МІНІСТЕРСТВО КУЛЬТУРИ УКРАЇНИ

НАЦІОНАЛЬНА АКАДЕМІЯ КЕРІВНИХ КАДРІВ КУЛЬТУРИ І МИСТЕЦТВ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ПЕРФОМАТИВНИХ МИСТЕЦТВ
Кафедра музичного мистецтва

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття освітнього ступеня «Магістр»

на тему:

«Особливості роботи звукорежисера під час створення музично-звукового образу відеогри»

Виконав:
студент ІІ курсу магістратури, групи МММ-23-24,
спеціальності 025 «Музичне мистецтво»
Цимбалюк Вадим Валентинович

Керівник: кандидат педагогічних наук, професор, Заслужений діяч мистецтв України
Бєлявіна Н.Д.
Рецензент: професор, доктор мистецтвознавства, професор кафедри музичного виховання Київського національного університету театру, кіно і телебачення ім. І.К.Карпенка-Карого **Станіславська К.І.**

Допустити до захисту:
протокол засідання кафедри
від « » листопада 2025р. №
Завідувач кафедри _____
(_____)_____)

КИЇВ-2025

ЗМІСТ

АНОТАЦІЯ

ABSTRACT

ВСТУП……5

РОЗДІЛ 1. **ІСТОРИКО-ТЕОРЕТИЧНІ ЗАСАДИ ДОСЛІДЖЕННЯ**…….11

1.1 Огляд наукових досліджень та літератури……...…11

1.2 Історичний розвиток відеоігор та еволюція ігрового звуку……15

1.3. Етапи розвитку комп'ютерних ігор та їх техніко-художні трансформації……..29

РОЗДІЛ 2. САУНД-ДИЗАЙН У КОМ'ЮТЕРНИХ ІГРАХ……. ……..44

2.1. Основні функції звукового оформлення ігрового середовища…..44

2.2. Характерні особливості інтерактивної музики у відеоіграх.....47

РОЗДІЛ 3. ІМПЛЕМЕНТАЦІЯ ЗВУКОВОГО ДИЗАЙНУ ТА ІНТЕРАКТИВНОЇ МУЗИКИ У ІГРОВИЙ ПРОТОТИП.....51

3.1. Створення ігрового прототипу у середовищі Unity.....51

3.2. Розробка та інтеграція звукового дизайну з використанням FMOD Studio…….…….73

3.3. Композиція та інтерактивна імплементація звукового супроводу у FMOD Studio.....80

3.4 Написання та імплементація музичного супроводу……90

ВИСНОВКИ.....92

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....95

ДОДАТКИ.....99

АНОТАЦІЯ

Кваліфікаційна робота присвячена комплексному дослідженню специфіки діяльності звукорежисера в індустрії відеоігор. У роботі проаналізовано еволюцію ігрового аудіо від найпростіших 8-бітних сигналів до сучасних систем імерсивного просторового звучання. Розкрито роль технічного прогресу як каталізатора художніх трансформацій у звуковому дизайні, що дозволило відеоіграм наблизитися до рівня кінематографічного мистецтва.

Теоретико-методологічну базу дослідження склали праці провідних вітчизняних науковців. Зокрема, на основі робіт В. Дьяченка розглянуто трансформацію професії звукорежисера в сучасному медіапросторі, дослідження К. Станіславської дозволили проаналізувати відеоігри в контексті видовищних форм культури, а праці Л. Рязанцева стали підґрунтям для вивчення технологічних аспектів звукозапису та звукорежисури.

У дослідженні висвітлено основні функції звукового оформлення ігрового середовища: атмосферну, ідентифікаційну, комунікативну, інтерактивну та імерсійну. Особливу увагу приділено феномену інтерактивної музики, яка функціонує як гнучка адаптивна система, що реагує на дії гравця та зміни ігрового контексту.

Практична значущість роботи полягає у створенні власного ігрового прототипу на базі рушія Unity (сцена «Viking Village») та розробці для нього повного циклу звукового дизайну. Автором здійснено запис та обробку оригінальних звукових семплів, написано авторський музичний супровід та реалізовано їх технічну імплементацію за допомогою аудіо-мідлверу FMOD Studio. Продемонстровано методи створення адаптивних аудіосистем: залежність звуку кроків від типу поверхні, динамічні системи вітру та океану, а також вертикальне ремікшування музики.

ABSTRACT

The qualification paper is dedicated to a comprehensive study of the specifics of the sound engineer's activity in the video game industry. The work analyzes the evolution of game audio from the simplest 8-bit signals to modern immersive spatial sound systems. It reveals the role of technical progress as a catalyst for artistic transformations in sound design, which has allowed video games to approach the level of cinematic art.

The theoretical and methodological basis of the research is formed by the works of leading domestic scholars. In particular, based on the works of V. Diachenko, the transformation of the sound engineer profession in the modern media space is examined; the studies of K. Stanislavska allowed for the analysis of video games within the context of spectacular forms of culture, while the works of L. Ryazantsev served as a foundation for studying the technological aspects of sound recording and sound engineering.

The research highlights the main functions of sound design in the gaming environment: atmospheric, identifying, communicative, interactive, and immersive. Special attention is paid to the phenomenon of interactive music, which functions as a flexible adaptive system that responds to player actions and changes in the game context.

The practical significance of the work lies in the creation of a custom game prototype based on the Unity engine ("Viking Village" scene) and the development of a full cycle of sound design for it. The author performed the recording and processing of original sound samples, composed an original musical score, and realized their technical implementation using the audio middleware FMOD Studio. Methods for creating adaptive audio systems were demonstrated, including the dependence of footstep sounds on surface type, dynamic wind and ocean systems, as well as vertical music remixing.

ВСТУП

Актуальність дослідження. З розвитком індустрії відеоігор та аудіовізуальної сфери робота звукорежисера набула нових форм і специфікацій. Звукорежисер відеоігор, або ігровий аудіо-дизайнер, – це одна з наймолодших і найдинамічніших професій у сучасному медіа-середовищі. За останні чотири десятиліття звукове оформлення у відеоіграх зробило стрімкий ривок від примітивних монофонічних сигналів до складних багатоканальних систем просторового звуку, що не поступаються високобюджетним кінематографічним проєктам.

Еволюція ігрового аудіо підтверджує загальну тенденцію розвитку професії, виявлену В. Дьяченком [6]: відбувається трансформація від суто технічного фіксування звуку до творчого моделювання акустичної реальності, де звукорежисер стає повноправним творцем художнього образ. Це підтверджує тезу К. О'Доннелла про те, що сучасна розробка ігор є складною соціотехнічною системою, де фахівець повинен розуміти не лише свій вузький профіль, а й загальну архітектуру проєкту [32].

Мета дослідження – комплексно охарактеризувати особливості роботи звукорежисера у процесі створення звукового оформлення відеоігор, розглянувши історичний контекст, теоретичні аспекти та практичну реалізацію звукового дизайну на прикладі авторського ігрового прототипу.

