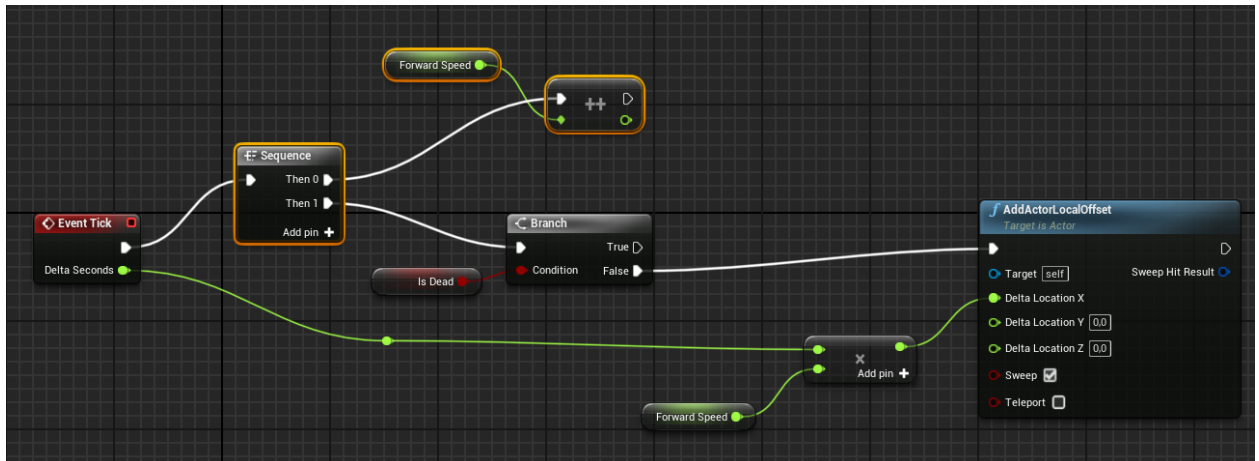
1. Get Owing Player Pawn повертає Pawn, яким зараз управляє гравець
2. Cast to BP_Player перевіряє, чи належить Pawn до класу BP_Player
3. Якщо це так, він викликає функцію RestartGame. Ця функція скидає гравця та приховує кнопку перезапуску.
4. Get All Actors of Class і Get повертає BP_TunnelSpawner і викликає SpawnInitialTunnels. Ця функція видаляє всі існуючі тунелі та спаунить нові.

Тиснемо на Compile та переходимо у основний редактор.

Поступове прискорення руху гравця

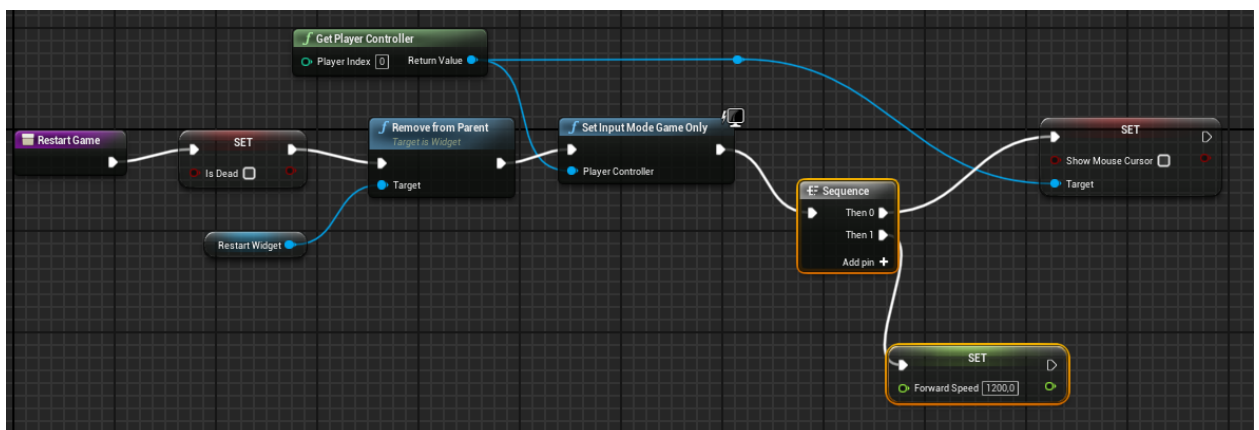
Зараз гра здається дуже легкою, тому ми зробимо поступове прискорення гравця для ускладнення гри.

