МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
МАРІУПОЛЬСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

ЕКОНОМІКО-ПРАВОВИЙ ФАКУЛЬТЕТ
КАФЕДРА СИСТЕМНОГО АНАЛІЗУ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

До захисту допустити:
Завідувач кафедри
_____ _____
(підпис) (ПІБ завідувача кафедри)
«__» _____ 20__р.

«ГЕЙМДЕВ ТА КУЛЬТУРА ВІДЕОІГОР»

Кваліфікаційна робота
здобувача вищої освіти першого
рівня вищої освіти
освітньо-професійної програми

«_____»
(назва освітньо-професійної програми)

Гівчак Катерина Леонідівна
(прізвище, ім'я по батькові здобувача вищої освіти)

Науковий керівник:
Мартинюк Г. В., доцент кафедри
засобівзахисту інформації, кандидат
технічних наук

Рецензент:
_____
(прізвище, ініціали, науковий ступінь, вчене звання, місце роботи)

Кваліфікаційна робота захищена
з оцінкою_____
Секретар ЕК_____
«____» 20__ р.

Маріуполь – 2023

ЗМІСТ

ВСТУП

У сучасному світі відеоігри втілилися в справжню культуру, визнану мільйонами людей по всьому світу. Ця культура проникає у всі сфери нашого життя, включаючи медіа, розваги, освіту і навіть мистецтво. Вона стала великою галуззю, що виробляє значні прибутки і залучає талановитих професіоналів. Однією з ключових граней цієї галузі є геймдев, який відповідає за розробку відеоігор.

Вони стали невід'ємною частиною сучасної культури і індустрії розваг, залучаючи мільйони гравців з усього світу. У сучасному світі відеоігри є не тільки розважальним заняттям, але й важливою складовою культури, яка сильно впливає на суспільство та має значний вплив на різні аспекти нашого життя, стають засобом комунікації, навчання та впливають на формування цінностей та ідеології. Становлення та розвиток індустрії геймдеву та відеоігор заслуговує на увагу та дослідження, особливо з точки зору програміста.

Геймдев - це складний процес, що вимагає поєднання технічних знань і творчого підходу. Від програмістів вимагається не лише вміння створювати функціональні програми, але й здатність розуміти унікальні вимоги геймплею, взаємодії з гравцем і створювати захоплюючі світи. В цьому контексті роль геймдеву, або розробки відеоігор, стає критично важливою. Це комплексна діяльність, яка включає в себе розробку концепції ігри, програмування, дизайн геймплею, створення графіки, звуку, анімації та інші аспекти, що необхідні для створення повноцінного відеоігрового досвіду, що здатний втілювати новаторські ідеї, розвивати технології та надихати гравців

Геймдев є складним і творчим процесом, який вимагає від програмістів глибоких знань у сфері програмування, графіки, фізики та штучного інтелекту. Вони використовують різноманітні мови програмування,

інструменти та фреймворки для створення реалістичного геймплею, візуальних ефектів, штучного інтелекту персонажів та багато іншого. Для програміста надзвичайно важливо розуміти взаємозв'язок між геймдевом та культурою відеоігор. Навички та знання в області програмування впливають на створення віртуальних світів, персонажів, геймплею та технічні аспекти гри, що безпосередньо впливає на досвід гравців та сприяє формуванню культурних цінностей. Розробка відеоігор з урахуванням культурних аспектів вимагає етичності, чутливості до різних культур та вміння відображати диверситет у геймерському середовищі.

Процес геймдеву зазвичай починається з формулювання ідеї гри та розробки концепції. Потім команда розробників (яка може складатися з програмістів, дизайнерів, художників, музикантів та інших спеціалістів) приступає до розробки самої гри. Це включає програмування геймплею, створення графічних ефектів, моделей персонажів, обробку звуку та музики, тестування та налагодження. Це творчий процес, який поєднує технічну експертизу з художнім баченням. Він вимагає розуміння геймплею, взаємодії гравця з віртуальним світом, а також урахування потреб і очікувань цільової аудиторії. Геймдев може займати тривалі періоди часу і вимагати співпраці кількох спеціалістів для створення високоякісних ігор.

Багато ігор включають елементи культури, історії, мистецтва та соціальних проблем, що стимулює гравців до критичного мислення та розуміння світу навколо них.

По мірі розвитку галузі геймдеву з'являються нові виклики та можливості для програмістів. Ігрові платформи розширюються, реалізуються інтерактивність, віртуальна реальність та розширена реальність. Геймдев також впливає на інші сфери, такі як освіта, медицина, симуляції та тренування. Розуміння цих тенденцій і викликів, з якими стикаються

програмісти в галузі геймдеву, є надзвичайно важливим для успішної реалізації ігрових проектів.

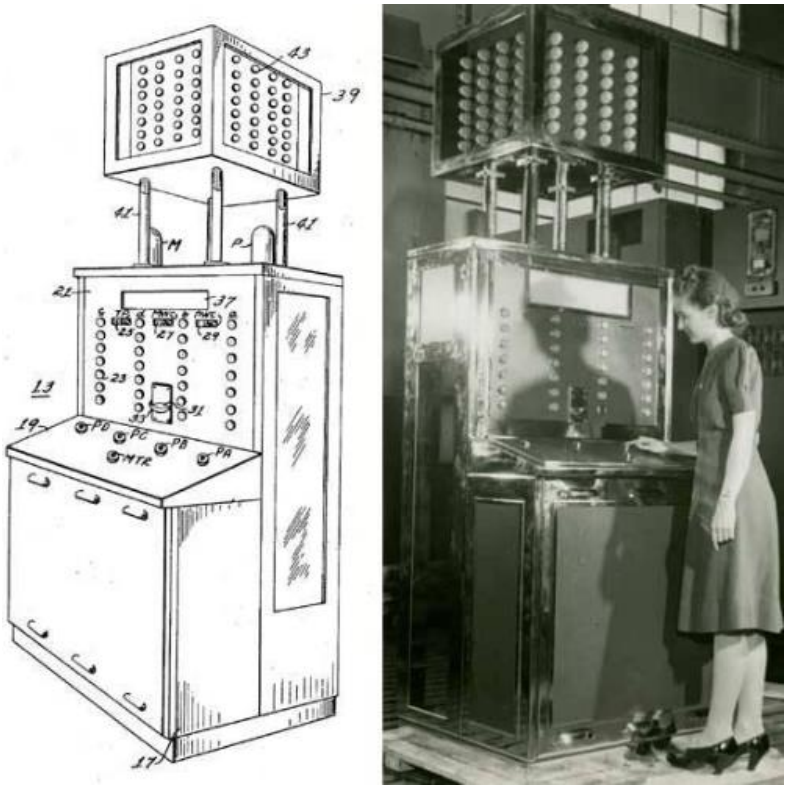
Ця дипломна робота присвячена дослідженню впливу геймдеву та культури відеоігор. Вона має на меті розкрити значення та роль програмістів у створенні відеоігор, а також їх вплив на сучасну культуру.

РОЗДІЛ І. АНАЛІЗ ІГРОВОЇ ІНДУСТРІЇ У СВІТІ

1.1 Історія відеоігор

Відеоігри — це надзвичайно популярна форма розваги, яка викликає все більший інтерес у сучасному світі. За останні десятиліття відеоігри відіграли значну роль у розвитку технологій та культурного ландшафту. Вони стали справжньою формою мистецтва, в якій поєднується графіка, музика, сюжет та інтерактивність. Ця робота пропонує огляд історії відеоігор, починаючи з їх появи і розвитку до сучасних досягнень та впливу на сучасну культуру. Історія відеоігор є багатогранною та цікавою, розпочавшись зі свого виникнення в другій половині 20 століття. Вона пов'язана з низкою ключових подій, розробниками та технологічними проривами, що допомогли сформувати цю надзвичайно популярну галузь розваг.

1940 рік був другим роком Другої світової війни, яка тривала з 1939 по 1945 роки. Це був період інтенсивного розвитку науки і техніки, що сприяв появі перших комп’ютерів і відеоігор. В 1940-х роках деякі вчені та інженери почали експериментувати зі створенням електронних ігор. Перша ігрова комп’ютерна програма була написана у 1940 році американським фізиком Едвардом Кондоном. Вона симулювала гру в нім - логічну гру, в якій два гравці по черзі беруть камені з купок, а той, хто забере останній камінь, програє. Програма була реалізована на електронно-релейній машині Nimatron, яка була продемонстрована на Всесвітній виставці в Нью-Йорку. Nimatron вважається першим комп’ютером, призначеним для розваг, і претендентом на статус першої комп’ютерної гри.



(Рис. 1.1 Електронно-релейна машина Nimatron)

У 1952 році британський математик Александр Дуглас створив першу графічну комп’ютерну гру OXO (також відому як Noughts and Crosses), що симулювала гру «хрестики-нолики» на електронно-цифровому комп’ютері EDSAC. Гра була написана як частина його дисертації про взаємодію людини і комп’ютера і була призначена для демонстрації можливостей комп’ютера. Гравець мав вводити свої ходи за допомогою телефонної клавіатури, а комп’ютер показував результат на катодно-променевому екран.

У 1958 році американський фізик Вільям Хігінботам створив першу інтерактивну відеогру Tennis for Two (Теніс для двох), що симулювала гру в теніс на осцилографі. Гра була розроблена як атракціон для відвідувачів лабораторії Brookhaven National Laboratory, де Хігінботам працював. Гравці керували своїми ракетками за допомогою двох коробок з кнопками і поворотними ручками. Гра була дуже популярною серед відвідувачів, але не була комерційно розповсюджена.

У 1961 році студенти Массачусетського технологічного інституту (MIT) Стів Рассел, Мартин Грац і Вейн Вайттен створили першу відеогру для комп'ютера PDP-1 - Spacewar! (Космічна війна!). Це була двомірна гра про космічний бій між двома космонавтами, які стріляли один в одного з лазерних пушок і намагалися уникати тяжкості зоряного ядра. Гра була написана на мові програмування FLOW-MATIC і використовувала графічний дисплей комп'ютера. Гра швидко поширилася серед студентської спільноти MIT і інших університетів, де були доступні комп'ютери PDP-1. Spacewar! вважається першою цифровою відеогрою і одним з найважливіших досягнень у історії відеоігор. Гра привернула увагу інших розробників, що почали робити свої власні ігри, використовуючи подібні концепції та технології.

Наслідком "Spacewar!" було зростання зацікавленості до відеоігор як розважальної форми. Вона відкрила двері для подальшого розвитку галузі, з'явлення аркадних ігор та першої домашньої консолі, що набули популярності у 1970-х та 1980-х роках.
"Spacewar!" також вплинула на подальший дослідницький напрямок відеоігор. Багато розробників та студій використовували цю гру як вихідний пункт для експериментів з графікою, фізикою та ігровим дизайном.

Справжній прорив відбувся в 1970-х, коли було створено перші аркадні ігри, такі як "Pong" і "Space Invaders". Ці прості, але захоплюючі ігри сприяли популяризації відеоігор серед широкої аудиторії. У 1972 році Бушнелл і Дабні заснували компанію Atari і випустили першу популярну аркадну відеогру Pong (Понг), що симулювала гру в настільний теніс. Гра була простою у керуванні і мала чорно-білий дисплей. Гра мала два режими: для одного або двох гравців. Гравці керували вертикальними платформами за допомогою поворотних ручок і намагалися вдарити м’яч так, щоб суперник не зміг його повернути. Гра стала дуже популярною і принесла Atari великий прибуток. Pong вважається першою масовою відеогрою і тим, що

започаткувало золотий вік аркадних відеоігор.
А згодом у 1975 році Atari випустила першу домашню версію гри Pong, яка підключалася до телевізора. Гра була продана в магазинах Sears і стала першим масовим хітом серед домашніх відеоігор. Гра була також першою грою, що використовувала мікросхеми інтегральних схем, що дозволило зменшити розмір і ціну пристрою.

У 1979 році американська компанія Activision стала першою незалежною компанією-розробником відеоігор для домашнього ринку. Компанія була заснована колишніми співробітниками Atari, які були незадоволені умовами праці і відсутністю авторських прав на свої твори.

У 1980 році японська компанія Namco випустила аркадну відеогру Pac-Man (Пак-Мен), яка стала одним з найвпливовіших та найпопулярніших творів у світовому мистецтві. Гра була першою грою, що мала персонажа з характером і особистістю. Гра також була першою грою, що мала складний штучний інтелект для ворогів, якими були чотири привиди з різними стратегіями. Гра стала культурним феноменом і породила багато наслідків, пародій і мерчандайзингу.
У цих роках графічні можливості відеоігор почали стрімко зростати. З'явилися перші домашні комп'ютери, такі як Commodore 64 та ZX Spectrum, які дозволяли грати в більш складні ігри з кольоровою графікою. В цей період з'явилися такі культові ігри, як "Super Mario Bros" та "The Legend of Zelda", які стали символами індустрії відеоігор.
"Super Mario Bros" була випущена в 1985 році і стала однією з найбільш впливових гри в жанрі платформерів. У грі гравець керує персонажем на ім'я Маріо, який повинен пройти рівні, перемігаючи ворогів, збираючи монети та виконуючи завдання, щоб врятувати принцесу Піч. "Super Mario Bros" встановила стандарти для майбутніх ігор в жанрі платформерів і стала культовою назвою в ігровій індустрії.

"The Legend of Zelda" була випущена в 1986 році і представляє собою серію пригодницьких рольових ігор. У грі гравець керує головним героєм на ім'я Лінк, який вирушає в епічну подорож по фантастичному світу Гідрюл, щоб врятувати принцесу Зельду та перемогти злого Короля Ганонда. "The Legend of Zelda" славиться своїм відкритим світом, головоломками та глибокою сюжетною лінією. Гра отримала велику популярність серед геймерів і стала однією з найуспішніших серій в історії відеоігор.
Обидві серії, "Super Mario Bros" і "The Legend of Zelda", дали початок багатьом продовженням та спін-оффам, а також стали символами компанії Nintendo. Персонажі Маріо та Лінк стали незмінними символами гри, а їх ігри отримали велику кількість нагород і визнання від критиків та гравців.

У 1983 році японська компанія Nintendo випустила свою першу домашню гральну консоль Famicom (Family Computer), яка була перейменована на Nintendo Entertainment System (NES) для міжнародного ринку. Консоль мала 8-бітний процесор і 16-колірну графіку. Консоль мала багато ексклюзивних ігор, таких як Super Mario Bros. (Супер Маріо Брати), The Legend of Zelda (Легенда про Зельду), Metroid (Метройд) і Final Fantasy (Файнал Фентези). Консоль допомогла врятувати індустрію в США після кризи 1983 року і стала найпопулярнішою консоллю свого покоління.