Об’єкт – процес моделювання звукового образу сучасної інтерактивної відеоігри.

Предмет дослідження – робота звукорежисера над розробкою та імплементацією інтерактивного звукового дизайну у відеогру

Задачі дослідження:

- вивчити джерельну базу дослідження, систематизувати наукові праці та професійні матеріали з ігрового аудіо-дизайну;

- здійснити історичний огляд розвитку індустрії відеоігор, проаналізувати етапи еволюції комп'ютерних ігор та їх техніко-художні трансформації;
- дослідити основні функції звукового оформлення ігрового середовища та його вплив на імерсію гравця;
- розкрити характерні особливості інтерактивної музики у відеоіграх, описати її архітектурні принципи та відмінності від лінійного музичного супроводу;
- вивчити архітектурні особливості ігрового рушія Unity, його складові та вбудовану аудіосистему;
- розробити ігровий прототип у програмі Unity, створити рухомого персонажа та налаштувати базову механіку руху для реалізації звукових рішень;
- описати процес роботи зі звуковим рушієм FMOD Studio, продемонструвати створення звукових подій та їх інтеграцію у Unity;
- розробити та імплементувати авторський звуковий дизайн для ігрового прототипу, включаючи звуки кроків, вітру, атмосфери та інтерактивну музику.

Методологія та методи дослідження:

- теоретичні: опрацювання джерельної бази, літератури та наукових праць з проблем дослідження, їх систематизація та узагальнення;
- історичні: аналіз історичної ретроспективи розвитку відео- та комп'ютерних ігор, еволюції ігрового звуку;
- аналітичні: вивчення функцій звукового дизайну, особливостей інтерактивної музики та її впливу на геймплей;

- емпіричні: практична діяльність аудіо-дизайнера відеоігор, створення звукових ефектів, написання інтерактивної музики, імплементація аудіо у ігровий прототип;
- технологічні: робота з програмним забезпеченням Unity та FMOD Studio, базове програмування для реалізації звукових рішень.

Теоретична база дослідження:

Праці зарубіжних дослідників комп'ютерних ігор та ігрового звуку:

- Kent, S. L. «The Ultimate History of Video Games: From Pong to Pokémon and Beyond»;
- Donovan, T. «Replay: The History of Video Games»;
- Aaron Marks «Complete Guide to Game Audio: For Composers, Sound Designers, Musicians, and Game Developers»;
- Karen Collins «Game Sound: An Introduction to the History, Theory, and Practice of Video Game Music and Sound Design»;
- Ric Viers «Sound Effects Bible: How to Create and Record Hollywood Style Sound Effects»;
- Bridgett, R. «Dynamic Range: Subtlety and Silence in Video Game Sound»;
- Jørgensen, K. «What are Those Grunts and Growls Over There? Computer Game Audio and Player Action»;
- Grimshaw, M. «Sound and Immersion in the First-Person Shooter»;
- Farnell, A. «Designing Sound»;
- Dolby Laboratories «Dolby Atmos for Gaming: Technical Overview».:
- З. Лісса «Естетика кіно музики».

Дослідження вітчизняних авторів:

- К. Станіславська «Мистецько-видовищні форми сучасної культури: монографія»;
- Н. Бєлявіна «Методологія та методика викладання фахових мистецьких дисциплін»;
- Л. Рязанцев «Звукорежисура: навчальний посібник»;
- А. Ананьев «Акустика для звукорежиссеров»;
- О. Бут «Звук як компонент образної структури фільму»;
- В. Дьяченко «Творча діяльність українських звукорежисерів другої половини XX – початку XXI століття: теорія, історія, практика»;
- В. Грищенко «Композиція та комп'ютерне аранжування».

Роботи, опубліковані в інтернет-ресурсах:

- Офіційна документація Unity Technologies та FMOD;
- Сайт «Гільдії розробників інтерактивного аудіо» (Interactive Audio Special Interest Group);

Наукова новизна дослідження:

Вперше:

- комплексно досліджено специфіку роботи звукорежисера у процесі створення відеоігор в українському науковому контексті;
- систематизовано історію розвитку ігрового звукового дизайну з акцентом на техніко-художні трансформації;
- проаналізовано роль базових навичок програмування як дотичної функції сучасного ігрового звукорежисера.

Набуло подальшого розвитку:

- дослідження функцій звукового дизайну та інтерактивної музики у контексті відеоігор;
- вивчення практичного процесу створення аудіо-дизайну та його імплементації у ігровий прототип з використанням професійного програмного забезпечення;
- аналіз особливостей інтерактивної музики як гнучкої системи фрагментів, шарів і переходів.

Джерельна та матеріальна база дослідження: праці з феномену комп'ютерних ігор; матеріали професійних курсів XSSR Academy; офіційна документація Unity та FMOD Studio; ігрові прототипи та звуковий дизайн сучасних комп'ютерних ігор, у тому числі українських; авторський ігровий прототип з реалізованим звуковим оформленням; а також практична діяльність у Студії звукозапису «StartREC Sound Production».

Теоретичне і практичне значення дослідження полягає у комплексному висвітленні специфіки роботи звукорежисера у сфері відеоігор, включаючи міждисциплінарні аспекти професії. Робота містить теоретичні й практичні рекомендації щодо створення звукового та музичного оформлення, роботи з ігровими рушіями та імплементації аудіо у геймплей. Матеріали дослідження можуть бути використані у навчальних курсах дисциплін зі звукорежисури: «Мистецтво звукорежисури», «Звукорежисура», «Композиція та комп'ютерне аранжування», «Студійна звукорежисура», «Аналіз аудіовізуальних творів», а також у спеціалізованих курсах з ігрового аудіо-дизайну.

Апробації та публікації:

Участь у конференції Культура і мистецтво: сучасний науковий вимір : матеріали VIII Всеукр. наук. конф. молод. вч., асп. та магістран. / М-во культ. та страт. ком. України ; Нац. акад. кер. кадрів культ. і мистец. ; Наук. тов. студ., асп., доктор. і молод. вч. (Київ, 06 листопада 2025 р.). Київ : НАКККіМ, 2025.

Публікація тез конференції Цимбалюк В.В. Динамічна обробка звуку: компресія. Культура і мистецтво: сучасний науковий вимір : матеріали VIII Всеукр. наук. конф. молод. вч., асп. та магістран. / М-во культ. та страт. ком. України ; Нац. акад. кер. кадрів культ. і мистец. ; Наук. тов. студ., асп., доктор. і молод. вч. (Київ, 06 листопада 2025 р.). Київ : НАКККіМ, 2025. С. 341–343

Структура кваліфікаційної роботи обумовлена логікою розкриття теми, метою та завданням дослідження. Вона складається зі вступу, трьох розділів, 8 підрозділів, висновків, списку використаних джерел (48 позицій), додатків. Загальний обсяг роботи – 106 сторінок, із них основний текст складає – 94 сторінки.