Для цього переходимо у BP_Player та редагуємо граф Event Graph таким чином:



1. Sequence – після виконання ноду Event Tick по черзі виконуються ноди з вітки Then 0 та Then 1
2. Increment Float – додає до змінної Forward Speed одну одиницю, таким чином збільшуючи швидкість руху кожен тик на 1

Після рестарту гри швидкість руху залишається такою ж, як у попередній ігровій спробі, щоб це виправити, у функції Restart Game вносимо наступні зміни:



SET Forward Speed після рестарту гри задає вказане значення змінної, тобто змінює швидкість. Ми виставляємо значення, яке було у змінної на момент її створення.

Тиснемо на Compile та переходимо у основний редактор. Натиснувши кнопку Play можна грати.

4. Економічна частина

4.1 Резюме

В даному дипломному проекті розроблена кроссплатформна гра на ігровому двигуні Unreal Engine 4 з використанням Blue prints.

Ефективність кожного програмного продукту визначається його якістю та ефективністю процесу розробки. Якість ПП визначається наступними складовими: з точки зору користувача; з позиції використання ресурсів; виконання вимог до програмного забезпечення.

Оцінка якості програмного продукту з точки зору користувача визначається необхідним на стадії функціонування розміром оперативної пам'яті ЕОТ, витратами машинного часу, пропускнуою спроможністю каналів передачі даних. Оцінка якості програмного продукту включає визначення трудомісткості і вартості його створення.

					КГ 05.19.000 ДП ПЗ	Лист
						53
Изм.	Лист	№ докум.	Подпись	Дата		

4.2. Визначення трудомісткості розробки програмного забезпечення.

Тривалість розробки програмного продукту залежить від його обсягу, трудомісткості розробки, кваліфікації виконавців, а також планових термінів, визначених умовами ринку. Методом структурної аналогії по відповідних каталогах аналогів програмного забезпечення визначається обсяг програмних засобів, у тисячах умовних машинних команд програми аналога

Каталог аналогів

Таблиця 4.1

Найменування ПП	Обсяг функції ПП – V_o , усл. машинних командах.
1. ПП автоматизованих розрахунків	1300 – 8600
3. ПП введення інформації	1060 – 5750
7. ПП імітаційного моделювання	7800 – 8800

У таблиці 4.1 представлені аналоги програмного забезпечення, функції яких, у більшому або меншому ступені, виконує розроблений програмний продукт. Для нашого варіанта виділено сірим кольором. Вибравши аналог ПП, що містить V_o в умовних машинних командах, трудомісткості визначати на основі табл.4.2

Таблиця.4.2

Обсяг ПП, тис.умов.машинних команд	Норма часу, люд/год
1.00	229
2.00	244
3.00	262

На підставі отриманого значення, по довіднику, визначається укрупнена норма часу на розробку аналога програмного забезпечення (коректується поправочним коефіцієнтом враховуючої умови розробки ПП, тобто в умовах комп'ютера, $K_k=0,7\div 0,8$): $T^a = 229 \times 0,8 = 183,2$ (люд/годин).

Трудовісткість програмного продукту визначається по кожному етапу розробки окремо на підставі трудовісткості аналога з урахуванням складності

розробки, ступеня новизни і ступеня використання в розробці стандартних модулів на підставі формул:

$$T_{T3} = T^a p \times L_1 \times K_H \quad (4.1)$$

$$T_{TP} = T^a p \times L_2 \times K_H \quad (4.2)$$

$$T_{PP} = T^a p \times L_3 \times K_H \times K_T \quad (4.3)$$

Для розрахунку необхідні наступні коефіцієнти:

L_i – питома вага i -го етапу розробки (див. табл. 4.2.);

K_H – поправочний коефіцієнт, що враховує ступінь новизни (див. табл. 4.3.);

K_T – поправочний коефіцієнт, що враховує ступінь використання в розробці типових програм (див. табл. 4.4.).

Таблиця 4.2.Значення питомих коефіцієнтів трудовісткості стадії в загальній трудовісткості розробки ПП.

Код стадії	Ступінь новизни		
	А	Б	В
ТЗ (L_1)	0,15	0,12	0,12
ТП (L_2)	0,16	0,15	0,11
РП (L_3)	0,55	0,58	0,61

Для нашого варіанта виділено сірим кольором.