З появою 3D-графіки у 1990-х роках відеоігри зазнали ще одного революційного зрушення. Такі ігри, як "Doom" та "Quake", стали популярними завдяки своїм реалістичним 3D-світам та мультиплеєрному режиму. У цей період також з'явилися перші консолі нового покоління, такі як PlayStation та Nintendo 64, які забезпечили високу якість графіки та звуку. "PlayStation", так і "Nintendo 64" були першими консолями, які використовували повноцінну 3D-графіку для відтворення геймплею. Це дозволило створити нові ігрові світи з більш реалістичними об'єктами та оточенням. Поява 3D-графіки значно розширила можливості геймдевелоперів та змінила підходи до геймплею. "Nintendo 64" вперше

використовувала трійку рухомих джойстиків на своєму контролері, що дозволяло більшу свободу рухів та точність у керуванні персонажем. Це було революційно-інноваційним рішенням для контролерів консолей. У свою чергу, "PlayStation" впровадила CD-диски замість картриджів, що дозволило зберігати більшу кількість даних та забезпечило випуск більш грандіозних та амбітних ігор. Багато ігор стали культовими ігровими шедеврами та вирізнялися своїм геймплеєм, сюжетом та інноваційними рішеннями. Це додало велику цінність і привабливість до володіння цими консолями.

Поступово відеоігри стали більш складними та глибокими у своєму змісті. З'явилися ігри з глибокими сюжетами та персонажами, такими як "Final Fantasy VII" та "Metal Gear Solid", які викликали емоції та захоплення у гравців. Відеоігри почали розвивати не тільки навички реакції та швидкості, але й здатність до стратегічного мислення та вирішення проблем.
З'явлення масового Інтернету принесло ще більшу інтерактивність відеоігор. Мультиплеєрні онлайн-ігри, такі як "World of Warcraft" та "Fortnite", стали популярними через можливість взаємодії з іншими гравцями у віртуальних світах.

В останні роки відеоігри стали ще більш доступними та інноваційними. З'явилися віртуальна реальність (VR) та розширена реальність (AR), що змінило спосіб, якими ми взаємодіємо з відеоіграми. Приклади успішних ігор цього періоду включають "The Legend of Zelda: Breath of the Wild", "Fortnite", "Red Dead Redemption 2" та "The Last of Us Part II".
Відеоігри змінилися від простих аркадних ігор до складних, глибоких інтерактивних досвідів. Вони мають великий вплив на сучасну культуру, їх використовують як форму мистецтва, засіб розваги та навчання. Історія відеоігор продовжує розгортатися, і можна очікувати ще більш захоплюючих та інноваційних творів у майбутньому.

1.2 Культура відеоігор

Відеоігрова культура - це комплексний феномен, що охоплює різні аспекти взаємодії людини з відеоіграми та їх вплив на суспільство. Вона включає в себе різні практики, які гравці та розробники використовують для створення, споживання та аналізу відеоігор, такі як ігрова термінологія, історія відеоігор, кіберспорт, машиніма, летсплеї тощо.

Одним із проявів відеоігрової культури є практика швидкого проходження відеоігор - **спідрани** (англ. speedrunning), яка полягає у досягненні заданої мети в грі за мінімальний проміжок часу. Швидкісне проходження відеоігор може мати різну специфіку залежно від обраних гравцями правил, категорій та платформ. Ця практика також супроводжується формуванням спеціалізованих спільнот, які займаються реєстрацією, публікацією та аналізом результатів швидкого проходження відеоігор.
Методологія спідранив базується на розробці оптимального маршруту (англ. routing) - послідовності дій та етапів у грі, яка дозволяє пройти її за найменший час. Маршрут може включати пропуск одного або кількох важливих предметів або секцій гри, що називається порушенням послідовності (англ. sequence breaking). Для досягнення цього можуть використовуватися помилки (англ. glitches) у програмуванні гри, які дозволяють пропустити або швидше пройти деякі частини гри. Існують також інструментально підтримані спідрани (англ. tool-assisted speedruns), які використовують емуляційне програмне забезпечення та інструменти для створення теоретично ідеального проходження гри.
Швидкісне проходження відеоігор має свої корені ще в 1970-х роках, коли гравці змагалися за найвищий бал у аркадних іграх. З розвитком інтернету в 1990-х роках спідрани стали більш популярними та доступними, оскільки гравці могли обмінюватися своїми записами та порадами на форумах та веб-сайтах. Однією з перших ігор, яка стала предметом інтенсивного

спідранингу, була Doom (1993), яка мала вбудовану функцію запису та відтворення гри. З того часу спідрани поширилися на багато інших жанрів та платформ, створивши розмаїття категорій та правил.
Ігрові спідрани можна назвати кіберспортом, але це масове явище у ігровій індустрії більш грандіозне, ніж спідрани.

В культурі відеоігор є два популярні формати відеоконтенту - **летсплеї та стрімінг:**

- Летсплей - це відеоролик, в якому гравець показує та коментує свій ігровий процес у певній грі.
- Стрімінг - це онлайн-трансляція ігрового процесу у реальному часі, яка дозволяє гравцю спілкуватися з глядачами.

Летсплеї та стріминг викликали великий інтерес серед геймерів та негеймерів, яким подобається дивитися за ігровим процесом інших людей. Для деяких це спосіб навчитися грати краще, для інших - розважитися або отримати новий досвід. Ці два види відеоконтенту стали також джерелом доходу для багатьох гравців, які отримують гроші за перегляди своїх відео, підписки на їх канали, донати (пожертвування) від глядачів, спонсорство та рекламу.
Летсплеї почали з’являтися ще в 1970-х роках, коли гравці записували свої рекорди у аркадних іграх. Першими летсплеями в інтернеті вважаються скріншоти з гри «The Oregon Trail», які були опубліковані на форумі «Something Awful» у 2007 році користувачем Майколом Сауєром.
А стрімінг відеоігор став можливим завдяки розвитку технологій, таких як швидкісний інтернет, веб-камери та спеціальне програмне забезпечення. Першими сайтами для стрімінгу відеоігор були Justin.tv (заснований у 2007 році) та Ustream (заснований у 2006 році). Поточним лідером у цьому напрямку є Twitch (заснований у 2011 році).

За даними Newzoo, у 2020 році аудиторія летсплеїв та стрімів склала 1,2 мільярда людей, а дохід - 9,3 мільярда доларів. А за даними Statista, у 2020 році найпопулярнішими іграми для летсплеїв та стрімів були League of Legends (1,4 мільярда годин перегляду), Fortnite (904 млн годин перегляду), Valorant (814 млн годин перегляду), Grand Theft Auto V (669 млн годин перегляду) та Counter-Strike: Global Offensive (597 млн годин перегляду).

З одного боку, можна сказати, що стрімінг є еволюцією летсплеїв, оскільки він використовує сучасні технології та надає більше можливостей для взаємодії з глядачами. З іншого боку, можна вважати, що стрімінг і летсплей - це два різних формати, які мають свої особливості та переваги. Летсплей дозволяє гравцю показати свою точку зору на гру, редагувати відео та додавати ефекти. Стрімінг же дозволяє гравцю показати свою реакцію на гру в реальному часі, спілкуватися з глядачами та адаптуватися до ситуації. Тому можна сказати, що стрімінг і летсплей - це два паралельних напрямки, які розвиваються в ігровій культурі.

Говорячи про культуру у відеоігровій індустрії не можна обійти заходи на яких збираються любителі відеоігор, щоб дізнатися про новинки індустрії, пограти в різні ігри, поспілкуватися з однодумцями та розробниками, взяти участь у конкурсах та турнірах - **відеоігрові фестивалі**!

Фестивалі формують ігрову спільноту, яка об’єднується за спільними інтересами, цінностями та емоціями, створюючи дружню атмосферу та сприяючи соціалізації. Для самих розробників вони стимулюють розвиток відеоігрової індустрії, надаючи розробникам можливість отримати зворотний зв’язок від гравців, презентувати свої проекти та залучати інвесторів. І загалом сприяють популяризації відеоігор як виду розваги та культури, показуючи їх різноманітність та якість.

Одним з найбільших і найвідоміших відеоігрових фестивалів є E3 (Electronic Entertainment Expo), який проводиться щорічно у Лос-Анджелесі з 1995 року.

На цьому фестивалі презентуються найочікуваніші ігри і приставки від провідних компаній і студій. Іншими популярними фестивалями є Gamescom у Німеччині, Tokyo Game Show у Японії, PAX (Penny Arcade Expo) у США і Австралії та інші.

(Рис. 1.2 один з багатьох стендів та локацій фестивалю E3)

Відеоігрові фестивалі проходять по-різному в залежності від їх формату, масштабу та тематики. Зазвичай вони тривають кілька днів і включають такі елементи:

- Презентації нових ігор, приставок, технологій та трендів від розробників, видавців та експертів індустрії.
- Демонстрації ігор, де відвідувачі можуть пограти в різні ігри на спеціально обладнаних стендах або на власних пристроях.
- Конкурси та турніри, де гравці можуть змагатися між собою за призи та славу в різних іграх і жанрах.
- Косплей, де учасники одягаються в костюми своїх улюблених персонажів з ігор та беруть участь у парадах, шоу та фотосесіях.
- Майстер-класи, лекції та семінари, де відвідувачі можуть дізнатися про різні аспекти створення, розробки та маркетингу відеоігор від професіоналів.

- Виставки, де можна побачити раритетні або ексклюзивні предмети пов’язані з історією та культурою відеоігор.
- Концерти, де виступають музиканти, які грають музику з відеоігор або натхненну ними.

Однак у 2020-2021 роках багато відеоігрових фестивалів було скасовано або перенесено через пандемію COVID-19. Це призвело до занепаду фестивальної активності та до переходу на онлайн-трансляції. Онлайн-трансляції мають свої переваги та недоліки. З одного боку, вони дозволяють досягти більшої аудиторії, зекономити час і гроші на подорожі, уникнути скупчення людей і інших ризиків. З іншого боку, вони не можуть передати атмосферу живого спілкування, емоції та враження від гри на повноцінному екрані і з якісним звуком. Також онлайн-трансляції можуть мати проблеми з технічною якістю, затримкою або перебоями зв’язку. Тому багато гравців і розробників сподіваються на повернення офлайн-фестивалів у майбутньому. Перший відеоігровий фестиваль був проведений у 1972 році в Стенфордському університеті США. Він називався Intergalactic Spacewar Olympics і був присвячений грі Spacewar!, яка була створена ще в 1962 році на комп’ютері PDP-1. На фестивалі змагалися 24 гравці, які керували космічними кораблями і намагалися знищити один одного. Переможець отримав підписку на журнал Rolling Stone. Також цей фестиваль вважається першим прикладом електронного спорту.

Далі можна розказати про той самий електронний спорт або **кіберспорт.** Він має свої корені ще в 1970-х роках, коли гравці змагалися за найвищий бал у аркадних іграх. Першою лігою кіберспорту стала Cyberathlete Professional League (CPL), яка була заснована в 1997 році в США. Кіберспорт став популярним видом розваги та культури, який приваблює мільйони глядачів та спонсорів. Кіберспорт також створює можливості для професійного розвитку та освіти у цій сфері. Він поділяється на різні дисципліни, які визначаються жанром ігри, правилами та форматом змагань.

Кіберспорт проводиться як на масштабних офлайн-турнірах, так і на онлайн-платформах. Деякі з найбільших кіберспортивних організацій - це ESL, DreamHack, StarLadder, Valve та інші. Онлайн-трансляції кіберспортивних подій можна дивитися на сервісах Twitch, YouTube, Facebook Gaming тощо.

(Рис. 1.3 Турнір ”The International” від студії Valve)

Кіберспорт є високо конкурентною та прибутковою індустрією, яка привертає увагу багатьох інвесторів, спонсорів та медіа. Та за даними Newzoo, світовий дохід від кіберспорту в 2020 році склав 947 млн доларів, а глобальна аудиторія - 495 млн людей.

1.3 Жанри та їх специфіка

Жанри відеоігор - це категорії або класифікації, які використовуються для опису та визначення типу геймплею, тематики та механік, які присутні в грі. Вони використовуються для упорядкування ігрових досвідів та допомагають гравцям зорієнтуватися в широкому асортименті ігор, що існують. Кількість жанрів відеоігор не є точною чи остаточною, оскільки геймінгова індустрія постійно розвивається, а з нею з'являються нові жанри та гібриди. Значна кількість жанрів вже існує і визнана в гральній спільноті, але їхнє число може

змінюватися та розширюватися з часом, на сьогоднішній день можна виокремити сотні різних жанрів ігор.

Відеоігри мають широкий спектр жанрів, кожен з яких має свою власну специфіку та особливості:

— Екшн (Action): Цей жанр характеризується швидким темпом, бойовими сценами та високою динамікою. Він може поділятися на піджанри, такі як шутери (первинна особа або третинна особа), файтинги (бойовики один на один) та платформери (пересування по різних рівнях).

— Рольові ігри (RPG): Цей жанр зазвичай включає розвиток персонажа, систему прокачки навичок та управління історією. Гравцю надається можливість приймати рішення, що впливають на подальший розвиток сюжету. Рольові ігри можуть бути як одиночними, так і мультиплеєрними.

— Пригодницькі ігри (Adventure): Цей жанр фокусується на розв'язуванні головоломок, пошуку предметів та дослідженні світу гри. Часто включає в себе наративні елементи, а гравець має вирішувати різні завдання, щоб продовжити історію.

— Стратегії (Strategy): Цей жанр передбачає планування та керування ресурсами для досягнення певних цілей. Існують різні види стратегій, такі як в реальному часі (Real-Time Strategy, RTS), пошагові (Turn-Based Strategy) та великомасштабні стратегії (Grand Strategy).

— Головоломки (Puzzle): У цьому жанрі гравець залучений до вирішення різних головоломок та логічних завдань. Це можуть бути такі ігри, як тетріс, головоломки зі з'єднанням ліній або вирішенням складних загадок.

— Симулятори (Simulation): Цей жанр створює віртуальні умови для імітації реального життя або конкретної ситуації. Це можуть бути авіасимулятори, сімулятори будівництва міст, фермерські симулятори та інші.