РОЗДІЛ 1. ІСТОРИКО – ТЕОРЕТИЧНІ ЗАСАДИ ДОСЛІДЖЕННЯ

1.1 Огляд наукових досліджень та літератури

Історія відеоігор бере свій початок не з комерційної індустрії розваг, а з лабораторій, де науковці й дослідники шукали способи використання обчислювальної техніки для моделювання різних процесів, у тому числі ігрових. У другій половині XX століття, на тлі стрімкого розвитку комп'ютерних технологій, стали з'являтися перші проєкти, які сьогодні можна класифікувати як попередники відеоігор. Ці проєкти створювалися не з метою розваги, а як демонстрація технічного потенціалу новітніх пристроїв або як експериментальні системи для тестування програмування, штучного інтелекту чи взаємодії між людиною і машиною.

Одним із ранніх прикладів такої діяльності є робота британського математика та одного з батьків сучасної комп'ютерної науки Алана Тюрінга. Ще у 1950-х роках він розробив алгоритм для гри в шахи, який можна вважати однією з перших комп'ютерних ігор. Попри те, що тоді ще не існувало відповідного обладнання для виконання програми в реальному часі, ідея створення віртуального супротивника, здатного аналізувати ходи та приймати рішення, стала передвісником як майбутніх шахових симуляторів, так і ширше — систем штучного інтелекту у відеоіграх.

Значним кроком у напрямку створення інтерактивної візуальної гри стала розробка **Tennis for Two** (1958) — експериментального проєкту, створеного американським фізиком Вільямом Гігінботамом. Метою створення гри було показати відвідувачам Брукгейвенської національної лабораторії, як комп'ютерні технології можуть застосовуватися не лише в науці, але й для розваг. Гра демонструвала двовимірне зображення тенісного корту на екрані осцилографа [Додаток А] і дозволяла двом гравцям, за допомогою контролерів, відбивати м'яч. Попри свою простоту, Tennis for Two продемонструвала, що комп'ютер може взаємодіяти з користувачем у режимі реального часу, створюючи захопливий ігровий досвід.

Ще однією віховою подією стала поява гри **Spacewar!** у 1962 році, яку розробили Стів Рассел, Мартін Ґрецц і кілька інших студентів Массачусетського технологічного інституту (MIT) на мінікомп'ютері PDP-1. Spacewar! стала першим зразком багатокористувацької гри, в якій два гравці керували космічними кораблями, що маневрували в умовах гравітації зірки та намагалися знищити одне одного. Гра використовувала графічний дисплей — інноваційне на той час рішення, що виводило зображення безпосередньо на монітор, а не лише на паперовий принтер, як це було поширено раніше.

Spacewar! стала не просто технічним досягненням — вона мала вагомий вплив на майбутніх розробників відеоігор. У багатьох випадках вона надихала цілі покоління інженерів і дизайнерів на створення власних продуктів. Зокрема, один із творців першої комерційно успішної аркадної гри Computer Space (1971), Нолан Бушнелл, прямо зазначав, що Spacewar! стала для нього прототипом. Саме з цього моменту ігри починають рух у напрямку комерціалізації та трансформації в масову індустрію.

З технічної точки зору, Spacewar! запровадила низку принципів, які стали канонічними для подальших відеоігор: фізичну симуляцію (гравітація, інерція), конкуренцію між гравцями, а також інтерактивність у реальному часі. Крім того, вона була однією з перших ігор, яка поширювалася через мережу університетських комп'ютерів, що дало змогу їй набути широкого розголосу в академічному середовищі.

Таким чином, початковий етап розвитку відеоігор був тісно пов'язаний із науково-дослідницькими лабораторіями, університетами та технічними інститутами. Це був період експериментів, який поступово трансформувався у більш структурований підхід до створення ігор як окремого виду цифрової творчості. Саме ці перші кроки заклали фундамент для розробки ігрових механік, інтерактивних інтерфейсів, а також візуального та звукового оформлення, які у майбутньому визначили еволюцію ігрової індустрії.

Дослідження відеоігор як міждисциплінарного явища, що поєднує у собі елементи культури, технологій, мистецтва та соціальної взаємодії, є відносно новим напрямом у гуманітарних та технічних науках. Однак з кожним роком обсяг наукових джерел, присвячених цьому феномену, стрімко зростає. Відеоігри вже давно перестали бути виключно розважальним продуктом; вони перетворилися на серйозний об'єкт академічного аналізу, зокрема у галузях медіазнавства, культурології, соціології, психології, а також прикладних технологічних наук. Відеоігри розглядаються як частина сучасних видовищних форм культури, що підтверджується дослідженнями К. Станіславської [9].

Серед найвпливовіших праць, що досліджують структуру, функції та вплив відеоігор, , можна виокремити книгу Джессі Шелла *The Art of Game Design: A Book of Lenses* [35]. Автор, який є не лише теоретиком, а й практиком у сфері геймдизайну, розглядає відеогру як складну інтерактивну систему, яка одночасно є художнім витвором, програмним продуктом і психологічним експериментом. У своїй концепції "лінз" (lenses) він пропонує багатовимірний підхід до створення ігор, що дозволяє враховувати не лише геймплей, але й наратив, естетику, зручність користування, звукове оформлення та інші аспекти, які формують цілісний досвід гравця.

У ширшому культурному контексті слід згадати працю Йогана Хейзінги *Homo Ludens* [4], яка хоч і з'явилася задовго до ери цифрових технологій, заклала концептуальні основи сучасного розуміння гри як фундаментального чинника культури. Хейзінга розглядав гру не просто як дозвілля, а як первинну, архетипову форму культурної діяльності, яка передує та формує інші соціальні інститути — мову, право, релігію, мистецтво. Його концепція "людини-гравця" відкрила шлях до розуміння ігрових практик не лише як способу розваги, а як глибинної культурної моделі, що структурує людське буття.

Ця ідея отримала подальший розвиток у працях таких дослідників, як Ерік Циммерман і Кеті Сален (Rules of Play), Гонсало Фраска, Ієн Богост,

Еспен Аарсет та інших, які заклали основи теорії ігор як нової форми культурної репрезентації. Вони наголошують, що відеоігри не лише відображають реальність, а й активно її моделюють, створюючи нові способи сприйняття часу, простору, причинно-наслідкових зв’язків, а також емоційного та сенсорного досвіду.

Водночас сучасні дослідження все частіше звертають увагу на важливу роль аудіовізуального оформлення у формуванні емоційного фону гри. Особливу увагу при цьому приділяють звуку — одному з найменш досліджених, але надзвичайно впливових елементів ігрового процесу. Роботи таких авторів, як Карен Коллінз (Game Sound: An Introduction to the History, Theory, and Practice of Video Game Music and Sound Design), Міша Качан (The Sound of Games) та Брендан Кіллелі (Sound Design for Games), висвітлюють технічні та естетичні аспекти створення ігрового звуку. Зокрема, вони розглядають динамічне мікшування, процедурну генерацію звуків, інтерактивне озвучення об’єктів і середовищ, а також використання музичних тем для емоційного маніпулювання гравцем.