Таблиця 4.3. Значення поправочного коефіцієнта, що враховує ступінь новизни

Код ступеня новизни	Ступінь новизни	Значення K_n
А	Принципово нові ПП	1,75 – 1,2
Б	ПП – розвиток визначеного параметричного ряду	1,0 – 0,8
В	ПП маючий аналог	0,7

Для нашого варіанта виділено сірим кольором.

Таблиця 4.4. Значення коефіцієнта ступеня використання в розробці типових програм

Ступінь охоплення реалізованих функцій розроблювального ПП типовими програмами, %	Значення K_T
60 і вище	0,6
40-60	0,7
20-40	0,8
До 20	0,9

Для нашого варіанта виділено сірим кольором.

Тепер розраховуємо трудомісткість по кожному етапу окремо:

Трудомісткість технічного завдання

$$T_{T3} = T^a * L_1 * K_n = 183,2 * 0,12 * 0,7 = 15,4 \quad (\text{люд/годин}) \quad (4.1)$$

Трудомісткість розробки технічного проекту

$$T_{Tn} = T^a * L_2 * K_n = 183,2 * 0,11 * 0,7 = 14,1 \quad (\text{люд/годин}) \quad (4.2)$$

Трудомісткість розробки робочого проекту

$$T_{pp} = T^a * L_3 * K_H * K_T = 183,2 * 0,61 * 0,7 * 0,7 = 54,8 \quad (\text{люд/годин}) \quad (4.3)$$

Для подальших розрахунків визначили кількість папера, витраченого на кожен етап: технічне завдання $N_{ТЗ} = 2$ (стр), розробка ТП $N_{ТП} = 13$ (стр), розробка робочого проекту $N_{рп} = 22$ (стр), пояснювальна записка відповідно $N_{пз} = 8$ (стр) Розрахунок зведений у таблицю 4.5

Таблиця 4.5. Розрахунок трудомісткості ПП

Найменування етапів	Розрахунок, годин.		
1	2	3	4
1.ТЗ	$T_{PTЗ} = 15,4$	$T_{KK} = 0,7 * N_{ТЗ} = 1,4$	$T_{HK} = 0,15 * N_{ТЗ} = 0,3$
2.Розробка ТП	$T_{PTП} = 14,1$	$T_{KK} = 0,7 * N_{ТП} = 9,1$	$T_{HK} = 0,15 * N_{ТП} = 2$
3.Розробка РП	$T_{PpП} = 54,8$	$T_{KK} = 0,7 * N_{рп} = 15,4$	$T_{HK} = 0,15 * N_{рп} = 3,3$
4.Розробка ПЗ	$T_{ПЗ} = 1,5 * N_{ПЗ} = 12$	$T_{KK} = 0,7 * N_{ПЗ} = 5,6$	$T_{HK} = 0,15 * N_{ПЗ} = 1,2$
Усього, в т.ч.:	$\Sigma T = 134,6$		
- на розробку	$\Sigma T_p = 96,3$		
- контроль керівника		$\Sigma T_{KK} = 31,5$	

- нормоконтроль			$\Sigma T_{\text{нк}} = 6,8$
--------------------	--	--	------------------------------

4.3 Розрахунок ціни програмного продукту.

У цьому розділі для визначення ціни розраховуємо основну заробітну плату виконавців, матеріальні витрати, вартість машино – години і витрати на розробку ПО. Розрахунок основної заробітної плати виконавців приведений у таблиці 4.6. Відповідно до статті 8 «Закону про Державний бюджет України на 2022» встановлено мінімальну заробітну плату у місячному розмірі з 1 січня 2022 року - 6500 гривень; мінімальну погодинну тарифну ставку – 39.26 грн.

Таблиця 4.6 Розрахунок основної заробітної плати виконавців.

Найменування робіт	Трудовісткість робіт, години	Погодинна тарифна ставка, грн.	Розрахунок, грн.
1.Розробка ПП	96,3	39.26	3781
2.Контроль керівника	31,5	50.00	1575
3.Нормоконт-роль	6,8	50.00	340
Усього	-	-	$\Sigma Z_o = 5696$

Зробимо розрахунок матеріальних витрат на розробку ПП. Розрахунок зведемо в таблицю 4.7

Таблиця 4.7 Розрахунок матеріальних витрат на розробку ПО

Найменування	Тип,	Кількість	Ціна	Вартість, грн.
--------------	------	-----------	------	----------------

					КГ 05.19.000 ДП ПЗ				Лист
Изм.	Лист	№ докум.	Подпись	Дата					58

матеріальних витрат	модель		одиниці, грн.	
Папір	Лист А4	37	2	
				$B_{Mi} = 74$
Транспортно–заготівельні витрати (10%)				$B_{тр_з} = 0,1 \times B_{M1} = 7,4$
Усього				$B_M = B_{Mi} + B_{тр_з} = 81,4$

На підставі отриманих даних по окремих статтях витрат складена калькуляція планової собівартості в цілому ПП за формою, приведеною в таблиці 4.8.