— Спортивні ігри (Sports): Цей жанр орієнтований на імітацію різних видів спорту, таких як футбол, баскетбол, гольф або автоспорт. Гравці можуть змагатися один з одним або грати проти штучного інтелекту.

— Хоррор (Horror): Цей жанр спрямований на створення атмосфери страху та жаху. Гравцеві доводиться зустрічатися з небезпеками, ворожими створіннями або розв'язувати загадки в моторошному середовищі.

— Метроїдванія (Metroidvania): Цей жанр поєднує елементи платформера та рольової гри. Гравець досліджує великий світ з відкритими регіонами, отримує нові навички та здібності, щоб розблоковувати раніше недоступні ділянки.

— Гонки (Racing): Цей жанр фокусується на швидкісних гонках транспортних засобів, таких як автомобілі, мотоцикли або космічні кораблі. Гравцеві потрібно долати траси, змагатися з іншими гравцями або штучним інтелектом.

— MMO (Massively Multiplayer Online): Цей жанр включає велику кількість гравців, які одночасно взаємодіють у віртуальному світі. MMO-ігри можуть належати до різних жанрів, включаючи MMO-RPG, MMO-стратегії та MMO-шутери.

— Інтерактивні історії (Interactive Story): Цей жанр зосереджений на наративі та взаємодії гравця з історією. Гравець може впливати на події, приймати рішення та відкривати різні сюжетні лінії.

— Бойові арени (Battle Royale): Цей жанр передбачає битви гравців на великій мапі, де єдиний вижившій або команда виграє гру. Гравці збирають ресурси, знаряддя та зброю, щоб залишитися в живих і перемогти інших гравців.

— Стелс (Stealth): Цей жанр базується на уникненні прямого конфронтації з ворогами шляхом хитрощів, хування та непомітного проникнення. Гравцеві

доводиться діяти тихо і незамітно, використовуючи різні прийоми для досягнення цілей.

— Відкритий світ (Open World): Цей жанр відкриває широкі простори для гравця, де він може вільно досліджувати великий віртуальний світ і взаємодіяти з різними персонажами та елементами оточення. Відкритий світ часто пропонує багато завдань, місій та можливостей для самовираження.

— Будівництво/Симуляція (Construction/Simulation): Цей жанр зосереджений на будівництві, управлінні та розвитку різних об'єктів або ситуацій. Включає в себе такі піджанри, як симулятори міста, симулятори транспорту, будівництво парку атракціонів та інші.

— Мистецтво та експериментальні (Art/Experimental): Цей жанр фокусується на творчому виразі, естетиці та експериментах з геймплеєм, наративом та візуальними елементами. Ігри цього жанру часто надають особливий акцент на наратив, емоції та враження, які вони створюють.

— Бойові ігри (Fighting): Цей жанр зосереджений на бойових поєдинках між гравцями або контрольованими комп'ютером персонажами. Гравці використовують різноманітні рухи, комбінації та стратегії, щоб перемогти свого суперника.

— Рогалик (Roguelike): Цей жанр характеризується випадково згенерованими рівнями, смертельними наслідками за втрату гри та постійними випробуваннями. Гравці досліджують підземелля або інші середовища, збирають предмети та борються зі ворожими силами.

— Карткові ігри (Card Games): Цей жанр використовує колоди карт або деки карт для гри. Це можуть бути стратегічні карткові ігри, колекційні карткові ігри (CCG), де гравці збирають і покращують свої колекції карт, або різноманітні ігри з використанням стандартних карткових колод.

— Музичні ігри (Music Games): Цей жанр використовує музику та ритм як основну механіку гри. Гравцям доводиться виконувати різні дії або натискати на кнопки відповідно до музичного супроводу, що дозволяє створити музичний ритм або виконати певні мелодії.

— Казуальні ігри (Casual Games): Цей жанр спрямований на простоту та доступність для гравців будь-якого рівня досвіду. Казуальні ігри мають прості правила та геймплей, і їх можна легко засвоїти та грати в них без великих зусиль.

— Екшн-пригоди (Action-Adventure): Цей жанр комбінує елементи екшну та пригодницького жанру. Гравцеві доводиться виконувати дії швидкості, реакції та бойових навичок, а також розв'язувати головоломки та досліджувати світ.

— Виживання (Survival): Цей жанр ставить гравця у ворожому середовищі, де йому доводиться боротися за виживання. Гравці повинні забезпечити собі їжу, воду та інші ресурси, уникати небезпек та ворожих сил, будувати притулки та розвивати навички, щоб протистояти складнощам.

Нові жанри можуть виникати внаслідок технологічних інновацій, змін в геймплеї, розвитку віртуальної реальності та інших факторів. Тому кількість жанрів може продовжувати зростати, розширюючи палітру геймінгових можливостей.

Для програміста жанри відеоігор мають меншу вагу, ніж для гравців або дизайнерів ігор. Програмісти зазвичай концентруються на реалізації функціональності ігрових систем та механік, незалежно від конкретного жанру. Однак, програмістам важливо мати розуміння про різні жанри відеоігор, оскільки це може вплинути на архітектуру програмного забезпечення та технічні рішення, які вони приймають. Різні жанри можуть вимагати різних алгоритмів, фізичної симуляції, штучного інтелекту та інших технічних аспектів.

Наприклад, програмістам, які працюють над шутерами, може знадобитися розуміння алгоритмів обробки стрільби, розрахунку траєкторій куль, взаємодії зі зброєю та штучного інтелекту ворожих персонажів. Для розробки рольових ігор програмістам можуть знадобитися навички управління персонажами, розвитку навичок та системи квестів.

Таким чином, розуміння різних жанрів відеоігор допомагає програмістам враховувати особливості кожного жанру та забезпечувати відповідну функціональність та досвід для гравців.

1.4 Симулятори та їх призначення

Симулятори - це вид відеоігор або програмного забезпечення, яке створює реалістичну модель або симуляцію певної ситуації, процесу, об'єкта або системи. Вони дозволяють користувачеві відчути або експериментувати з віртуальною версією реального світу або імітувати певний досвід.

Вони можуть охоплювати різні сфери і тематики, такі як авіація, автомобільна промисловість, залізниця, судноплавство, космос, риболовля, військові операції, будівництво, управління містом, фермерство, хірургія, спорт і багато інших. Вони можуть бути реалістичними симуляторами, що детально відтворюють фізичні аспекти та правила, або ж більш аркадними симуляторами, які спрощують деякі аспекти для більш простого геймплею.

Симулятори можуть надавати гравцям можливість виконувати реальні завдання, тренуватись у певних навичках, досліджувати та вивчати різні аспекти певного предмета або просто насолоджуватись іммерсивним віртуальним досвідом. Вони можуть використовувати різні технології, такі як віртуальна реальність (VR), датчики руху, спеціальні контролери та інші пристрої, щоб забезпечити більш інтерактивний та реалістичний досвід. Симулятори також застосовуються в різних галузях, таких як навчання, тренування, військова сфера, медицина, наука, інженерія та інші. Вони можуть бути використані для симуляції складних процесів, прогнозування

результатів або вивчення поведінки системи в різних сценаріях.
Іммерсивність (від англ. immersive - "присутність, занурення") - це ступінь занурення гравця у віртуальний світ або різні види змішання реальної і віртуальної реальності. Іммерсивність може залежати від багатьох факторів, таких як графіка, звук, ігрова механіка, сюжет, інтерфейс та інтерактивність. Іммерсивні технології можуть використовувати різні платформи та пристрої. Вона може забезпечувати ефект повної або часткової присутності в альтернативному просторі і тим самим змінювати досвід гравця в абсолютно різних сферах.

Одним із ключових аспектів симуляторів є намагання створити якомога більш реалістичний та іммерсивний досвід, щоб гравець міг відчути себе частково або повністю поглиненим у віртуальному світі або ситуації.

Розвиток симуляторів залежить від розвитку технологій, які дозволяють створювати більш реалістичну та іммерсивну графіку, звук, інтерактивність та іммерсивність. Також важливою є розробка пристроїв введення та виведення, які дозволяють гравцям керувати симуляторами та отримувати зворотний зв’язок. Наприклад, клавіатура, мишка, геймпад, джойстик, кермо, педалі, шолом віртуальної реальності тощо.
Сам розвиток симуляторів почався ще у 1940-х роках з появою перших електронних ігор, які використовували електронно-променеві трубки для відображення графіки. З того часу симулятори пройшли довгий шлях від простих ігор на базі математичних моделей до складних ігор на базі фізичних законів та штучного інтелекту.

Одним з перших симуляторів можна вважати “Cathode Ray Tube Amusement Device”, створений у 1947 році. Це була проста гра, у якій гравець мав прицілюватися в мету на екрані за допомогою електронно-променевої трубки1. Якщо ми розуміємо симулятор як комп’ютерну гру, яка використовує цифровий процесор для обробки інформації, то одним з

перших симуляторів можна вважати “Spacewar!”, створений у 1962 році. Та якщо ми розуміємо симулятор як фізичний пристрій, який використовує копії кабін реальних транспортних засобів, то одним з перших симуляторів можна вважати “Link Trainer”, створений у 1929 році. Це був симулятор польоту, який імітував реальне керування літаком і надавав зворотний зв’язок гравцеві.

З одного боку, повнокабінні симулятори використовують технології та принципи відеоігор, такі як графіка, фізика, інтерактивність тощо. З іншого боку, повнокабінні симулятори не розробляються для розваги або комерційного успіху, а для навчання та професійного розвитку. Також вони не доступні для широкої аудиторії, а тільки для спеціалізованих установ та організацій. Хоча обидва види симуляторів намагаються відтворити реальні або уявні ситуації, процеси та дії за допомогою візуальних, звукових та інтерактивних засобів. Обидва види симуляторів вимагають від гравця/пілота використання певних навичок та знань для досягнення цілей або задоволення інтересу. Обидва види симуляторів можуть мати навчальну або розважальну ціль.

Відеоігрові симулятори є цифровими програмами, які працюють на комп’ютерах, консолях або мобільних пристроях, вони доступні для широкої аудиторії та мають різноманітні жанри та піджанри1. Кабінні симулятори є фізичними пристроями, які використовують копії кабін реальних транспортних засобів. Вони контролюються за допомогою елементів керування, що імітують реальне управління. Вони обладнані системами руху, які створюють враження прискорення та інерції. Вони не доступні для широкої аудиторії та мають обмежене коло застосувань.

Симулятори та кабінні симулятори працюють на базі спільних та різних технологій та принципів:

- **Математичні моделі.** Симулятори та кабінні симулятори використовують математичні моделі для опису та відтворення реальних або уявних ситуацій, процесів та дій. Математичні моделі можуть бути складними або простими, залежно від мети та рівня деталізації симуляції. Наприклад, симулятор польоту може використовувати математичну модель Newton-Euler equations, яка описує рух твердого тіла в просторі за допомогою шести змінних: трьох лінійних координат і трьох кутових координат. Або симулятор життя може використовувати математичну модель Conway’s Game of Life, яка описує еволюцію клітин на двомірній сітці за допомогою двох правил: клітина залишається живою, якщо у неї два або три живих сусіда, і народжується нова клітина, якщо у неї рівно три живих сусіда.
- **Графічний двигун.** Симулятори та кабінні симулятори використовують графічний двигун для створення та виведення візуальної частини симуляції. Графічний двигун - це програмне забезпечення, яке надає інструменти та бібліотеки для роботи з графікою, анімацією, освітленням, текстурами тощо. Графічний двигун може бути загального призначення або спеціалізованого для певного типу симуляції. Наприклад, Unreal Engine - це графічний двигун загального призначення, який підтримує такі функції, як ray tracing, global illumination, particle system, skeletal animation тощо. Або X-Plane - це графічний двигун спеціалізованого для симуляторів польоту, який підтримує такі функції, як atmospheric scattering, cloud rendering, terrain mesh, flight instruments тощо.
- **Звуковий двигун.** Симулятори та кабінні симулятори використовують звуковий двигун для створення та виведення звукової частини симуляції. Звуковий двигун - це програмне забезпечення, яке надає інструменти та бібліотеки для роботи з звуком, музикою, ефектами, просторовим розташуванням тощо. Звуковий двигун може бути

загального призначення або спеціалізованого для певного типу симуляції. Наприклад, FMOD - це звуковий двигун загального призначення, який підтримує такі функції, як sound synthesis, sound mixing, sound effects, 3D sound positioning тощо. Або DCS World - це звуковий двигун спеціалізованого для симуляторів польоту, який підтримує такі функції, як engine noise, weapon fire, radio chatter, doppler effect тощо.

- **Система руху.** Кабінні симулятори використовують систему руху для створення та виведення кинестетичної частини симуляції. Система руху - це механічне устаткування, яке змінює положення кабіни в просторі в залежності від даних симуляції. Система руху може бути статичною або динамічною, залежно від кута нахилу та ступеня свободи кабіни. Наприклад, гідравлічна система руху - це динамічна система, яка підтримує шість ступенів свободи (x,y,z,pitch,yaw,roll) і використовує гидравлійний насос і гидравлійний циліндр для перестановки кабіни. Або електромеханічна система руху - це система руху статична або динамічна (залежно від моделі), яка підтримує два або чотири ступеня свободи (pitch,yaw або pitch,yaw,x,z) і використовує електромотор і шарико-гвинтову пару для перестановки кабіни.

Чи можна вважати повнокабінні симулятори частиною геймдеву та ігрової індустрії?

Повнокабінні симулятори, хоча вони використовують деякі технології та принципи відеоігор, мають свої унікальні особливості та цільову аудиторію, що відрізняють їх від традиційних відеоігор. Спільного у спеціалістів повнокабінних симуляторів та відеоігрових симуляторів може бути те, що вони обидва працюють з тривимірною графікою, моделюванням, анімацією та іншими аспектами візуалізації. Вони також можуть використовувати схожі програми та інструменти для своєї роботи, такі як Autodesk Maya, ZBrush, Substance Painter тощо. Вони також можуть мати загальне розуміння процесу

розробки симуляторів та їх особливостей. Спільними мовами програмування для повнокабінних симуляторів та відеоігрових симуляторів можуть бути ті, що використовуються для розробки ігрових рушіїв, таких як Unreal Engine або Unity. Наприклад, Unreal Engine підтримує C++ та Blueprint, а Unity підтримує C# та JavaScript. Ці мови дозволяють створювати логіку, інтерфейс, анімацію та інші елементи симуляторів.