Окрему увагу слід приділити дослідженням, що аналізують вплив звуку на когнітивні процеси гравця. Наприклад, у роботах психологів і нейробіологів доведено, що звукові ефекти можуть покращувати концентрацію уваги, сприяти глибшому зануренню в ігровий світ та навіть викликати фізіологічну реакцію — пришвидшене серцебиття, потовиділення, зміни у диханні. Це дає підстави говорити про звук у відеоіграх не лише як про фон, а як про повноцінний механізм впливу, рівнозначний візуальній складовій.

В українському академічному дискурсі тема відеоігор поступово набуває ваги. Серед значущих вітчизняних праць варто виокремити дослідження С. Іжакевич [24], яка комплексно розглядає "світ комп'ютерних ігор" не лише як технологічний продукт, а як специфічне соціокультурне середовище.. Окремі автори звертаються до проблеми ідентичності, наративної структури, соціального впливу та креативного потенціалу гейм-

індустрії. Проте технічна складова, зокрема питання аудіодизайну, звукорежисури, інтерактивного звукового середовища, поки що досліджена вкрай недостатньо. Розвиток звукорежисури як окремого виду творчої діяльності та її трансформації в українському контексті детально проаналізовано у працях В. Дьяченка [6], який досліджує теорію та практику українських звукорежисерів. Технічні та естетичні аспекти створення ігрового звуку висвітлюють такі автори, як Карен Коллінз [14] та Аарон Маркс [30].

Останніми роками зростає інтерес до цієї теми серед фахівців у галузі комп'ютерних наук, інженерії програмного забезпечення та звукового дизайну. Це можна простежити на прикладі українських дослідницьких проєктів, студентських дипломних робіт, публікацій у спеціалізованих виданнях та зростання кількості курсів з інтерактивного аудіо в рамках освітніх програм з геймдизайну.

Таким чином, літературний огляд демонструє, що дослідження відеоігор, зокрема у контексті звукового супроводу, є багатошаровим, динамічним і перспективним напрямом, який потребує подальшого міждисциплінарного осмислення. Вивчення ролі звукорежисера у цьому процесі є важливим кроком до глибшого розуміння внутрішньої структури сучасної відеогри як складного інтерактивного твору.

1.2 Історичний розвиток відеоігор та еволюція ігрового звуку

Комерціалізація відеоігор та формування індустрії (1970-ті роки)

Після перших експериментів у наукових лабораторіях відеоігри поступово почали виходити за межі академічного середовища і перетворюватися на форму масових розваг. Саме 1970-ті роки стали періодом народження індустрії відеоігор у її комерційному розумінні. Цей етап позначений стрімким розвитком аркадних ігор, появою перших домашніх ігрових систем, а також закладенням основ ігрової економіки, яка у наступні десятиліття перетвориться на один з найприбутковіших секторів індустрії розваг.

Виникнення аркадної індустрії та феномен Pong

Однією з найвизначніших подій цього періоду стало створення у 1972 році компанії Atari, заснованої інженером Ноланом Бушнеллом та підприємцем Тедом Дабні. Того ж року Atari випустила свою першу аркадну гру — Pong, яка стала справжньою сенсацією і каталізатором комерційного успіху відеоігор [26]. Гра являла собою просту симуляцію настільного тенісу на двох гравців з мінімалістичною чорно-білою графікою, але саме її доступність, інтуїтивність управління за допомогою обертових контролерів та змагальний елемент сприяли масовій популяризації відеоігор серед широкого загалу. Atari

Pong започаткував справжній «аркадний бум»: до середини 1970-х років автомати з відеоіграми стали з'являтися у торгових центрах, барах, ресторанах, кінотеатрах та спеціалізованих ігрових залах (arcades), а сама ідея цифрової гри увійшла в культуру як нова форма проведення дозвілля. Аркадні зали перетворилися на соціальні простори, де молодь збиралася не лише для гри, але й для спілкування та змагань за найкращі результати [27]. Цей період також ознаменувався появою перших професійних турнірів з відеоігор, що заклало основи для майбутньої індустрії кіберспорту. За оцінками дослідників, до кінця 1970-х років аркадна індустрія генерувала мільярдні прибутки, перевершуючи касові збори Голлівуду [22].

Розвиток домашніх ігрових консолей

У відповідь на зростання інтересу до відеоігор як масового явища та прагнення споживачів грати вдома почали з'являтися перші домашні ігрові консолі. Хоча перші спроби створення домашніх систем (зокрема, Magnavox Odyssey у 1972 році Ральфа Баєра) мали обмежений успіх через високу вартість, технічні обмеження та складність у використанні, саме друга половина 1970-х років стала переломним моментом у цьому напрямку [10].

У 1977 році Atari випустила Atari Video Computer System (VCS), пізніше відому як Atari 2600 пізніше відому як Atari 2600 [Додаток Б], яка

революціонізувала ринок домашніх розваг [31]. Ця система стала знаковою для індустрії з кількох причин. По-перше, вона дозволяла грати у різні ігри завдяки інноваційній системі змінних ROM-картриджів, що кардинально відрізняло її від попередніх консолей, які мали вбудовані ігри без можливості розширення бібліотеки. Концепція змінних картриджів дозволила створити новий ринок програмного забезпечення та відкрила можливості для сторонніх розробників.

Система дозволяла гравцям відчути елемент контролю та занурення в ігровий світ через джойстики та інші контролери, не виходячи з дому. Atari 2600 також першою продемонструвала успішну бізнес-модель, де прибуток отримувався не стільки від продажу самої консолі (яка часто продавалася зі збитком або мінімальною маржею), скільки від ліцензування та продажу ігрових картриджів [22]. Ця модель, відома як "razor and blades" (бритва та лезо), стала стандартом для індустрії консолей на десятиліття вперед.

Таким чином, відбувся важливий перехід від публічного (аркадного) до персонального (домашнього) формату гри, що мав глибокі соціальні та культурні наслідки. Відеоігри почали проникати в сімейний побут, змінюючи патерни дозвілля та споживання медіа-контенту. До початку 1980-х років Atari 2600 було продано понад 30 мільйонів примірників у всьому світі, що зробило її однією з найуспішніших консолей свого покоління [44]. Бібліотека ігор для системи налічувала понад 500 тайтлів, включаючи такі культові ігри, як Space Invaders, Pac-Man та Pitfall! [29].

Формування ігрової економіки

1970-ті роки також ознаменувалися формуванням основних економічних моделей індустрії відеоігор. Поява аркадних автоматів створила новий сегмент розважальної індустрії з постійним потоком доходів через монетизацію за принципом "монета за гру". За деякими оцінками, один успішний аркадний автомат міг генерувати від 200 до 400 доларів на тиждень, що робило бізнес надзвичайно прибутковим [26]. Паралельно розвивалася

модель продажу домашніх консолей та картриджів, яка передбачала одноразову покупку апаратного забезпечення з подальшим придбанням програмного забезпечення.