Таблиця 4.8. Розрахунок статей витрат планової собівартості

Стаття витрат	Значення, грн.	Формула розрахунку
1. Матеріали	81,4	B_M (див. табл. 4.7)
2. Основна заробітна плата	5696	$З_o$ (див. табл. 4.6)
3. Додаткова заробітна плата	854,4	$З_d = 0,15 \times З_o$
4. Відрахування до єдиного фонду соціального внеску	1441,1	$В_{\epsilon.c.в.} = 0,22 \times (З_o + З_d)$
5. Накладні витрати	1708,8	$В_{нак.} = 0,3 \times З_o = 0,3$
6. Повна собівартість	9781,7	$C_{пов} = B_M + З_o + З_d + В_{\epsilon.c.в.} + В_{нак.} =$

Розмір прибутку, що включається в ціну, визначаємо по наступній формулі:

$$П = (C_{пов} \cdot P) / 100 = 978,2 \text{ грн} \quad (4.4)$$

Де P – плановий рівень рентабельності (10-15%).

Оптова ціна (кошторисна вартість) визначається по формулі:

$$\text{Ц}_0 = \text{C}_{\text{пов}} + \text{П} = 10759,9 \text{ грн.} \quad (4.5)$$

Податок на додану вартість визначаємо по наступній формулі:

$$\text{ПДВ} = 0.2 * \text{Ц}_0 = 2152 \text{ грн} \quad (4.6)$$

Виходячи з отриманих даних, ціна реалізації розробленого програмного продукту на основі наступної формули, становитиме:

$$\text{Ц}_p = \text{Ц}_0 + \text{ПДВ} = 12911,9 \text{ грн} \quad (4.7)$$

					КГ 05.19.000 ДП ПЗ	Лист
						60
Изм.	Лист	№ докум.	Подпись	Дата		

5. Охорона праці

Вступ

Через масовий характер робіт, що виконуються працівниками за допомогою комп'ютера, законодавством України чітко врегульовано норми та вимоги до використання комп'ютерної техніки на підприємстві, безпосередньо і охорона праці при роботі з комп'ютером.

В даному розділі дипломного проекту розглядається питання безпеки робочого місця працівників, що працюють за комп'ютером.

5.1 Аналіз небезпечних та шкідливих чинників, що впливають на працівника.

Працівники зіштовхуються із впливом таких фізично небезпечних і шкідливих виробничих факторів, як підвищена температура зовнішнього середовища, низька вологість повітря, відсутність або недостатня освітленість робочої зони, електричний струм, статична електрика та малорухомість.

Під час роботи за комп'ютером працівник проводить багато часу в сидячому, майже нерухомому положенні, що може призвести до порушення кровообігу та проблемам зі спиною.

Перебування довго в одній позі в будь-якому випадку призводить до постійного навантаження на одні групи м'язів та відсутність його на інші.

Відсутність навантаження викликає слабкість м'язів, внаслідок чого з'являється викривлення хребта (сколіоз), а також руйнування міжхребцевих дисків, що називається остеохондрозом. Крім того, навантаження на міжхребцеві диски, яке виникає в сидячому положенні, набагато більше, ніж стоячи або лежачи.

					КГ 05.19.000 ДП ПЗ	Лист
						61
Изм.	Лист	№ докум.	Подпись	Дата		

Відповідно до регламенту охорони праці при безперервній роботі за комп'ютером кожен годину потрібно проводити невелику паузу. В цей час можна встати з-за столу, розім'яти спину, зробити обертальні рухи шиєю, попереком, суглобами рук та ніг.

Кращою профілактикою викривлень хребта та порушення кровообігу є рухливий спосіб життя, який повинен супроводжуватися постійним стеженням за власною поставою та положенням тіла, в якому працівник проводить велику частину свого часу.

На робочому місці повинні бути створені умови для безпечної та високопродуктивної праці.

5.1.1 Мікроклімат робочої зони працівників, вентиляція.

У виробничих приміщеннях на робочих місцях мають забезпечуватись оптимальні значення параметрів мікроклімату: температури, відносної вологості й рухливості повітря – ГОСТ 12.1.005-88, СН 4088-86.