Я думаю, що це все ще залежить від того, як ми визначаємо геймдев та ігрову індустрію. Якщо ми вважаємо геймдев за процес розробки відеоігор, а ігрову індустрію за сферу діяльності, пов’язану з виробництвом та продажем відеоігор, то спеціалісти та програмісти у цих сферах можуть мати спільні навички та знання. Наприклад, вони можуть використовувати однакові мови програмування, рушії, алгоритми тощо. Також вони можуть співпрацювати над спільними проектами або переходити з однієї сфери до іншої. Однак це не означає, що кабінні симулятори є частиною геймдеву та ігрової індустрії, адже вони мають різну мету та аудиторію для своїх продуктів.
Якщо ж ми вважаємо геймдев за вид мистецтва, а ігрову індустрію за сферу діяльності, пов’язану з створенням та споживанням відеоігор як форми культурного вираження, то спеціалісти та програмісти у цих сферах можуть мати різні навички та знання. Наприклад, вони можуть використовувати різні стилі, жанри, тематики тощо. Також вони можуть мати різне бачення та цінності щодо своїх продуктів. Отже, кабінні симулятори не є частиною геймдеву та ігрової індустрії, адже вони не створюються для розваги або комерційного успіху, а для навчання та професійного розвитку.

Отже, можна зробити висновок, що повнокабінні симулятори представляють окрему галузь, яка використовує певні аспекти відеоігрової технології, але зорієнтована на навчання та професійний розвиток у специфічних галузях.

РОЗДІЛ 2. ОСОБЛИВОСТІ ГЕЙМДЕВУ

2.1 Характеристика геймдизайнера

Геймдизайнер - це фахівець, який відповідає за процес створення геймплею, механік і систем в комп'ютерних іграх. Геймдизайнер відповідає за створення ідеї гри, розробку сценарію, геймплейних механік, рівнів, персонажів, правил гри, балансу та інших аспектів, що формують геймплей.

Геймплей - це термін, який використовується для опису гри в комп'ютерних іграх. Він відноситься до того, як гра взаємодіє з гравцем, як гравець взаємодіє з грою і як відбувається взаємодія між елементами гри. Геймплей охоплює різні аспекти гри, такі як механіка, контроль, системи, взаємодія з оточенням, завдання та цілі, виконання дій і прийняття рішень гравцем. Він визначає, як гра відтворюється, які можливості та обмеження є у гравця, як розвиваються події і як впливають вчинки гравця на подальший хід гри. Геймплей може бути різноманітним в залежності від жанру і типу гри. Наприклад, у шутерах від першої особи головним аспектом геймплею може бути точність стрільби та тактичні рішення, а в рольових іграх - розвиток персонажа, виконання завдань і взаємодія зі світом гри.

Геймплей впливає на відчуття, задоволення та захоплення гравця. Його якість залежить від балансу, чіткості правил, викликів та цікавості завдань, рівня складності, плавності керування та багатьох інших факторів. Хороший геймплей робить гру захоплюючою, цікавою та дозволяє гравцеві відчувати контроль і вплив на події у грі.
Геймплей є однією з ключових складових успіху гри, і геймдизайнери працюють над його розробкою та вдосконаленням, щоб забезпечити незабутній ігровий досвід.

Геймдизайнери працюють у команді з розробниками програмного забезпечення, художниками, сценаристами та іншими фахівцями, щоб втілити концепцію гри в реальному продукті. Вони визначають цілі та завдання гри, проектують різноманітні рівні та загадки, налаштовують баланс геймплею, тестують ігрові механіки та вносять корективи для покращення якості гри.
Він повинен мати розуміння геймплейних принципів, сприйняття гравця, технічних можливостей і обмежень платформи, на якій розробляється гра. Він також має аналітичні навички, щоб оцінювати та налагоджувати геймплей під час тестування гри.

У процесі своєї роботи геймдизайнер може використовувати різні інструменти, такі як прототипування геймплею, діаграми, флоу-чарти, документацію гри, розробку рівнів тощо.

Загалом, геймдизайнер відіграє ключову роль у створенні захопливих та цікавих ігрових досвідів, і його робота вимагає поєднання творчості, аналітики та глибокого розуміння геймплейних принципів та ігрових механік.
Механіки у відеоіграх - це набір правил і методів, які визначають ігровий процес, поведінку об’єктів і зв’язок між ними. Механіки впливають на те, як гравець взаємодіє з ігровим світом, які цілі і завдання він має виконувати, які наслідки його дій і т.д Механіки можуть бути різними залежно від жанру, платформи, стилю і концепції відеогри. Наприклад, механіки стратегії зазвичай включають управління ресурсами, будівництво бази, планування тактики тощо. Механіки рольової гри часто передбачають розвиток персонажа, діалоги з NPC, вибір моральних рішень тощо. Механіки симулятора намагаються відтворити реалістичну фізику, моделювання систем, управління складними пристроями тощо.
Механіки можуть бути також поділені на основні і допоміжні. Основні механіки - це ті, що формують основу ігрового процесу і без яких гра не

може функціонувати. Допоміжні механіки - це ті, що додають розмаїття, складність або цікавинку до гри, але не є обов’язковими для її проходження. Наприклад, у шутеру основною механікою може бути стрільба з різних видів зброї, а допоміжною - прицілювання через приціл, кидання гранат, перезарядка тощо. Механіки у відеоіграх є одним з ключових елементов ігрового дизайну і потребують ретельного проектування, програмування і тестування. Вони також повинні бути зрозумілими і доступними для гравців і підтримувати їх зацікавленість і мотивацію.

Існує кілька типів геймдизайнерів, кожен з яких має свої спеціалізації та області відповідальності. Ось декілька основних типів геймдизайнерів:

- **Головний геймдизайнер:** Це вищий рівень геймдизайнера, який відповідає за загальну концепцію гри, її механіку, геймплей та емоційний досвід. Він визначає стратегічні напрямки розробки гри та керує роботою інших геймдизайнерів у команді.
- **Дизайнер геймплею:** Цей тип геймдизайнера спеціалізується на створенні геймплею, механік гри та правил. Він визначає взаємодію гравця з грою, розвиває завдання та виклики для гравця, працює над балансом гри та забезпечує захоплюючий геймплейний досвід.
- **Дизайнер рівнів:** Цей тип геймдизайнера спеціалізується на створенні рівнів та просторів у грі. Він проектує ландшафти, пастки, загадки та виклики на різних рівнях гри. Дизайнер рівнів стежить за прогресом гравця та розробляє унікальні та цікаві локації для подорожей гравця.
- **Дизайнер персонажів:** Цей тип геймдизайнера відповідає за створення персонажів у грі. Він розробляє їх зовнішній вигляд, особливості, властивості, навички та характеристики. Дизайнер персонажів також працює над системою прогресу, розвитку персонажа та взаємодії з іншими персонажами.
- **Дизайнер історії:** Цей тип геймдизайнера займається розробкою сюжету, діалогів, квестів та кампаній у грі. Він створює цікаві та

захоплюючі історії, які поглинають гравця в ігровий світ і роблять гру більш насиченою.

Ці типи геймдизайнерів можуть перетинатися та доповнюватися в залежності від розміру команди та конкретних вимог проекту. У більших командах розробки гри можуть бути додаткові спеціалізовані ролі геймдизайнерів, такі як дизайнер звуку, дизайнер мережевої гри тощо.

Геймдизайнер і програміст - це дві різні ролі в процесі розробки комп'ютерних ігор, хоча в деяких випадках можуть існувати люди, які об'єднують ці дві функції. Геймдизайнер займається проектуванням геймплею, створенням механік гри, рівнів, персонажів, правил гри та інших аспектів, що впливають на геймплей. Він фокусується на творчому процесі, вирішенні головоломок та завдань для гравців, створенні емоційних досвідів, розробці ідеї гри та налаштуванні балансу.

Програміст, з іншого боку, відповідає за технічну реалізацію гри. Він програмує логіку гри, реалізовує геймплейні механіки, рівні, штучний інтелект та інші технічні аспекти гри. Програміст займається розробкою коду, роботою з графікою та звуком, оптимізацією продукту та вирішенням технічних проблем.
Хоча деякі геймдизайнери можуть мати базові навички програмування і вміти працювати з інструментами для розробки ігор, аби прототипувати свої ідеї, загальновизнаною практикою є те, що геймдизайнери та програмісти співпрацюють як частина команди розробки гри.

Великі команди розробників ігор можуть включати окремих геймдизайнерів та окремих програмістів, які спілкуються між собою, обговорюють ідеї та працюють разом, щоб створити високоякісний продукт.

Взаємодія геймдизайнера та програміста є критично важливою для успішної розробки комп'ютерних ігор. Обидва фахівці вносять свій внесок у різні аспекти гри і спілкуються для досягнення спільних цілей. Ось декілька

способів, якими геймдизайнери та програмісти взаємодіють під час розробки гри:

- **Збір вимог:** Геймдизайнер та програміст спільно обговорюють вимоги до гри. Геймдизайнер пояснює свої ідеї, концепції та вимоги до геймплею, механік, рівнів та інших аспектів гри. Програмісти уточнюють технічні можливості та обмеження і допомагають зрозуміти, які елементи можна реалізувати в межах даної технічної платформи.
- **Прототипування:** Геймдизайнер може створювати прототипи геймплею або конкретних механік, щоб продемонструвати свої ідеї. Програмісти спілкуються з геймдизайнером, аналізують його прототипи і допомагають реалізувати їх у вигляді функціонального коду. Взаємодія між геймдизайнером і програмістами на цьому етапі допомагає вирішувати проблеми та вносити корективи для поліпшення геймплею.
- **Зворотній зв'язок та ітерації:** Під час розробки гри геймдизайнер і програміст мають постійний зв'язок. Геймдизайнери надають програмістам зворотний зв'язок щодо реалізації геймплею, рівнів, персонажів та інших елементів гри. Програмісти, у свою чергу, спілкуються з геймдизайнерами щодо технічних обмежень та можливостей, а також пропонують свої ідеї та рішення для поліпшення гри. Цей процес зворотного зв'язку та ітерацій допомагає досягти кращого збалансованого результату.
- **Тестування та відладка:** Після реалізації геймплею та інших елементів гри, геймдизайнери та програмісти спільно проводять тестування гри. Вони спостерігають за роботою гри, виявляють проблеми, помилки та неузгодженості. Геймдизайнери надають фідбек щодо геймплею, балансу та інших аспектів гри, а програмісти

виправляють технічні проблеми та вдосконалюють гру згідно з вимогами геймдизайнера.

Усі ці етапи взаємодії геймдизайнера та програміста є динамічними та вимагають постійного спілкування та співпраці. Обидва фахівці вносять свої унікальні знання та навички для створення захопливих та якісних ігор.

Ігрова індустрія має багато видатних геймдизайнерів, які внесли великий вклад у розвиток цієї галузі. Наприклад, Хідео Кодзіма - це відомий японський геймдизайнер, сценарист, продюсер та керівник розробки відеоігор1. Він є творцем серії ігор Metal Gear, яка вплинула на жанр стелс-ігор та ігрову індустрію в цілому. Кодзіма також розробив пригодницькі ігри Snatcher і Policenauts, а також Death Stranding - першу гру поза Konami - одина з провідних компаній розробників і видавців відеоігор. Кодзіма мав складні відносини з Konami, особливо після 2010 року. Він часто зазнавав обмежень, тиску та конфліктів з керівництвом компанії щодо своєї креативності, бюджету та строків своїх проектів. Він також не отримав повного контролю над своєю інтелектуальною власністю - серією ігор Metal Gear.

Кодзіма славиться своїм авторським підходом до створення ігор, який поєднує складні сюжети, кінематографічну презентацію, експериментальний геймплей та соціальну критику. Він також часто використовує метафори, алегорії та посилання на реальні події, особистості та медіа. Кодзіма називає свої ігри “творами”, а не “продуктами”, і вважає їх формою художньої творчості. Кодзіма отримав багато нагород та визнання за свою роботу в ігровій галузі. В 2009 році він отримав премію Game Developers Choice Award - Lifetime Achievement за свою довготривалу кар’єру. В 2020 році він отримав премію академії BAFTA за досягнення у кінематографі за свою вагомий вклад у розвиток ігрової культури1. В 2021 році він був удостоєний

ордена Почесного легіону Франції за свою роль у популяризації французької культури через свої ігри.

(Рис. 2.1 Той самий геній Хідео Кодзіма)

Кодзіма часто називають генієм через його оригінальну та новаторську візію ігор, яка не обмежується стандартами та очікуваннями. Він також має сильний авторитет та вплив на ігрову спільноту та інших розробників. Кодзіма є одним з найвизначніших та найбільш поважаних геймдизайнеров сучасності.

Ще його геніальність проявляється в неочевидних дрібницях у своїх іграх. Наприклад: у серії ігор Metal Gear Solid є організація “Патріоти”, котра заразила наномашинами більшість людей з влади, і якщо інфіковані хотіли

розказати щось про організацію чи вимовити її назву, вони говорили дивний набір звуків - “Ла-лі-лу-ле-ло”, у контексті сюжету це зроблено для того, що присікти розповсюдженню інформації про організаці, але чому саме ці звуки? Справа у тому, що ці звуки в японській мові не мають ієрогліфа та звуку, котрий означав би букву “Л”, тому якщо для більшості людей світу ці звуки звучать просто дивно, то для японців, які навіть не могли вимовити або написати їх, це додавало завісу таємничості та шпигунської хитрості.

Геймдизайнери використовують свою креативність, талант та технічні навички, щоб розробляти ігри, які захоплюють, розважають та надихають гравців. Вони також часто вводять нововведення, експерименти та революції в індустрії. Геймдизайнерів часто називають "художниками відеоігор" через їхню здатність створювати емоційні, захоплюючі та візуально привабливі відеоігри.

2.2 Симбіоз мистецтва та програмування

Програмування та мистецтво у відеоіграх - це дві сторони однієї медалі. Вони обидва необхідні для створення захоплюючих ігрових світів, які вражають гравців своєю красою та функціональністю. Програмування дозволяє реалізувати ідеї та концепції гри у вигляді коду, який виконується на комп’ютері або іншому пристрої. Мистецтво додає грі естетичної цінності та емоційного зв’язку з гравцем. Програмування та мистецтво у відеоіграх поєднуються у різних аспектах розробки відеогри, таких як:

- **Ігровий дизайн** - це процес придумування ігрового світу, сюжету, персонажів, правил та цілей гри. Ігровий дизайн вимагає креативності та логічного мислення. Ігровий дизайнер повинен враховувати бажання та очікування гравця, а також можливості та обмеження технологій. Наприклад, ігровий дизайнер може придумати історію про подорож у часі, але повинен також продумати, як реалізувати цю ідею у коді та графіці.