У цей період також з'явилися перші незалежні розробники ігор. У 1979 році група програмістів Atari — Девід Крейн, Ларрі Каплан, Алан Міллер та Боб Вайтхед — незадоволених політикою компанії щодо авторських прав, відсутності визнання розробників та низької винагороди, заснувала компанію Activision — першого стороннього видавця ігор для консолей. Ця подія стала важливим прецедентом, що заклав основи для майбутньої диверсифікації ринку та появи незалежної ігрової індустрії [37].

Золота ера аркад та відеоігрова криза (1980-ті роки

Розквіт аркадної індустрії

Початок 1980-х років ознаменувався золотою ерою аркадних ігор, коли з'явилися культові тайтли, що визначили майбутнє індустрії. У 1980 році компанія Namco (у США видана Midway) випустила Pac-Man — гру, яка стала культурним феноменом і вийшла далеко за межі ігрових залів [26]. Pac-Man відрізнявся від більшості тогочасних аркадних ігор, які зосереджувалися на космічній тематиці та стрілянині. Замість цього гравцям пропонувалося керувати жовтим персонажем, що поїдав точки в лабіринті, уникаючи привидів [48].

Успіх Pac-Man був безпрецедентним: гра згенерувала понад 1 мільярд доларів прибутку тільки в США протягом перших років після випуску, стала об'єктом масового мерчандайзингу (від іграшок до одягу) та навіть отримала власний хіт-сингл "Pac-Man Fever" [28]. Персонаж Pac-Man став одним з найвпізнаваніших символів поп-культури 1980-х років та залишається іконою відеоігор донині.

У 1981 році Nintendo випускає Donkey Kong, ще одну революційну гру, яка знайомить гравців із персонажем на ім'я Маріо (спочатку відомим як Jumpman). Donkey Kong стала важливою віхою в розвитку платформерів та наративного дизайну в іграх, впроваджуючи елементи історії та характеру

персонажів [26]. Гра була створена молодим дизайнером Сігеру Міямото і стала його першим великим успіхом.

Це стало початком однієї з найвідоміших ігрових франшиз у світі. У 1985 році Nintendo запускає консоль Nintendo Entertainment System (NES) разом із грою Super Mario Bros., яка кардинально змінила уявлення про геймплей, структуру рівнів та музичне супроводження. Super Mario Bros. не тільки вразила рівнем технічної реалізації, а й продемонструвала нову форму наративного ігрового досвіду — переходу від простих етапів до повноцінної історії з персонажами, середовищем та чітко визначеними цілями [38]. Гра встановила стандарти дизайну рівнів, які використовуються донині, включаючи поступове введення механік, криву складності та приховані секрети для дослідження [35-].

Відеоігрова криза 1983 року

Важливою подією у цей період був відеоігровий кризис 1983 року (North American video game crash), який виник через перенасичення ринку низькоякісними продуктами, зниження довіри споживачів та відсутність регуляції. Компанії, не маючи єдиних стандартів якості, випускали численні копії успішних ігор або технічно слабкі продукти, що призвело до значних фінансових втрат у галузі. Особливо символічною стала ситуація з грою E.T. the Extra-Terrestrial для Atari 2600, яка була розроблена всього за п'ять тижнів і виявилася комерційним провалом, попри величезні інвестиції в ліцензію [7].

За оцінками, північноамериканський ринок відеоігор скоротився з 3,2 мільярда доларів у 1983 році до приблизно 100 мільйонів доларів у 1985 році — падіння на 97% [48]. Багато компаній збанкрутували, а роздрібні мережі почали відмовлятися від продажу відеоігор.

Проте саме Nintendo, із запуском NES у 1985 році, вдалося відродити індустрію, встановивши жорсткий контроль за якістю ігор та авторизацією сторонніх розробників через систему ліцензування, чим започаткувала нову модель стосунків між платформоутримувачами та контентмейкерами [38].

Nintendo впровадила "Seal of Quality" — знак якості, який гарантував, що гра відповідає технічним та змістовним стандартам компанії.

Розвиток поліфонії та семплування в ігровому аудіо (середина 1980-х – початок 1990-х)

Технологічні прориви в ігровому звуку

Середина 1980-х років ознаменувалася значним стрибком у розвитку ігрового аудіо з переходом до цифрового звуку та семплування. Це стало можливим завдяки появі більш потужного обладнання та інноваційних підходів до синтезу та відтворення звуку. Якщо раніше ігрова музика обмежувалася простими тонами та мелодіями, то нова ера принесла багатошарові композиції, реалістичні звукові ефекти та складні музичні аранжування [14-].

Комп'ютер Amiga (Commodore, 1985) став переломним моментом для цифрового звуку в іграх. Система була оснащена передовим чотириканальним звуковим чипом Paula, що дозволяв відтворювати складні оцифровані семпли з частотою дискретизації до 28 кГц, досягаючи безпрецедентного для домашніх систем реалізму. Це також сприяло популяризації формату MOD (module file format) — революційного формату для зберігання музики, який містив як семпли інструментів, так і інформацію про їх відтворення, створюючи компактні, але якісні музичні композиції [12].

Консоль Famicom/NES (Nintendo, 1983/1986) мала п'ять звукових каналів: два прямокутних хвильових (pulse wave) канали для мелодій, один трикутний хвильовий канал для басів, один канал білого шуму для ударних та один канал Delta Modulation для простих семплів PCM [14-]. Хоча технічно система була обмеженою, талановиті композитори, такі як Кодзі Кондо, створювали незабутні мелодії, які залишаються культовими донині.

Sega Genesis/Mega Drive (1988) використовувала гібридний підхід з потужним чіпом Yamaha YM2612, який надавав 6 FM-звукових каналів (можливість використати один канал для PCM-семплів), плюс окремий чіп

Texas Instruments SN76489 PSG з 4 додатковими каналами, загалом даючи 10 каналів звуку. Така архітектура дозволяла створювати дуже багатий звуковий ландшафт, хоча й вимагала високої кваліфікації від композиторів для ефективного використання FM-синтезу.

На платформі IBM PC з'явилися перші спеціалізовані звукові карти, які кардинально змінили можливості комп'ютерних ігор. Sound Blaster (1989) від Creative Labs підтримувала як FM-синтез, так і PCM-семпли з частотою дискретизації до 22.05 кГц (пізніше моделі підтримували до 44.1 кГц — CD-якість), а також мала вбудований MIDI-інтерфейс [14-]. Sound Blaster швидко стала індустріальним стандартом для PC-ігор завдяки своїй універсальності та доступній ціні.

Super Famicom/SNES (Nintendo, 1990 в Японії, 1991 в Північній Америці) здійснила революцію в ігровому аудіо завдяки повноцінному 16-бітному звуку, стерео-панорамуванню та ефектам реверберації. Її незалежна звукова система включала 8-бітний процесор Sony SPC700 та 16-бітний цифровий сигнальний процесор (DSP), що працювали паралельно з основним CPU, дозволяючи створювати насичені багатошарові звукові ландшафти без навантаження на головний процесор [36].