Параметри мікроклімату	Значення параметрі	значення параметрі
	Взимку	влітку
Температура, С ⁰	22-24	23-25
Відносна вологість, %	40-60	40-60
Швидкість руху повітря, м/с	0,1	0,1-0,2

Для підтримки в приміщеннях нормального, що відповідає гігієнічним вимогам, складу повітря, видалення з нього шкідливих газів, пилу використовують вентиляцію. Механічна вентиляція (кондиціонери, вентилятори і т.д.) залежно від напрямку руху повітряних потоків, може бути витяжною, припливною і припливно-витяжною. При природній вентиляції (за допомогою вікон) повітря надходить у приміщення і видаляється з нього внаслідок різниці температур і тиску.. Механічна вентиляція забезпечується

вентиляторами, що забирають повітря зовні і направляє його до будь-якого робочого місця, або устаткування, а також видаляють забруднене повітря.

5.1.2 Освітлення робочого місця, шум, вібрація

Приміщення для роботи з ВДТ повинні мати природне та штучне освітлення. У приміщеннях, призначених для роботи з ПК, доцільно, щоб вікна були орієнтовані на північ або північний захід. На вікнах повинні бути штора або жалюзі, що регулюють рівень освітленості і захищають від прямого влучення сонячних променів на робоче місце.

Для штучного освітлення у приміщенні використовуються люмінесцентні лампи типу ЛБ.

У приміщеннях з ЕОМ рівні звукового тиску, рівні звуку та еквівалентні рівні звуку на робочих місцях повинні відповідати вимогам ГОСТ 12.1.003 ССБТ "Шум. Общие требования безопасности", СН 3223-85 "Санітарні норми допустимих рівнів шуму на робочих місцях", затверджених Міністерством охорони здоров'я СРСР, ГР N 2411-81 "Гігієнічні рекомендації по встановленню рівнів шуму на робочих місцях з урахуванням напруженості та тяжкості праці", затверджених Міністерством охорони здоров'я України

5.1.3 Електробезпека.

Використання електричної енергії на виробництві пов'язане з небезпекою дії електроструму на організм людини.

Ураження струмом може виникнути при роботі під напругою і при несправному стані електроустановок, а саме при дотику до оголених проводів, незаземлених металевих корпусів електричного обладнання, при відкритих рубильниках і других струмоведучих частинах.

Для захисту працюючих від ураження електричним струмом передбачені наступні заходи:

- недоступність струмоведучих частин;
- захисне заземлення (занулення) корпусів електрообладнання;

					КГ 05.19.000 ДП ПЗ	Лист
						63
Изм.	Лист	№ докум.	Подпись	Дата		

- передбачені рубильники закритого типу;
- блокування, надписи, плакати, засоби індивідуального захисту

У приміщенні, де одночасно експлуатуються понад п'ять ЕОМ, на помітному та доступному місці встановлюється аварійний резервний вимикач, який може повністю вимкнути електричне живлення приміщення, крім освітлення. ЕОМ з ВДТ і ПК повинні підключатися до електромережі тільки за допомогою справних штепсельних з'єднань і електророзеток заводського виготовлення. У штепсельних з'єднаннях та електророзетках, крім контактів фазового та нульового робочого провідників, мають бути спеціальні контакти для підключення нульового захисного провідника. Їхня конструкція має бути такою, щоб приєднання нульового захисного провідника відбувалося раніше, ніж приєднання фазового та нульового робочого провідників. Порядок роз'єднання при відключенні має бути зворотнім.

Не допускається підключати ЕОМ з ВДТ і ПК до звичайної двопровідної електромережі, в тому числі – з використанням перехідних пристроїв.

5.2 Організація робочого місця

При розміщенні робочих місць з персональними комп'ютерами відстань між робочими столами з моніторами (у напрямі тилу поверхні одного відеомонітора і екрану іншого відеомонітора) повинно бути не менше 2,0 м, а відстань між бічними поверхнями відеомоніторів - не менше 1,2 м.

При виконанні творчої роботи, що вимагає значного розумового напруження або високої концентрації уваги, робочі місця з персональними комп'ютерами рекомендується ізолювати один від одного перегородками висотою 1,5-2,0 м.

Екран відеомонітора повинен знаходитися від очей користувача на відстані 600-700 мм, але не ближче 500 мм з урахуванням розмірів алфавітно-цифрових знаків і символів.