- **Графіка у відеоіграх** - це створення візуального контенту для гри, такого як моделі, текстури, освітлення, тіньовання, анімація тощо. Графіка у відеоіграх вимагає художнього смаку та навичок роботи з програмами для 2D- та 3D-моделювання та анімації. Графік повинен створювати зображення, які відповідають стилю та настрою гри, а також оптимізовані для швидкого завантаження та відображення. Наприклад, графік може створити деталізований ландшафт для пригодницької гри або мінімалістичний фон для пазла.
- **Музика у відеоіграх** - це створення звукового супроводу для гри, який складається з музики, озвучення персонажів та навколишнього середовища, звукових ефектів тощо. Музика у відеоіграх вимагає музичного слуху та обладнання для запису і обробки звуку. Композитор повинен писати музику, яка пасує до жанру та ситуацій у грі, а також стимулює емоції гравця. Наприклад, композитор може написати захоплюючий саундтрек для екшн-гри або спокійну мелодію для релаксаційної гри.

Для поєднання програмування та мистецтва у відеоіграх програмісти та геймдизайнери використовують ряд інструментів та методик:

- **Редактори рівней** - це спеціальні програми, які дозволяють легко розставляти об’єкти на сценах гри без необхідності писати код. Редактори рівней допомагають створювати цікаві і різноманітні ігрові локації і наповнювати їх життям та динамікою.
- **Графічні рушії** - це програмне забезпечення, яке забезпечують базовий функціонал для створення і запуску ігор на різних платформах і пристроях. Графичны рушії надають можливості для роботи з 2D- і 3D-графикой, фізикою, штучним інтелектом, мережевими протоколами тощо. Прикладами популярних графічних рушіїв є Unity, Unreal Engine, Godot тощо.

- **Скриптові мови** - це мови програмування високого рівня, які не потребують компіляції і легко інтегруються в графічні рушії. Скриптові мови дозволяють швидко і легко писати код для ігровой логіки та поведінки об’єктів. Прикладами скриптових мов є Lua, Python, C# тощо.

Деякі люди вважають, що відеоігри можна розглядати як форму синтетичного мистецтва, оскільки вони поєднують різні художні та технологічні елементи для створення нових віртуальних світів та вражень. Відеоігри мають складну взаємодію між графікою, звуком, музикою, сценарієм та геймплеєм.

2.3 UX/UI дизайн

UX/UI дизайн у геймдеві - це процес створення зручних та привабливих інтерфейсів для ігор, які враховують потреби та очікування гравців. UX/UI дизайн має велике значення для гри, адже він впливає на те, як гравець сприймає гру, наскільки легко йому досягати цілей, які емоції викликає гра. UX/UI дизайнер повинен мати навички аналізу, креативності, знання психології кольору та сучасних тенденцій.
Для прикладу, розглянемо деякі аспекти UX/UI дизайну у геймдеві:

- **Психологічний портрет користувача.** Дизайнер повинен вивчати цільову аудиторію гри, її стать, вік, вподобання, хобі тощо. На основі цих даних він може планувати стратегію побудови дизайну та формувати бачення проекту.
- **Психологія кольору.** Кожен колір та поєднання кольорів викликає різні емоції й підходить під різні направлення проекту. Наприклад, синій та його відтінки — це про бізнес. Ці кольори викликають відчуття довіри, зосередженості, вимагають вдумливо сприймати інформацію. Зелений — найбільш спокійний колір. З ним людині найлегше сприймати інформацію. Це значить, що він найкраще

підходить для освітніх сервісів1. Дизайнер повинен обирати кольори з урахуванням жанру та стилістики гри.

- **Шаблони та конкуренти.** Дизайнер повинен аналізувати чужі напрацювання, використовувати виділені шаблони, які можна вбудувати в програму. Експерти вважають — UI-фахівці з'ясували, що краще працює в іграх. Саме це є патерном — загальноприйнятим варіантом проектування. Дизайнер повинен мати розуміння очікувань аудиторії у своїй розробці та не повторювати чужих досягнень.
- **Ескіз та оформлення.** На етапі початкового проектування дизайнер створює першу версію інтерфейсу, що є дуже спрощеною подобою очікуваного оформлення. Колір з декоративними елементами не дуже важливий, адже головне — розташувати іконки з піктограмами на своїх місцях. Після створення ескізу дизайнер адаптує оформлення під власні завдання, особливості стилістики, намагаючись внести новизну та зручність для кінцевого користувача.

Програміст має важливе значення у створенні UX/UI дизайну, адже він втілює в життя ідеї та ескізи дизайнера. Програміст повинен знати основи UX/UI дизайну, щоб розуміти логіку та цілі інтерфейсу, а також здатний бути гнучким та адаптивним до змін та вимог дизайнера. Програміст також повинен співпрацювати з дизайнером, щоб давати конструктивну критику, поради та пропозиції щодо можливостей та обмежень технологій. Програміст і дизайнер повинні мати спільну мову та бачення проекту, щоб створити якісний та функціональний UX/UI дизайн.

Якщо у відеогрі буде поганий UX/UI дизайн, то це може призвести до багатьох негативних наслідків. Якщо інтерфейс гри не пояснює свої функції, не дає підказок або не відповідає очікуванням гравця, то це може зробити гру складною та фруструючою. Наприклад, у грі Dark Souls немає чіткого та детального пояснення багатьох механік та статистик, що змушує гравця шукати інформацію в інтернеті або експериментувати на свій страх і ризик.

(Рис. 2.2 Інтерфейс гри Dark Souls: 1. Характеристики персонажа 2. Під час бою 3. Дослідження локацій 4. Інвентар)

Або перевантажений інтерфейс гри виводить забагато інформації на екран, і це може відволікати гравця від геймплею, знизити іммерсію та ускладнити сприйняття важливих даних. Наприклад, у грі World of Warcraft є багато різних показників та елементів на екрані, що робить інтерфейс забитим та непривабливим.

(Рис. 2.3 Інтерфейс з різними показниками під час бою у World of Warcraft)

Також можна додати що, якщо інтерфейс гри вимагає від гравця забагато клавіш або рухів мишкою, не підтримує налаштування або адаптацію під різні пристрої, то це може зробити гру незручною та втомлюючою. Наприклад, у грі Resident Evil 4 керування мишкою було дуже непродуманим та негладким, що робило стрільбу та рухи камери складними.

І навіть якщо інтерфейс гри не пасує до стилістики та жанру гри, не використовує психологію кольору або не привертає уваги гравця, то це може знизити естетичне задоволення від гри та її атмосферу. Наприклад, у грі Mass Effect: Andromeda багато елементів інтерфейсу були сірими та нудними, що не відображало космічної тематики та пригодницького настрою гри.

(Рис. 2.4 Інтерфейс апгрейду у Mass Effect: Andromeda)

Наслідком цього можуть стати три простих, але зрозумілих катастрофи для студії-розробника: фрустрація гравця - якщо інтерфейс гри не зручний, не інтуїтивний, не привабливий або не відповідає стилю та жанру гри, то це може знизити мотивацію гравця продовжувати грати або повертатися до гри. Це означає втрату аудиторії, яка буде шукати інші ігри з кращим дизайном. Також гра може отримати негативні відгуки та оцінки від гравців та критиків, що погіршить її репутацію та популярність. І, у купі з іншими можливими недоліками гри, усе може завершитися збитками для розробників, які вклали час та ресурси в створення гри. Гра може не окупитися або навіть принести збитки. Розробники можуть втратити свою репутацію та довіру в очах гравців та видавців.

2.4 Ігрові рушії

Ігрові рушії - це програмні платформи, які надають розробникам ігор набір інструментів та функцій для створення і запуску ігор. Ігрові рушії мають велике значення у геймдеві, оскільки вони визначають можливості та обмеження гри. Термін “ігровий рушій” з’явився у середині 1990-х років, особливо у зв’язку з 3D іграми, такими як шутери від першої особи з використанням шутерного рушія. Одним з перших прикладів ігрового рушія був Quake Engine, розроблений компанією id Software у 1996 році для гри Quake. Quake Engine був першим ігровим рушієм, який повністю використовував тривимірну графіку, а не спрощену 2.5D графіку, як попередні ігри Doom та Wolfenstein 3D. Quake Engine також вводив нові можливості, такі як динамічне освітлення, кольорове освітлення, маппінг тіней, мережевий мультиплеєр та модифікація ігор.

Quake Engine став основою для багатьох інших ігрових рушіїв, таких як GoldSrc (Half-Life), Source (Half-Life 2), Unreal Engine (Unreal), id Tech 2 (Quake II), id Tech 3 (Quake III Arena) та інші. З того часу ігрові рушії постійно еволюціонують та вдосконалюються, надаючи розробникам ігор все

більше можливостей та функцій для створення захоплюючих і різноманітних ігрових досвідів.

Ігрові рушії можна розбити на різні види та типи за різними критеріями, такими як:

Вид за призначенням - ігрові рушії можуть бути призначені для створення ігор для розваги, навчання, дослідження, симуляції тощо. Наприклад, Unreal Engine, Unity та Godot призначені для створення ігор для розваги, але також можуть використовуватися для інших цілей. Serious Engine, VBS3 та Unreal Engine 4 Simulation призначені для створення ігор для навчання та симуляції.

Вид за архітектурою - ігрові рушії можуть мати різну архітектуру, яка визначає їх структуру та спосіб взаємодії з іншими компонентами. Наприклад, ігрові рушії можуть мати монолітну архітектуру, коли всі компоненти зв’язані між собою та залежать один від одного. Або вони можуть мати модульну архітектуру, коли кожен компонент є окремою одиницею, яка може бути додана, вилучена або замінена без впливу на інші компоненти.

Тип за орієнтацією на 2D або 3D графіку - 2D ігрові рушіі використовують двовимірну графіку для створення плоских зображень та сцен. 3D ігрові рушіі використовують тривимірну графіку для створення об’ємних моделей та сцен. Деякі ігрові рушіі підтримують обидва типи графіки, а деякі спеціалізуються на одному з них. Наприклад, Unity, Unreal Engine та Godot підтримують як 2D, так і 3D графіку, а RPG Maker, GameMaker Studio та Ren’Py спеціалізуються на 2D графіці.

Тип за жанром або стилем ігри - деякі ігрові рушіі призначені для створення певного жанру або стилю ігри, а деякі є більш універсальними та гнучкими. Наприклад, Adventure Game Studio, Ren’Py та RPG Maker призначені для створення пригодницьких, візуальних новел та рольових ігор

відповідно, а Unity, Unreal Engine та Godot можуть використовуватися для створення ігор різних жанрів та стилів.

Тип за платформою або пристроєм - деякі ігрові рушії підтримують багатоплатформену розробку, дозволяючи створювати ігри для різних пристроїв та операційних систем. Деякі ігрові рушії обмежені однією або кількома платформами. Наприклад, Unity, Unreal Engine та Godot підтримують багатоплатформену розробку для Windows, Linux, macOS, iOS, Android тощо. A RPG Maker, GameMaker Studio та Ren'Py обмежені Windows, Linux та macOS.

Тип за ліцензією або ціною - деякі ігрові рушії є вільними та відкритими програмами з вільною ліцензією, яка дозволяє використовувати і модифікувати рушій без обмежень. Інші ігровіс рушії є пропрієтарними програмами з комерційною ліцензією, яка вимагає сплати певної суми або винагороди за використання. Наприклад, Godot є вільним і відкритим ігровим рушієм з ліцензією MIT, яка дозволяє використовувати та зміснювати його без обмежень. Unity ж є пропрієтарним ігровим рушієм з комерсійною ліцензією, яка дозволяє безкоштовно використовувати для некомерційних проектів з доходом менше 100 000 доларів на рік, але вимагає сплати певної суми за комерційне використання або проекти з доходом вище 100 000 доларів на рік.

Вибір ігрового рушія є одним з найважливіших кроків у процесі розробки гри, оскільки він впливає на графіку, геймплей, продуктивність, сумісність та інші характеристики гри. Різні ігрові рушії мають різні переваги та недоліки, тому розробники повинні обирати ігровій рушій в залежності від своїх цілей та потреб.

Ігрові рушії дозволяють полегшити розробку гри шляхом уніфікації та систематизації її внутрішньої структури, а також надають можливість створення багатоплатформових ігор. Існує багато різних ігрових рушіїв, які

мають свої унікальні технології та значення для геймдеву, а також свої недоліки. Давайте розглянемо деякі з них.

Один з найпопулярніших ігрових рушіїв - Unreal Engine від Epic Games. Це багатоплатформенний ігровий рушій, який підтримує високоякісну графіку, фізику, анімацію, звук та інші аспекти ігрового процесу. Використовується в багатьох жанрах ігор, таких як шутери, пригоди, стратегії, гонки тощо. Деякі приклади ігор на Unreal Engine: Gears of War, BioShock, Fortnite. Перевагами Unreal Engine є високий рівень оптимізації та продуктивності, потужна система скриптингу на мові Blueprint, багато готових компонентів та активна спільнота розробників. Недоліками Unreal Engine є висока складність для новачків, велика вартість ліцензії для комерційних проектів та вимогливість до апаратних ресурсів.

Unreal Engine 5 - це найновіша версія Unreal Engine, одного з найпотужніших ігрових рушіїв, який включає нові функції, основні новинки з них:

Nanite - це віртуалізована геометрична система, яка дозволяє створювати деталізовані сцени з мільярдами полігонів без втрати продуктивності. Nanite динамічно адаптує розмір та кількість полігонів в залежності від розташування камери та роздільності екрану. Nanite також спрощує процес імпорту та оптимізації фотограметричних асетів.

Lumen - це система динамічного освітлення і відбиття світла, яка дозволяє створювати реалістичні сцени з індиректним освітленням у реальному часі. Lumen автоматично адаптує освітлення до змін у сцені, таких як переміщення об’єктів, зміна погоди або часу доби. Lumen також покращує якість та деталізацію тіней і відбиття світла.

World Partition System - це система розподілу світу на логічні частини, яка дозволяє створювати величезні відкриті світи з високою деталізацією. World Partition System автоматично завантажує та вивантажує частини світу в залежності від потреб розробника або гравця. World Partition System також сприяє спільному редагуванню світу за допомогою системи One File Per

Actor.