Поява професійних композиторів

Цей період також ознаменувався появою професійних композиторів та саунд-дизайнерів, які почали активно працювати над відеоіграми як над повноцінною формою музичного мистецтва. З середини 1980-х років такі композитори, як Коїчі Сугіяма (Dragon Quest, 1986), Кодзі Кондо (Super Mario Bros., 1985; The Legend of Zelda, 1986) та Нобуо Уемацу (Final Fantasy, 1987), почали створювати музику для ігор, яка виходила за рамки простого звукового супроводу та ставала невід'ємною частиною ігрового досвіду [40-].

Коїчі Сугіяма, професійний композитор класичної музики, підняв планку ігрової музики, створюючи повноцінні оркестрові партитури для серії Dragon Quest. Його роботи стали першими ігровими саундтреками, які

виконувалися симфонічними оркестрами на концертах. Кодзі Кондо створив одні з найвпізнаваніших мелодій в історії відеоігор, включаючи тему Super Mario Bros., яка стала культурною іконою. Нобуо Уемацу привніс емоційну глибину та епічність у серію Final Fantasy [40-].

Епоха CD-ROM та потокового аудіо (середина 1990-х – початок 2000-х)

Революція оптичних носіїв

Середина 1990-х років стала переломною для ігрового аудіо завдяки масовому впровадженню технології CD-ROM (Compact Disc Read-Only Memory). Це дозволило значно розширити можливості звукового оформлення ігор, подолавши жорсткі обмеження пам'яті, характерні для картриджної ери. Якщо стандартний картридж для SNES міг вмістити лише 64-128 КБ звукових даних, то CD-ROM надавав 650-700 МБ загального простору, що дозволяло розробникам включати годинники високоякісної музики та озвучення [14-].

Sony PlayStation (1994) стала найуспішнішою консоллю цього покоління та встановила нові стандарти для ігрового аудіо. Система підтримувала 24 канали 16-бітних ADPCM-семплів з частотою дискретизації до 44.1 кГц, що відповідало якості CD-аудіо, а також мала апаратні ефекти DSP, такі як реверберація, що дозволяло створювати просторовий та атмосферний звук [11].

Завдяки цим можливостям на PlayStation з'явилися ігри з кінематографічним звуковим оформленням, такі як Final Fantasy VII (1997), Metal Gear Solid (1998) та Resident Evil (1996), які використовували оркестрові записи та професійне озвучення для створення глибокого емоційного занурення [40-].

Розвиток голосової акторської майстерності

Ця епоха також стала періодом значного розвитку голосової акторської майстерності у відеоіграх, що кардинально змінило спосіб розповідання історій в інтерактивних медіа. Якщо раніше ігри покладалися виключно на

текст для передачі діалогів, то тепер з'явилася можливість використовувати живе мовлення для створення емоційно насичених персонажів [25].

Metal Gear Solid (1998) підняла планку голосового акторства в іграх, використовуючи кінематографічний підхід до запису діалогів. Режисер Хідео Кодзіма наполягав на використанні професійних голливудських акторів та застосував технології motion capture та facial capture для синхронізації губ з діалогами. Гра містила понад 2 години кат-сцен з повним озвученням, що було безпрецедентним для того часу.

Final Fantasy X (2001, PlayStation 2) відзначилася тим, що стала першою грою в серії Final Fantasy з повністю озвученими діалогами та мала великий склад професійно озвучених персонажів, що стало значним кроком для японських RPG [28].

Поява аудіо-мідлверу

На початку 2000-х років відбулася ще одна революція в ігровому аудіо завдяки появі мідлверу (Middleware) — спеціалізованого програмного забезпечення, що виступає посередником між контентом (музикою та звуковими ефектами) та ігровим рушієм. Такі рішення, як FMOD (Firelight Technologies, 2002) та Wwise (Audiokinetic, 2006), змінили підхід до впровадження аудіо в ігри [14-].

Мідлвер надає розробникам потужні інструменти для створення складних аудіосистем, включаючи контроль параметрів у реальному часі, просторове аудіо, адаптивні звукові ландшафти та інтерактивні музичні системи. Ці інструменти значно розширюють можливості звукових професіоналів, дозволяючи їм самостійно впроваджувати аудіо без глибоких знань програмування, зменшуючи залежність від програмістів [18].

Сучасні тенденції та технології ігрового аудіо (2000-ті – сьогодення)

Імерсивне та тривимірне аудіо

Сучасна ера ігрового аудіо характеризується безпрецедентним рівнем занурення та інтерактивності, що стало можливим завдяки значним технологічним досягненням. Імерсивне та тривимірне аудіо займають провідну роль у створенні звукового ландшафту сучасних відеоігор. Surround sound у форматах 5.1 (п'ять основних каналів плюс один сабвуфер) та 7.1 (сім основних каналів плюс сабвуфер) став стандартом для домашніх консолей та ПК-ігор, створюючи 360-градусне звукове середовище, яке посилює відчуття присутності гравця в ігровому світі [11]. Технології на кшталт Dolby Digital 5.1 були інтегровані в консолі Xbox 360 (2005) та PlayStation 3 (2006), дозволяючи розробникам створювати складні звукові ландшафти, де звуки точно локалізовані в тривимірному просторі.

Особливу увагу привертає бінуральне аудіо — технологія, яка записується за допомогою двох мікрофонів, розташованих на манекені, що імітує людську голову та вуха, забезпечуючи точну локалізацію джерела звуку та реалістичне сприйняття відстані навіть при прослуховуванні через звичайні навушники. Яскравим прикладом є Senua's Saga: Hellblade II (Ninja Theory, 2024), де внутрішні голоси героїні, що страждає від психозу, створюють унікальний та тривожний ефект, начебто звучачи безпосередньо в голові гравця.

Сучасні консолі, такі як PlayStation 5 (2020) та Xbox Series X/S (2020), підтримують Dolby Atmos — об'єктно-орієнтовану аудіотехнологію, що забезпечує кінематографічну глибину звучання, дозволяючи розробникам розміщувати звукові об'єкти не лише навколо слухача, але й над ним, створюючи справжнє тривимірне звукове поле [15].

Просторове аудіо стало важливим елементом тактичної переваги у жанрі FPS (First-Person Shooter), дозволяючи гравцям орієнтуватися за напрямком пострілів, кроків чи наближенням ворогів. В іграх, таких як Counter-Strike: Global Offensive (2012), Rainbow Six Siege (2015) та Call of Duty: Modern Warfare (2019), точна локалізація звуку може бути вирішальною для успіху в змаганні.

Адаптивна та динамічна музика

Не менш важливим напрямом є адаптивна та динамічна музика — музична система, яка реагує на дії гравця і змінюється залежно від його місцезнаходження, стану персонажа та ігрових подій, створюючи атмосферу відповідно до настрою та масштабу оточення [39].