Конструкція робочого столу повинна забезпечувати оптимальне розміщення на робочій поверхні використовуваного обладнання з урахуванням його

					КГ 05.19.000 ДП ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		64

кількості і конструктивних особливостей, характеру виконуваної роботи. При цьому допускається використання робочих столів різних конструкцій, що відповідають сучасним вимогам ергономіки. Поверхня робочого столу повинна мати коефіцієнт відбиття 0,5-0,7.

Висота робочої поверхні столу для дорослих користувачів повинна регулюватися в межах 680-800 мм; при відсутності такої можливості висота робочої поверхні столу повинна складати 725 мм.

Робочий стіл повинен мати простір для ніг висотою не менше 600 мм, шириною - не менше 500 мм, глибиною на рівні колін - не менше 450 мм, на рівні витягнутої ноги - не менше 650 мм.

Конструкція робочого стільця (крісла) повинна забезпечувати підтримку раціональної робочої пози під час роботи на персональному комп'ютері, дозволяти змінювати позу з метою зниження статичного напруження м'язів шейноплечової області і спини для попередження розвитку втоми. Тип робочого стільця (крісла) слід вибирати з урахуванням зростання користувача, характеру та тривалості роботи з ПК.

5.3 Пожежна безпека

Для профілактики пожеж у приміщеннях з ПК та іншою електричною технікою слід уникати перевантаження електромережі.

При використанні потужних споживачів електроенергії вони повинні вмикатися лише до розеток, які розраховані на таке навантаження. Перевантаження електромережі може призвести до нагрівання дрота та його оплавлення, що може призвести до пожежі.

Протипожежний захист приміщення забезпечується застосуванням автоматичної установки пожежної сигналізації, наявністю засобів пожежогасіння, застосуванням основних будівельних конструкцій будинку з регламентованими межами вогнестійкості, організацією своєчасної евакуації людей.

					КГ 05.19.000 ДП ПЗ	Лист
						65
Изм.	Лист	№ докум.	Подпись	Дата		

Приміщення повинні бути оснащені вуглекислотними або порошковими вогнегасниками. У випадку виникнення пожежі необхідно відключити електроживлення, викликати пожежну команду за номером 101, евакуювати людей із приміщення відповідно до плану евакуації і приступити до ліквідації пожежі.

					КГ 05.19.000 ДП ПЗ	Лист
						66
Изм.	Лист	№ докум.	Подпись	Дата		

Висновки

У сучасному світі майже будь яка людина має можливість створити гру. Сучасні ігрові двигуни йдуть по шляху спрощення розробки. Звісно, розробка вимагає деяких навичок, вмінь та часу, і чим більше вони, тим більш велику та складну гру можна створити. Такі інструменти як мова візуального скриптіngu дозволяють навіть не програмістам створювати код для гри і вимагають менше часу на навчання.

Для прикладу була створена невелика кросплатформна гра на двигуні Unreal Engine 4 з використанням мови візуального скриптіngu Blue prints.

					КГ 05.19.000 ДП ПЗ	Лист
						67
Изм.	Лист	№ докум.	Подпись	Дата		

Списко використаних джерел

1. Що означає кросплатформність для ігор та інших програм – (Електронний ресурс) - <https://cpab.ru/chto-oznachaet-krossplatformennost-dlya-igr-i-drugih-prilozhenij/>
2. Крос-платформна гра (Електронний ресурс) - https://hmong.ru/wiki/Cross-platform_play
3. Тьюторіал з Unreal Engine 4 (Електронний ресурс) - <https://habr.com/ru/post/344394/>
4. Unreal Engine, матеріал з вікіпедії (Електронний ресурс) - https://ru.wikipedia.org/wiki/Unreal_Engine
5. Офіційний сайт Unreal Engine (Електронний ресурс) - <https://www.unrealengine.com/en-US>
6. Бойчик І. М. Економіка підприємства : навчальний посібник для студентів економічних спеціальностей вищих навчальних закладів І-ІV рівнів акредитації. Третє видання, випр. і доп. / І. М. Бойчик, П. С. Харів., М. І. Холчан, Ю. В. Піча. – К. : Каравела, 2016. – 328 с.
7. Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин (Електронний ресурс) - <https://zakon.rada.gov.ua/rada/show/v0007282-98#Text>
8. Вимоги до рівнів шуму та вібрації (Електронний ресурс) - https://vuzlit.com/135897/vimogi_rivniv_shumu_vibratsiyi