MetaSounds - це система звукового програмування, яка дозволяє створювати складні звукові ефекти та музику у реальному часі2. MetaSounds дозволяє контролювати параметри звуку за допомогою графового інтерфейсу або скриптингу. MetaSounds також інтегрується з іншими системами Unreal Engine 5, такими як Niagara Visual Effects System та Chaos Physics and Destruction System.

Анимації - Unreal Engine 5 пропонує новий пакет інструментів для створення і редагування анімацій персонажів та об'єктів. До них належать Control Rig, який дозволяє створювати скелетну анимацію за допомогою графічного інтерфейсу, Sequencer, який дозволяє створювати камерні анимації за допомогою нелійного монтажу, Motion Warping, який дозволяє адаптувати анимації до нерегулярної поверхні, та інші.

Unreal Engine використовувався для створення високоякісних візуальних ефектів та анімації для фільмів та серіалів. Наприклад, серіал "Мандалорець" використовує Unreal Engine для створення реалістичних фонових сцен на основі LED-екрана. А також використовується для створення тренажерів, які імітують реальні ситуації та навчають персонал роботи з обладнанням, технологіями та навичками. Наприклад, NASA використовує Unreal Engine для підготовки астронавтів до місій на МКС та Марс.

Ще один популярний ігровий рушій - Unity від Unity Technologies. Це багатоплатформенний ігровий рушій, який дозволяє легко створювати інтерактивний 2D- та 3D-контент2. Має потужну систему скриптів на мові C#, редактор сцен та активний спільноту розробників. Використовується в різних жанрах ігор, таких як інді-гри, казуальні гри, VR-гри тощо. Деякі приклади ігор на Unity: Hearthstone, Ori and the Blind Forest, Among Us. Перевагами Unity є гнучкість та доступність для новачків та професіоналів, безкоштовна ліцензія для особистих проектів та підтримка багатьох платформ. Недоліками Unity є обмеження у функціональності та якості

графіки, нестабильність деяких версій та проблеми з оптимізацією.
Unity має власну мову програмування - C#, яка є об’єктно-орієнтованою та сумісною з .NET Framework. C# дозволяє розробникам писати ефективний та безпечний код, який може бути легко перенесений на різні платформи. Також має величезну спільноту розробників, яка налічує понад 13 мільйонів користувачів по всьому світу. Вони допомагають один одному з питань, порадами та ресурсами на форумах, блогах та соціальних мережах. Має власний онлайн-магазин - Asset Store, де можна купити або продати готові асети, такі як моделі, текстури, звуки, скрипти тощо.
Unity підтримує більше 25 платформ, включаючи Windows, Mac, Linux, iOS, Android, PlayStation, Xbox, Nintendo Switch, Oculus Rift, Steam VR та інші. Це робить Unity одним з найбільш універсальних ігрових рушіїв на ринку.

CryEngine - це потужний ігровий рушій від компанії Crytek, який підтримує високоякісну 3D-графіку, фізику, звук, анімацію та інші аспекти ігрового процесу. Використовується в багатьох жанрах ігор, таких як шутери, пригоди, гонки тощо. Деякі приклади ігор на CryEngine: Far Cry, Crysis, Hunt: Showdown. Перевагами CryEngine є реалістичне освітлення та тіні, динамічна деструкція середовища, гнучка система скриптингу на мові Lua та підтримка багатьох платформ. Недоліками CryEngine є висока складність для новачків, нестабильність деяких версій та вимогливість до апаратних ресурсів.
Гра Crysis була настільки вимогливою до апаратних ресурсів, що стала популярним тестом продуктивності для комп’ютерів. Фраза “Can it run Crysis?” стала мемом у геймерській спільноті, який використовувався для висловлення сумніву щодо можливостей будь-якого комп’ютера запустити гру з високими налаштуваннями. Навіть зараз, після 16 років від релізу гри, деякі сучасні комп’ютери все ще не можуть запустити гру з максимальною деталізацією.

Один з простих у використанні ігровий рушій для створення 2D-гри без необхідності програмування - GameMaker від YoYo Games. Має вбудовану

мову GML для створення логіки гри та редактор спрайтів та звуку. Використовується переважно для створення платформерів, аркад та пазлів. Деякі приклади ігор на GameMaker: Undertale, Hotline Miami, Hyper Light Drifter. Перевагами GameMaker є простота та швидкість для створення 2D-гри, безкоштовна версія для основних функцій та багато навчальних матеріалу. Недоліками GameMaker є обмеження у можливостях 3D-розробки та підтримки платформ, нестабільність деяких версій та слабка оптимізація. GameMaker має власну мову програмування - GameMaker Language (GML), яка є подібною до C та JavaScript. GML дозволяє розробникам писати складну логіку гри та керувати об’єктами, змінними, звуками тощо, та має вбудований інструмент Drag and Drop (DnD), який дозволяє створювати гру без коду за допомогою візуальних блоків. DnD може бути конвертований в GML для подальшого редагування або навпаки.
GameMaker, як і Unity, має власний онлайн-магазин - GameMaker Marketplace. Також на Marketplace, окрім того що можна купити або продати готові асети, такі як спрайти, фони, шрифти, музика, ефекти тощо, можна знайти безкоштовні або платні розширення для GameMaker, які додають нові функції та можливості.

Програмування ігрових рушіїв - це одна з найбільш вимогливих та цікавих галузей програмування, яка вимагає від програмістів високого рівня знань та навичок. Окрім того, щоб опанувати ігровий рушій, треба знання не тільки самого рушія, також потрібно знати мови програмування, таких як C++, C#, Java, які використовуються для створення ігрового коду та логіки. Знання математики, фізики, алгоритмів, структур даних, які використовуються для розв’язання проблем та оптимізації продуктивності. Графічних бібліотек, таких як OpenGL, JOGL, SDL, які використовуються для рендерингу 2D або 3D графіки на екрані. Знання звукових бібліотек, таких як OpenAL, FMOD, Wwise, які використовуються для відтворення звукових ефектів та музики у грі. Та знання мережевого програмування, такого як TCP/IP, UDP, HTTP,

WebSocket, які використовуються для комунікації між гравцями та серверами у мережевих іграх.

Ці навички потрібні програмісту, щоб працювати з ігровими рушіями, оскільки вони дозволяють створювати ігри, які захоплюють, зачаровують та залучають гравців. Програмісти ігрових рушіїв можуть брати участь у всіх етапах ігрового проекту, від концепції до публікації, а також підтримувати і оновлювати ігровий рушії після його запуску на ринок. Програмісти ігрових рушіїв є необхідною складовою геймдев-індустрії, оскільки без них не було б можливо створювати ігри, якими ми насолоджуємося.

2.5 Платформи та мультиплатформеність

Ігрові платформи - це електронні пристрої або програми, які дозволяють запускати відеоігри на різних екранах та з різними способами керування. Вони мають свої переваги та недоліки залежно від характеристик, ціни, доступності ігор та інших факторів.
Аркадні ігрові автомати були першими платформами для відеоігор, які стали популярними у 1970-1980-х роках. Деякими з найвідоміших аркадних ігор були Pong, Space Invaders, Pac-Man, Donkey Kong та інші. Вони зазвичай розміщуються в гральних залах або інших громадських місцях, де гравці повинні заплатити за кожну гру або за певний час гри.

Зараз у світі є чотири найрозповсюдженіших платформи:

ПК - це персональний комп’ютер, який може використовуватися не тільки для ігор, але й для роботи, навчання, розваг тощо. Персональний комп’ютер може мати різну операційну систему, таку як Windows, Linux, MacOS тощо. Персональний комп’ютер став платформою для відеоїгор у 1980-1990-х роках і також продовжує розвиватися до сьогодні. ПК має багато переваг для гравців, таких як: висока продуктивність, налаштування компонентів за бажанням, широкий вибір ігор та платформ (Steam, Epic Games, Origin, Uplay тощо), можливість модифікувати ігри, підтримка різних пристроїв введення

(миша, клавіатура, геймпад тощо). Недоліки ПК: висока ціна на деякі компоненти, необхідність оновлювати драйвери та операційну систему, проблеми з сумісністю або оптимізацією деяких ігор.

Консоль - спеціалізований пристрій для запуску відеоігор на телевізорі або моніторі. Консолей існує багато видів та поколінь, але найпопулярнішими є PlayStation, Xbox та Nintendo. Консолі мають такі переваги: простота використання, стабільність роботи, ексклюзивні ігри (такі як God of War, Halo або Mario), можливість грати в мережеві ігри з друзями та іншими гравцями, підтримка віртуальної реальності та інших нових технологій. Консолі також мають деякі недоліки: висока ціна на сам пристрій та ігри, обмежений вибір ігор та платформ, неможливість модифікувати або налаштовувати ігри, залежність від підписок та інтернет-з’єднання для деяких функцій.

Портативна ігрова консоль - це пристрій, який дозволяє грати в ігри без підключення до телевізора або монітора. Портативна ігрова консоль має вбудований екран, акумулятор, контролери та слот для картриджів або дисків.

Портативна ігрова консоль може бути використана як самостійний пристрій або підключена до телевізора для гри на великому екрані. Деякими з найвідоміших портативних ігрових консолей є Nintendo Switch, Nintendo Switch Lite, PlayStation Portable, PlayStation Vita та інші.

У портативної консолі такі переваги: дозволяє грати в ігри будь-де і будь-коли, не залежачи від наявності телевізора або розетки, часто має ексклюзивні ігри, які не доступні на інших платформах (наприклад, Nintendo Switch має такі ексклюзивні ігри, як The Legend of Zelda: Breath of the Wild, Animal Crossing: New Horizons, Super Mario Odyssey та інші), портативна ігрова консоль може підтримувати багатокористувацький режим, коли кілька гравців можуть грати разом на одному пристрої або підключитися до інших пристроїв через бездротове з’єднання (наприклад, Nintendo Switch дозволяє

грати до чотирьох гравців на одному екрані або до восьми гравців через локальне з'єднання).

Та має наступні недоліки: портативна ігрова консоль має компроміс між потужністю та автономністю, тому консоль не може забезпечити таку ж високу якість графіки, звуку та швидкості завантаження, як стаціонарна консоль або ПК. Портативна ігрова консоль працює на акумуляторі, який має обмежений час роботи, та портативна ігрова консоль не сумісна з іграми та аксесуарами з інших платформ.

Смартфон - це мобільний пристрій, який може виконувати багато функцій, в тому числі запускати відеоігри. Смартфони стали дуже популярними для гравців через їх портативність, доступність, розмаїття ігор та платформ (Google Play, App Store тощо), можливість грати в будь-який час і в будь-якому місці. Недоліки смартфонів: низька продуктивність, обмежена ємність батареї та пам'яті, не зручне керування сенсорним екраном для деяких жанрів ігор, проблеми з безпекою та конфіденційністю даних.

До цієї великої четвірки потихеньку прямує п'ятий гравець - **хмарні ігрові сервіси**. Хмарний геймінг є перспективним трендом останніх років, оскільки він дозволяє грати в ігри через Інтернет без необхідності завантажувати їх на своє обладнання. Хмарні ігрові сервіси запускають ігри на своїх серверах і транслюють кожен кадр на пристрій гравця. Це дозволяє грати в сучасні ігри на будь-якому пристрої, який має екран та підключення до Інтернету, навіть якщо вони не мають потужної апаратної частини. Деякими з найвидатніших хмарних ігрових сервісів є NVIDIA GeForce Now, PlayStation Now, Google Stadia, Amazon Luna та інші.

Мультиплатформа в іграх - це можливість запускати одну і ту ж відеогру на різних ігрових платформах, таких як ПК, консолі, смартфони тощо. Мультиплатформа має свої переваги та недоліки як для розробників, так і для гравців.

Переваги мультиплатформи:

- Збільшення потенційної аудиторії ігри, оскільки вона доступна для більшої кількості користувачів різних пристроїв.
- Збереження єдиності сюжету, геймплею та графіки ігри на всіх платформах, що дозволяє гравцям отримати однаковий досвід від гри незалежно від їх вибору пристрою.
- Можливість синхронізації прогресу та даних гравця між різними платформами, що зручно для тих, хто хоче продовжити гру на іншому пристрої.
- Можливість кросплатформенної мережевої гри, коли гравці з різних платформ можуть грати разом або проти один одного в одну і ту ж гру.

Недоліки мультиплатформи:

- Складність та витратність розробки та оптимізації ігри під різні платформи, оскільки кожна з них має свої особливості, обмеження та вимоги.
- Ризик зниження якості або продуктивності ігри на деяких платформах через несумісність або неадекватну адаптацію до їх характеристик.
- Розбіжності у керуванні або інтерфейсі ігри на різних платформах, що може впливати на зручність та справедливість гри.
- Неможливість повноцінно передати всю атмосферу та ефекти ігри на менш потужних або обмежених платформах, таких як смартфони або браузери.

Обирання платформи для відеоігор може бути важливим, оскільки кожна платформа має свої особливості, переваги та недоліки. Залежно від вашого стилю гри, бюджету, уподобань та очікувань, ви можете вибрати платформу, яка найкраще підходить вам.

2.6 Феномен VR та AR

Дві технології, які дозволяють користувачам переживати цифровий контент у змодельованих і реальних середовищах - VR та AR. Почнемо по порядку:

VR скорочення від віртуальна реальність (**virtual reality**). VR працює за допомогою спеціального обладнання, яке створює ілюзію присутності в комп'ютерно симульованому середовищі. Користувач може спостерігати, рухатися і взаємодіяти з об'єктами в VR. Він є феноменом у світі геймдеву, тому що він дає можливість створювати ігри з більшою реалістичністю, іммерсією і інноваційністю. Найпопулярнішими гарнітурами для VR є Oculus Rift, HTC Vive, PlayStation VR і Samsung Gear VR.

VR може використовуватися для розробки різних жанрів ігор, від казуальних до хардкорних, від симуляторів до пригодницьких. VR також може допомогти гравцям пережити нові емоції, враження і ситуації, які неможливо або небезпечно пережити в реальному житті.