Подібні рішення реалізовані, наприклад, у Red Dead Redemption 2 (Rockstar Games, 2018), де музика динамічно адаптується до дій гравця: під час спокійної їзди верхи звучать ліричні мелодії, а під час перестрілок музика різко змінюється на напружену екшн-тему. Гра використовує систему вертикального ремікшування, де окремі музичні шари (strings, brass, percussion) додаються або прибираються залежно від інтенсивності дії.

У жанрі хорору музика виконує прогностичну функцію, нарощуючи напругу ще до появи загрози. В Resident Evil, Silent Hill та Dead Space музика та звукові ефекти є ключовими інструментами для створення атмосфери жаху, використовуючи диссонанси, несподівані звуки та з ловісні тембри для маніпулювання емоціями гравця [40].

Окремо варто відзначити ритм-ігри, де музика може ставати елементом покарання або винагороди: неправильні дії гравця порушують музичну гармонію, тоді як точне виконання створює задовільний музичний досвід. Crypt of the NecroDancer (Brace Yourself Games, 2015) — унікальна гра, що поєднує roguelike-механіки з ритм-геймплеєм, де всі дії, включаючи рух та атаки, повинні виконуватися в такт музиці [22]. Інші приклади включають Guitar Hero, Rock Band та Beat Saber, де музика є центральним елементом геймплею.

Процедурна генерація аудіо

Іншим важливим трендом стала процедурна генерація аудіо — використання алгоритмів та штучного інтелекту для автоматичного створення звукових ефектів та музики в реальному часі. Ця технологія дає можливість

формувати унікальний звуковий досвід для кожного проходження гри, що особливо актуально для ігор з процедурно генерованими світами [17].

Відомими прикладами є No Man's Sky (Hello Games, 2016) та Spelunky (Mossmouth, 2008/2012), де процедурна генерація забезпечила майже безмежне варіювання музичних пейзажів. У No Man's Sky музика генерується на основі характеристик планети (клімат, флора, фауна), створюючи унікальний саундтрек для кожного з 18 квінтильйонів процедурно згенерованих світів. Група 65daysofstatic створила алгоритмічну систему, що комбінує попередньо записані музичні елементи в нові композиції залежно від контексту гри.

Разом з тим виникають проблеми, пов'язані з якістю звуку, що зазвичай поступається ретельно створеним та вручну обробленим семплам, а також обмеження у дизайні, що потребують подальших досліджень та вдосконалення алгоритмів [14-]. Процедурно згенерована музика може бути менш емоційно насиченою та не завжди здатна створити запам'ятовувані музичні теми, які є характерними для класичних ігрових саундтреків.

Однак технологія продовжує розвиватися. Штучний інтелект та машинне навчання починають відігравати все більшу роль у створенні адаптивного аудіо. Компанії, такі як Aiva Technologies та Amper Music, розробляють AI-системи, здатні композувати музику на основі заданих параметрів настрою, жанру та інтенсивності. Хоча ці системи ще не замінили людських композиторів, вони можуть бути корисними інструментами для створення динамічного фонового аудіо або прототипування музичних ідей.

Розвиток інструментарію та робочих процесів

Розвиток інструментарію та робочих процесів у сфері звукового дизайну став одним із ключових чинників еволюції індустрії відеоігор. З удосконаленням апаратного забезпечення та розширенням програмних можливостей з'явилися інструменти, що дозволили об'єднати творчі, технічні та інтеграційні процеси в єдину систему. Це сприяло переходу від простих

лінійних методів роботи до комплексного інтерактивного підходу, який відповідає сучасним вимогам геймдеву.

Одним із найважливіших кроків у цьому напрямі стало впровадження цифрових аудіоробочих станцій (DAW — Digital Audio Workstations), які стали універсальним стандартом у сфері звукового дизайну та музичного продакшену. Такі програми, як Pro Tools, Reaper, Logic Pro, Cubase та Ableton Live, забезпечують повний цикл роботи зі звуком — від запису й редагування до мікшування та фінального мастерингу. Їхня інтеграція з сучасними плагінами, системами автоматизації та MIDI-контролерами зробила можливим створення звукових ефектів найвищої якості, що відповідають кінематографічним стандартам.

Паралельно із цим активно розвивався аудіо-мідлвер — спеціалізоване програмне забезпечення, яке забезпечує інтеграцію звукових даних безпосередньо в ігрові рушії. Такі системи, як Wwise (Audiokinetic), FMOD Studio (Firelight Technologies) та Fabric (Tazman Audio), стали незамінним інструментом для професійних саунд-дизайнерів [22]. Вони надають розширені можливості, серед яких:

- Контроль параметрів у реальному часі — можливість пов'язувати звукові характеристики (гучність, тембр, фільтрацію тощо) з ігровими змінними, як-от швидкість руху персонажа або рівень здоров'я.
- Просторове аудіо — підтримка сучасних форматів об'ємного звучання (surround sound, Dolby Atmos), а також технологій бінурального та об'єктно-орієнтованого аудіо.
- Адаптивні звукові ландшафти — створення динамічних аудіосередовищ, що змінюються залежно від дій гравця або стану ігрового світу.
- Інтерактивні музичні системи — інструменти для реалізації вертикального та горизонтального ремікшування музики, що дозволяє формувати гнучкі музичні структури, адаптивні до ігрових ситуацій.

Використання подібних систем змінило підхід до організації робочого процесу: звукорежисери та композитори тепер мають змогу самостійно інтегрувати аудіо у проєкти, не залучаючи безпосередньо програмістів. Це значно зменшило часові витрати та підвищило ефективність ітераційного

процесу розробки. Крім того, сучасні мідлвер-платформи підтримують візуальне програмування (node-based scripting), що робить створення складних логічних аудіосистем доступним навіть для спеціалістів без глибоких знань у програмуванні [22].

Окремим напрямом розвитку є впровадження штучного інтелекту (AI) у процеси звукового дизайну. Алгоритми машинного навчання поступово беруть на себе рутинні або технічно складні завдання — автоматичне очищення записів від шумів, оптимізацію рівнів гучності, генерацію варіацій ефектів або створення динамічних звукових композицій у реальному часі. Сучасні AI-інструменти здатні також аналізувати контекст гри та пропонувати звукові рішення, що відповідають емоційному чи сюжетному стану сцени.

Історичний розвиток відеоігор та еволюція ігрового звуку демонструють тісний взаємозв'язок між технологічним прогресом та художніми інноваціями. Від простих електронних сигналів перших аркадних автоматів до складних адаптивних аудіосистем сучасних ігор — звук пройшов шлях від другорядного елемента до повноцінної складової ігрового досвіду, що формує атмосферу, емоції та занурення гравця у віртуальний світ.

Кожен етап розвитку індустрії відеоігор приносив нові виклики та можливості для звукорежисерів і композиторів. 1970-ті роки заклали комерційний фундамент індустрії та показали потенціал звуку як елемента геймплею. 1980-ті роки ознаменувалися появою культових мелодій та формуванням жанрових звукових канонів. Середина 1980-х – початок 1990-х принесли революцію цифрового звуку та семплування, що дозволило створювати багатошарові композиції та реалістичні звукові ефекти.