Однак VR також має свої недоліки. По-перше, VR потребує досить дорогого та складного обладнання, яке не доступне для всіх гравців. По-друге, VR може викликати побічні ефекти, такі як запаморочення, нудота, головний біль або кіберзалежність. По-третє, VR ще не досягнув свого повного потенціалу у технологічному та творчому планах, тому що він ще знаходиться на етапі розвитку і дослідження

VR-програмування також вимагає вибору підходящого VR-двигуна, або VR-game engine, який надає розробникам фреймворк для створення VR-ігор. VR-двигун містить VR-SDK (virtual reality - software development kit: набір інструментів, які дозволяють розробникам проектувати, створювати і тестувати VR-додатки. VR-SDK зазвичай містить бібліотеки, API, документацію, приклади коду і т.д.), який дозволяє розробникам проектувати, будувати і тестувати свої ігри. Деякі з найпопулярніших VR-двигунів - це Unity, Unreal Engine, 3ds Max Design. Також потребує розуміння

специфіки VR-платформи, для якої розробляється гра або додаток. Різні VR-платформи мають різне обладнання, операційні системи, API і стандарти. Деякі з найпоширеніших VR-платформ - це Oculus Rift, HTC Vive, PlayStation VR і Samsung Gear VR.

У свою чергу **AR** - це скорочення від доповненої реальності (a**ugmented reality**), яка є технологією, що накладає цифровий вміст на реальний світ, який ви бачите через камеру смартфона або спеціальні окуляри. AR працює за допомогою AR Foundation, який є фреймворком для розробки AR-додатків на різних платформах, таких як Android, iOS, Unity та Web. AR Foundation надає основні функції для створення AR-досвіду, такі як відстеження руху, якоря, розуміння навколишнього середовища, оцінка глибини та освітлення та інші.

У світі геймдеву ця технологія дозволяє створювати ігри, які залучають гравців до інтерактивної взаємодії з реальним світом. Наприклад, Pokemon Go - це популярна AR-гра, яка дозволяє гравцям полювати на покемонів у своєму місцевому середовищі за допомогою камери смартфона. Іншим прикладом є Minecraft Earth, яка дозволяє гравцям будувати та досліджувати 3D-світи у реальному масштабі.

Однак AR також має деякі недоліки, такі як:

- Високий розряд батареї смартфона або окулярів через постійне використання камери та GPS.
- Обмежена якість та точність визначення об’єктів та поверхонь у реальному світі.
- Потенційні проблеми з безпекою та конфіденційністю даних через доступ до камери та місцезнаходження користувача.
- Потенційні проблеми з безпекою та здоров’ям користувача через втрату уваги до реального середовища або перевтомлення очей.

Найбільшим ринком для AR є Китай, який очікується досягне 18 млрд доларів США до 2025 року. Китайські компанії активно інвестують у доповнену реальність для електронної комерції, освіти, туризму та ігор. Для розробки AR-додатків потрібно використовувати спеціальні інструменти та мови програмування, які підтримують AR-функції, такі як відстеження руху, розуміння навколишнього середовища, оцінка глибини та освітлення тощо.

Тож феномен VR та AR полягає в тому, що ці технології змінюють спосіб сприйняття та взаємодії з реальністю за допомогою цифрових зображень, звуків, текстур та інформації. VR та AR дозволяють користувачам переживати, впливати та створювати нові світи або доповнювати існуючі. Віртуальна та доповнена реальності використовуються для різних цілей, таких як розваги, навчання, медицина, промисловість, реклама тощо. Привертають увагу різних аудиторій, особливо молодих поколінь, які цінують інновації, емоції та соціальну взаємодію. VR та AR є феноменом, трендом і необхідністю у наш час, оскільки вони надають нових можливостей для навчання, співпраці, комунікації та розвитку, та продовжують розвиватися та вдосконалюватися завдяки новим пристроям, платформам, інструментам та контенту.

2.7 Класифікація ігор за бюджетом

Бюджет впливає на розробку ігор у багатьох аспектах. Зазвичай, чим вищий бюджет, тим більше можливостей для розробників створити якісну графіку, звук, сюжет, геймплей і інші елементи ігри. Бюджет також впливає на кількість людей, які працюють над проектом, терміни розробки, рекламу і просування ігри на ринку. Бюджет також визначає ризик для розробників і видавців: якщо гра не виправдає очікувань або не знайде свою аудиторію, то вони можуть зазнати великих збитків.

За бюджетом ігри можна поділити на різні класи:

ААА-ігри - це високобюджетні ігри, які створюються великими видавництвами з великими коштами на розробку і рекламу. Це високоякісні ігри з деталізованою графікою і різноманітним ігровим процесом. Цей термін відповідає терміну блокбастер з кіноіндустрії. Приклади ААА-ігор: Halo, Call of Duty, Assassin's Creed, Grand Theft Auto. Бюджет таких ігор може досягати кількох сотень мільйонів доларів1. Типово ціна на один примірник ААА-гри складає від $40 до $60.

АА-ігри - це середньобюджетні ігри, які мають менше коштів на розробку, але все ще забезпечують високий рівень якості і оригінальності. Це часто ігри середнього розміру або незалежні проекти з певною нішею аудиторії. Приклади АА-ігор: Resident Evil, Tomb Raider, The Witcher, Life is Strange. Бюджет таких ігор може коливатися від декількох мільйонів до десятків мільйонів доларів. Типово ціна на один примірник АА-гри складає від $20 до $40.

А-ігри - це низькобюджетні ігри, які створюються малими або середніми студіями з обмеженими коштами на розробку і рекламу. Це ігри, які мають просту графіку, короткий час гри і низьку ціну. А-ігри часто засновані на класичних жанрах або наслідують популярні ААА-ігри. Приклади А-ігор: Serious Sam, Max Payne, Postal, Worms1. Бюджет таких ігор може бути від декількох сотень тисяч до декількох мільйонів доларів. Типово цена на один примирник А-гра складає від $10 до $20.

Інді-ігри - це низькобюджетні ігри, які створюються невеликими командами або одними розробниками без підтримки великих видавництв. Це ігри, які зазвичай мають просту графіку, короткий час гри і низьку ціну. Інді-ігри часто експериментують з новими ідеями, жанрами і механіками. Приклади інді-игор: Minecraft, Stardew Valley, Undertale, Limbo3. Бюджет таких ігор може бути дуже низьким або нульовим3. Типова ціна на один примирник інді-гри складає від $0 до $20.

Наскільки бюджет впливає на розробку, та чи можуть низькобюджетні ігри бути більш успішними ніж ААА?
Низький бюджет не завжди означає низьку якість або невдачу. Існують приклади ігор, які були створені невеликими командами або одними розробниками з мінімальними коштами, але стали дуже успішними та популярними. Тіж інді-ігри часто вирізняються своєю оригінальністю, креативністю і інноваційністю. Прикладами таких ігор є тіж самі: Minecraft, Stardew Valley, Undertale, Limbo. Ці ігри заробили мільйони доларів і отримали високу оцінку від критиків і гравців.

Таким чином, можна сказати, що бюджет є важливим фактором для розробки ігор, але не єдиним. Існують інші фактори, які також впливають на успіх гри, такі як: якість сценарію, геймплею, дизайну; цільова аудиторія; конкуренція на ринку; маркетингова стратегія; випадковий фактор тощо. Тому низькобюджетні ігри можуть бути більш успішними ніж ААА-ігри за умови, що вони задовольнять потреби та інтереси гравців і матимуть достатньо видимості на ринку.

2.8 Типи гри за кількістю гравців

Відеоігри можна розподілити за кількістю гравців на такі типи:

Одноосібні ігри - це ігри, в яких гравець керує одним або декількома персонажами в ігровому світі без взаємодії з іншими гравцями. Приклади: Super Mario Bros., The Witcher 3, Tetris.

Багатоосібні ігри - це ігри, в яких одночасно грають від двох і більше гравців. Цей режим може бути реалізований на одному комп’ютері (hot-seat), на декількох комп’ютерах через локальну мережу (LAN) або через інтернет. Багатоосібні ігри можуть мати різну структуру та цілі, наприклад:

- ***Кооперативні ігри*** - це ігри, в яких гравці грають у команді проти комп'ютерних супротивників або складних завдань. Приклади: Portal 2, Left 4 Dead 2, Minecraft.
- ***Конкурентні ігри*** - це ігри, в яких гравці змагаються між собою за рахунок, час, ресурси або інші параметри. Приклади: Counter-Strike, FIFA, Mario Kart.
- ***Асиметричні ігри*** - це різновид багатоосібної гри, коли гравці умовно діляться на команди (або діють окремо) й мають різну мету. До прикладу, у грі Dead by Daylight на обмеженому за розмірами рівні група гравців ховається від іншого гравця й має докласти зусиль, щоби вижити та втекти. Гравець-мисливець повинен знайти гравців та знешкодити їх.
- ***Масові багатокористувацькі ігри*** - це ігри, в яких одночасно беруть участь сотні або тисячі гравців в одному великому ігровому світі. Такі ігри часто мають рольовий характер та побудовані на соціальній взаємодії. Приклади: World of Warcraft, EVE Online, Roblox.

Тип гри за кількістю гравців впливає на створення гри в різних аспектах, таких як дизайн, програмування, тестування та підтримка. Програміст грає важливу роль у цьому процесі, оскільки він відповідає за написання та оптимізацію коду гри, створення ігрової механіки, реалізацію мережевої взаємодії та інтерфейсу. Залежно від типу гри за кількістю гравців, програміст може зіткнутися з різними викликами та завданнями.

РОЗДІЛ 3. ПРАКТИЧНА РОЗРОБКА ВІДЕОГРИ

3.1 Прототип

Настав момент описати хід розробки відеогри для дипломної роботи.
Усе почалося з бажання зробити гру у жанрі метроїдванії під впливом іншої гри - Hollow Knight, яка була випущена інді-студією Team Cherry 24 лютого

2017 року. Ця гра приваблювала своєю атмосферою, музикою, геймплеєм, дизайном та стараннями самих розробників.

Наша команда складається з програміста-художника та режисера, який виконує функцію сценариста та помічника по матеріалам для сюжету та наративу. На початку були роздуми про те, де будуть відбуватися події гри та хто стане головним персонажем, про що буде сама гра.
Роздуми досить швидко привели до японського сетингу, де головною героїнею є персонаж японської міфології - кіцуне, а місцем подій магічний ліс де є дерева сакури, наповнений елементами синтоїзму. Ще одним варіантом була побудова сетингу, сюжету та персонажів на базі слов'янської міфології, але від цієї ідеї швидко відмовились, тому що було дуже важко встановити точного головного героя, який би органічно вписувався у історію. Так почались нариси подій, внутрішніх правил всесвіту гри та персонажів.

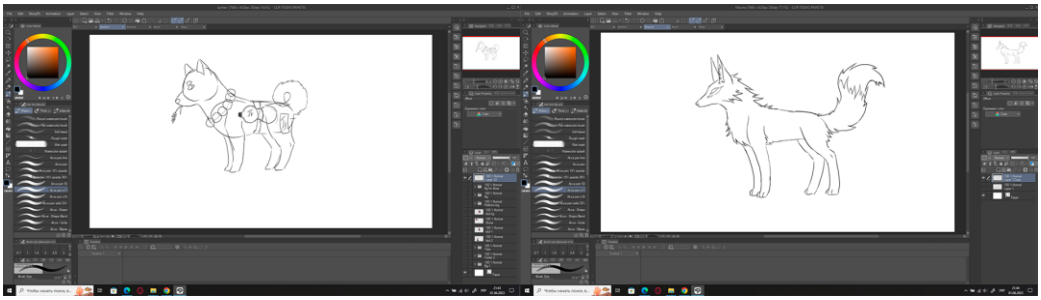
Потім було переорієнтування жанру на платформер з елементами рольової системи, тому що спершу масштаби гри, як за сюжетом, візуалом, так і геймплеєм, орієнтувалися на довгострокову розробку з додаванням бюджету на потреби, по типу музики або реклами.
Ця гра йде прототипом того, що ми хочемо згодом розширити у всіх напрямках, тому була обрана назва "Project Kitsune", а конкретно з підзаголовком "Prototype".
"Project Kitsune: Prototype" буде мати сюжетне відношення до наступних ігор цієї серії, що вже зараз потихеньку розробляються. Тому що наративи та сюжет можуть відгукуватись у майбутніх іграх, що розширяє історію, а це подобається геймерам.

3.2 Розробка

Для початку був обраний ігровий рушій - Unity, тому що він використовує мову програмування C# та є легким для програмістів з достатнім рівнем

знань, при цьому маючи багато корисних функцій та візуальне програмування, до того ж він безкоштовний.

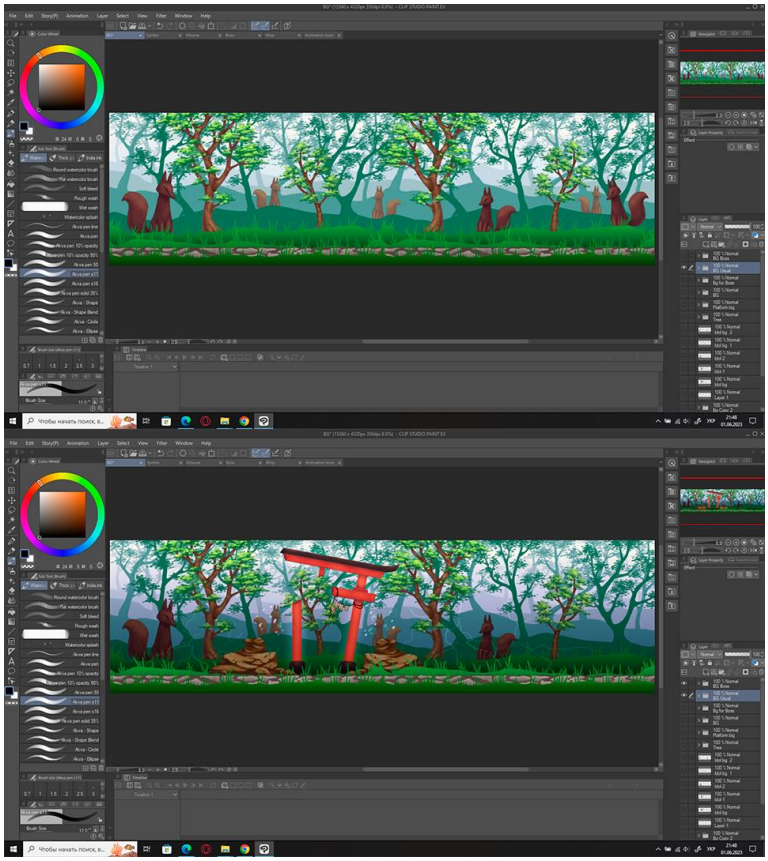
Далі починається розробка дизайнів персонажів та локацій. Для цього ми обрали Clip Studio Paint де багатий функціонал для малювання спрайтів (двомірне зображення, що застосовується в комп'ютерній графіці) та фону. І не виходячи з програми ці елементи можна анімувати.



(Рис. 3.1 Скріншот процесу роботи над персонажами)

Бекграунд для гри ми вирішили зробити у вигляді зеленого лісу. Серед дерев ми поставили кам’яні статуї, які є частиною коротенької історії Prototype.

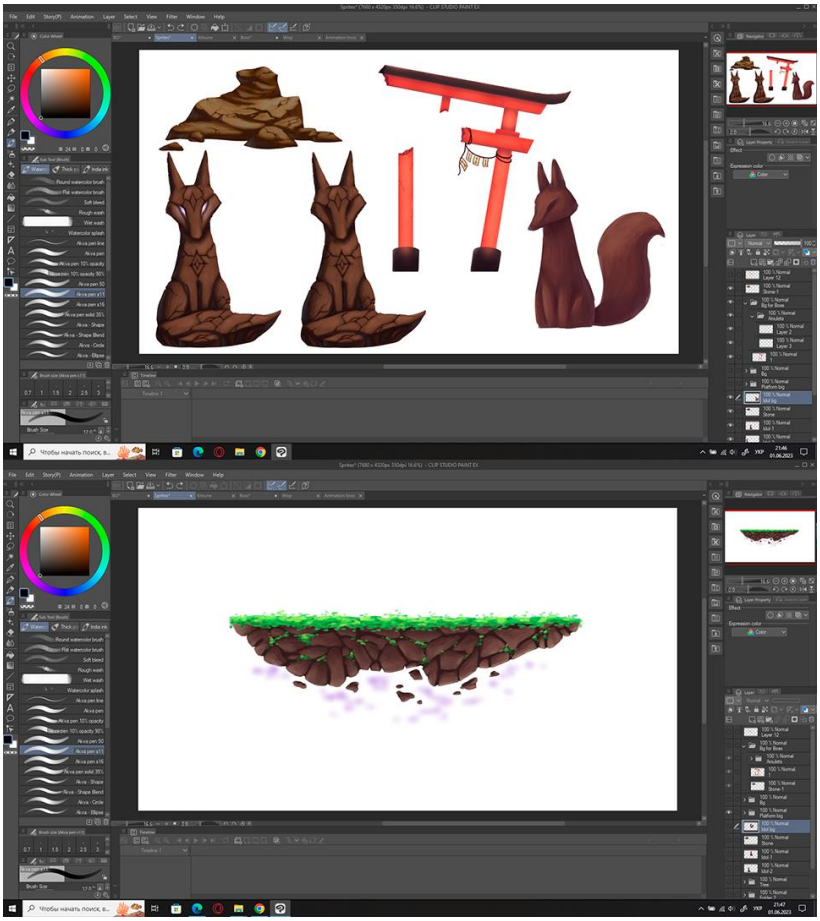
У локацію з босом ми додали невеликий елемент у вигляді торії (ворота перед синтоїстськими святилищами), на ній є порвана мотузка де залишились усього чотири захисні амулети. Це число обране, тому що в японській мові цифра 4 звучить «сі», «ши», що співзвучно зі словом «смерть». Тому все що містить четвірку вважається нещасливим. Така маленька деталь добре грає на наратив гри. Також ми змінили трішки кольори, щоб додати трішки атмосфери.



(Рис. 3.2 Скріншот основної локації та локації з босом)

Усі елементи на екрані, окрім доріжки та задніх дерев (які не мають деталей), є спрайтами, та технічно можуть бути інтерактивними. На локаціях будуть окрема статуя лисиці, що буде виділятися на фоні інших, і через них буде реалізована механіка зберігання. Тож, якщо персонаж помре, геймер зможе продовжити з останньої статуї.

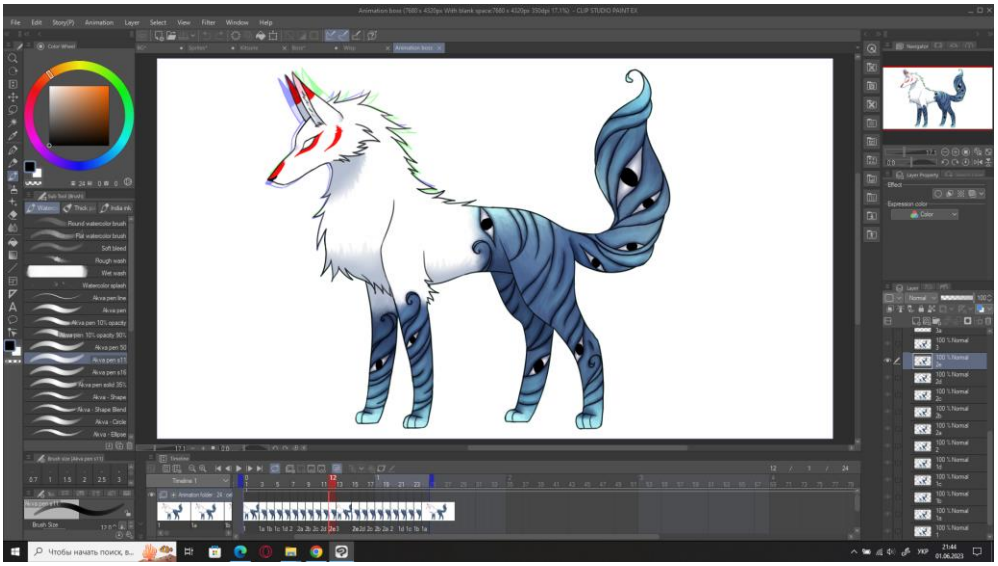
Звісно ж на локаціях будуть літаючі платформи, по яким можна та треба буде стрибати. Та як плюс, їх паріння у просторі є теж наративним елементом історії.



(Рис. 3.3 Фонові та активні спрайти)

Противниками геймера будуть сюрреалістичні створіння, яких поширено називають - “втіленнями вбивці”, вони є душами що керують штучними тілами. Ці душі вбивць, або тих чий розум заповнився жагою до вбивств (тобто при житті вони могли нікого і не вбити).

Фінальним випробуванням для геймера буде бос - кіцуне Момока, яка потрапила під вплив сил, що затьмарили їй розум, а тіло змусили мутувати.



(Рис. 3.4 Процес анімації Момоки)

Усі анімації реалізовані у тій самій Clip Studio Paint. Найбільше анімацій отримала головна героїня гри, тому що вона має більший мувсет (набір рухів), ніж інші створіння.

На разі основне навантаження при розробці прототипу є дизайн, створення спрайтів та анімації, тому що це вимагає багато годин творчої роботи та малювання. Деякі деталі, або повноцінні образи, вимагали переробки кольорів чи з нуля для того, щоб підв’язати візуал до сюжету та наративу.

Звуки для прототипу ми шукаємо у мережі, як і усі інші додаткові матеріали, які ми не можемо зробити без окремих людей у команді.

На виході ми маємо прототип майбутнього інді-проекту, де тестуємо свої сили у початкових механіках платформингу, анімації та сюжетно-наративній складовій. Платформою прототипу є персональні комп’ютери для облегшення розповсюдження і збору фідбеку від майбутньої аудиторії.

ВИСНОВКИ

Геймдев - це процес дизайну, розробки та релізу гри1. Ця індустрія має великий вплив на розвиток сучасних технологій, оскільки вона постійно стикається з технічними викликами та очікуваннями гравців. Геймдев використовує різні мови програмування, двигуни, платформи та інструменти для створення якісного контенту.

Роль програмістів у геймдеві дуже важлива, оскільки вони відповідають за реалізацію ігрової механіки, графіки, анімації, звуку, інтерфейсу та інших аспектів гри. Програмісти можуть мати різні спеціалізації, такі як Gameplay Developers, Engine Developers, Animation Developers, C++ Developerй, Unity Developer, UI/UX Designer тощо. Програмісти також співпрацюють з іншими фахівцями у геймдев-команді, такими як гейм-дизайнери, художники, сценаристи тощо.

Геймдев для програмістів є дуже перспективним напрямком, оскільки він дає можливість реалізувати свої творчі ідеї, використовувати різні мови програмування та інструменти, працювати над цікавими та складними проектами. Світ відеоігор також вимагає постійного саморозвитку та навчання, оскільки ігрова індустрія швидко змінюється та впроваджує нові технології. Кожна з цих професій має свої особливості, скіли, навички та перспективи. Програмісти у геймдеві також можуть працювати в різних форматах: великі студії, інді-команди, фріланс тощо. Це є для програмістів не тільки роботою, але й захопленням, яке дозволяє створювати унікальні ігрові світи та дарувати емоції гравцям.

Програміст може бути геймдизайнером, якщо він має не тільки технічні навички, але й творче мислення, уяву, аналітичні здібності та розуміння ігрової індустрії. Такий симбіоз може вплинути на якість ігрового процесу, оскільки програміст-геймдизайнер може краще розуміти можливості та обмеження ігрового рушія, оптимізувати код та ресурси та інструменти для реалізації своїх ідей. Він також може мати перевагу на ринку праці, оскільки

він може виконувати роль як програміста, так і геймдизайнера у різних проектах, і повинен балансувати між технологічною та творчою сторонами своєї професії, що є викликом для такого симбіозу.

Сфера геймдеву активно сприяє розвитку сучасних технологій за рахунок пошуку нових рішень для створення захоплюючих ігор. Програмісти у геймдеві мають велику роль у цьому процесі, оскільки вони забезпечують функціональну та естетичну складову гри. Програмування у геймдеві вимагає не тільки технічних навичок, але й креативності та командної роботи.

ВИКОРИСТАНІ ДЖЕРЕЛА

1. https://gamedev.dou.ua/
2. https://www.reddit.com/r/gamedev/
3. https://www.jetbrains.com/
4. https://www.gamedeveloper.com/blogs
5. https://indiegamedev.net
6. https://www.freecodecamp.org/news
7. https://www.gamedesigning.org
8. https://www.gdcvault.com
9. https://www.gamedev.net
10. https://unity.com
11. https://www.unrealengine.com/en-US
12. https://godotengine.org/
13. https://www.yoyogames.com/gamemaker
14. https://www.cryengine.com/
15. https://www.history.com/topics/inventions/history-of-video-games
16. https://en.wikipedia.org/wiki/History_of_video_games
17. https://www.si.edu/spotlight/the-father-of-the-video-game-the-ralph-baer-prototypes-and-electronic-games/video-game-history
18. https://www.youtube.com/watch?v=aWK3tOD-_co&list=PLetOKfueQ61BOjh5qAy97MhugwrELBQmr&index=2
19. https://www.prepar3d.com/
20. https://en.wikipedia.org/wiki/List_of_video_game_genres
21. https://www.techtarget.com/whatis/definition/augmented-reality-gaming-AR-gaming

ДОДАТОК

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Hero : MonoBehaviour
{
    [SerializeField] private float speed = 6f;
    [SerializeField] private int lives = 3;
    [SerializeField] private float jumpForse = 15f;
    private bool isGraunded = false;

    public bool isAttacking = false;
    public bool isRecharget = false;

    public Transform attackPos;
    public float attackRange;
    public LayerMask enemy;

    private Rigidbody2D rb;
    private Animator anim;
    private SpriteRenderer sprite;
```

```
public static Hero Instance { get; set; }

private States State
{
    get { return (States)anim.GetInteger("state"); }
    set { anim.SetInteger("state", (int)value); }
}

private void Awake()
{
    Instance = this;
    rb = GetComponent<Rigidbody2D>();
    anim = GetComponent<Animator>();
    sprite = GetComponentInChildren<SpriteRenderer>();
    isRecharget = true;
}

private void FixedUpdate()
{
    CheckGround();
}

private void Update()
```

```
    {
        if (isGraunded) State = States.idle;

        if (Input.GetButton("Horizontal"))
            Run();
        if (isGraunded && Input.GetButtonDown("Jump"))
            Jump();
        if (Input.GetButtonDown("Fire1"))
            Attack();
    }

    private void Run()
    {
        if (isGraunded) State = States.run;

        Vector3 dir = transform.right * Input.GetAxis("Horizontal");
        transform.position = Vector3.MoveTowards(transform.position,
transform.position + dir, speed * Time.deltaTime);
        sprite.flipX = dir.x < 0.0f;
    }

    private void Jump()
    {
```

```
    rb.AddForce(transform.up * jumpForse, ForceMode2D.Impulse);
}

private void CheckGround()
{
    Collider2D[] collider = Physics2D.OverlapCircleAll(transform.position, 0.3f);
    isGraunded = collider.Length > 1;

    if (!isGraunded) State = States.jump;
}

public void GetDamage()
{
    lives -= 1;
    Debug.Log(lives);
}

private void Attack()
{
    if (isGraunded && isRecharget)
    {
        State = States.attack;
        isAttacking = true;
```

```
            isRecharget = false;

            StartCoroutine(AttackAnimation());
            StartCoroutine(AttackCoolDown());
        }
    }

    private void OnAttack()
    {
        Collider2D[] collider = Physics2D.OverlapCircleAll(attackPos.position,
attackRange, enemy);
        for (int i = 0; i < collider.Length; i++)
        {
            collider[i].GetComponent<Entity>().GetDamage();
        }
    }

    private IEnumerator AttackAnimation()
    {
        yield return new WaitForSeconds(0.3f);
        isAttacking = false;
    }
```

```
    private IEnumerator AttackCoolDown()
    {
        yield return new WaitForSeconds(0.4f);
        isRecharget = true;
    }

}

public enum States
{
    idle,
    attack,
    jump,
    run
}
```

Entity

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Entity : MonoBehaviour
```

```
{
    public virtual void GetDamage()
    {

    }

    public virtual void Die()
    {
        Destroy(this.gameObject);
    }
}
```

Mob

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Mob : Entity
{
    [SerializeField] private float speed = 3.5f;
    [SerializeField] private Vector3 dir;
    [SerializeField] private SpriteRenderer sprite;
```

```
    [SerializeField] private int lives = 3;


    private void Start()
    {
        dir = transform.right;
    }


    private void Update()
    {
        Move();
    }
    private void Move()
    {
        Collider2D[] colliders = Physics2D.OverlapCircleAll(transform.position +
transform.up * 0.1f + transform.right * dir.x * 0.7f, 0.1f);


        if (colliders.Length > 0) dir *= -1f;
        transform.position = Vector3.MoveTowards(transform.position,
transform.position + dir, Time.deltaTime);
    }


    private void OnCollisionEnter2D(Collision2D collision)
```

```
    {
        if (collision.gameObject == Hero.Instance.gameObject)
        Hero.Instance.GetDamage();
    }
}
```