Епоха CD-ROM у середині 1990-х – початку 2000-х відкрила двері для кінематографічного озвучення, професійної голосової акторської майстерності та високоякісних оркестрових записів. Поява аудіо-мідлверу, такого як FMOD та Wwise, кардинально змінила робочі процеси, надавши звуковим професіоналам потужні інструменти для створення інтерактивних аудіосистем без необхідності глибокого програмування.

Сучасний етап (2000-ті – сьогодення) характеризується безпрецедентним рівнем технологічної складності та художньої виразності. Імерсивне та тривимірне аудіо, адаптивна та динамічна музика, процедурна генерація звуку та впровадження штучного інтелекту відкривають нові горизонти для створення унікальних ігрових досвідів. Звук у сучасних відеоіграх не просто супроводжує дію на екрані — він активно взаємодіє з гравцем, реагує на його рішення, формує емоційний ритм гри та стає невід'ємною частиною наративу.

Роль звукорежисера у створенні відеоігор постійно еволюціонує. Від простого підбору звукових ефектів професія трансформувалася у комплексну діяльність, що вимагає глибоких технічних знань, художнього чуття, розуміння психології сприйняття та вміння працювати в умовах міждисциплінарної командної співпраці. Сучасний звукорежисер повинен володіти не лише традиційними навичками звукозапису та мікшування, але й розуміти принципи інтерактивного дизайну, програмування аудіосистем, роботи з ігровими рушіями та мідлвер-платформами.

Перспективи розвитку ігрового аудіо пов'язані з подальшим вдосконаленням технологій просторового звуку, розширенням можливостей штучного інтелекту для створення адаптивних музичних систем, а також інтеграцією звуку з іншими сенсорними модальностями (тактильний зворотний зв'язок, віртуальна та доповнена реальність). Відеоігри продовжують утверджуватися як повноцінна форма сучасного мистецтва, де звук відіграє одну з провідних ролей у створенні емоційно насичених, інтерактивних та незабутніх досвідів для мільйонів гравців по всьому світу.

1.3. Етапи розвитку комп’ютерних ігор та їх техніко-художні трансформації

Розвиток відеоігор як культурного й технологічного феномену відбувався у тісному взаємозв’язку між художніми пошуками та інженерними досягненнями. Відеогра є синтетичним видом мистецтва, що поєднує

елементи графіки, звуку, музики, драматургії, літератури й інтерактивного дизайну. Від перших аркад до сучасних 3D-світів еволюція ігор визначалась не лише технічними можливостями обчислювальної техніки, але й розвитком художнього мислення розробників.

Технічна еволюція як основа художнього розвитку

Початок становлення відеоігор як нового виду цифрового мистецтва припадає на 1970–1980-ті роки, коли технічні можливості обчислювальних машин і домашніх консолей були надзвичайно обмеженими. У цей період апаратні ресурси дозволяли відображати лише найпростіші геометричні форми, а палітра кольорів складалася з кількох базових відтінків. Графіка мала суто піксельний характер, де кожен елемент — від гравця до об’єкта — зображався у вигляді квадратів або ліній. Незважаючи на такі обмеження, саме в цей час почала формуватись унікальна естетика ранніх ігор, заснована на простоті, контрастності та глибокому відчутті ритму.

Перші проєкти, такі як *Pong* (1972), *Space Invaders* (1978) чи *Asteroids* (1979), заклали основи не лише технічного, а й художнього підходу до створення інтерактивного досвіду. Графічна мінімалістичність цих ігор компенсувалася динамічністю геймплею, швидкою реакцією користувача і повторюваними аудіо-сигналами, які виконували функцію ритмічного фону. Звукове оформлення, хоча й складалося з простих електронних сигналів, створювало необхідне відчуття напруги та змагання. Таким чином, технічна обмеженість не заважала вираженню емоційного напруження — навпаки, спонукала розробників шукати нестандартні рішення, де кожен піксель і кожен звук набував значення.

У подальші роки, з появою восьмибітних та шістнадцятибітних платформ — таких як *Nintendo Entertainment System (NES)*, *Sega Master System* чи *Atari 2600* — ситуація змінилася кардинально. Технічний прогрес у сфері апаратного забезпечення дозволив розробникам реалізовувати більш складні форми візуалізації та звучання. З’явилися перші спроби створення об’ємних

зображень, деталізованих спрайтів, а також можливість використання кількох шарів фону для імітації глибини простору.

Звуковий супровід теж зазнав суттєвого вдосконалення. Завдяки появі чипів, здатних генерувати кілька звукових каналів одночасно, композитори змогли записувати багатоголосні теми, використовуючи формат MIDI. Це дало змогу створювати впізнавані мелодії, які надовго закарбувались у свідомості гравців. Наприклад, музичні теми з ігор *Super Mario Bros.*, *The Legend of Zelda* чи *Sonic the Hedgehog* стали не просто частиною ігрового процесу, а повноцінними культурними символами епохи.

Зростання технічних можливостей сприяло формуванню нових жанрових структур, що вимагали власних художніх рішень. Платформери, рольові ігри (RPG), шутери, стратегії та аркади — кожен із цих жанрів почав розвивати свою окрему візуальну і звукову мову. Наприклад, RPG-ігри на кшталт *Final Fantasy* вирізнялися епічними оркестровими композиціями, що підкреслювали розгорнутий сюжет і драматизм подій, тоді як аркадні шутери покладалися на швидкі повторювані мотиви, які стимулювали реакцію гравця.

Водночас, у міру розвитку техніки, ігри поступово набували кінематографічних рис. З’являється розуміння, що відеогра може виступати не лише розвагою, а й засобом художнього самовираження. Дизайнери почали експериментувати з кольором, композицією кадру, ритмом подій і звуковими ефектами. Візуальна складова перестала бути суто утилітарною — вона перетворилась на важливий інструмент емоційного впливу.

Технічна еволюція поступово зумовила появу перших форм віртуальної режисури. Навіть у межах обмежених ресурсів, розробники прагнули створювати сцени з динамічними переходами, зміною перспективи та ілюзією руху камери. У цьому сенсі відеоігри почали наслідувати естетичні принципи кіномистецтва, сформульовані З. Ліссою [7], де візуальний ряд і звуковий супровід утворюють синтетичну єдність, а музика виступає не просто ілюстрацією, а важливим драматургічним чинником.

Перехід до тривимірного простору

Справжнім переломним етапом у розвитку відеоігор став перехід до тривимірного простору, що відбувся у середині 1990-х років. Якщо раніше ігрові світи існували переважно у двовимірній площині, де дія обмежувалася горизонтальним чи вертикальним рухом, то впровадження технології 3D-графіки кардинально змінило саме уявлення про віртуальний простір. Поява таких рушіїв, як **Doom Engine**, **Quake Engine** та пізніше **Unreal Engine**, задала абсолютно новий стандарт візуальної реалістичності та інтерактивності.

Уперше гравець отримав можливість не просто спостерігати за подіями збоку, а занурюватися у світ гри, взаємодіяти з простором, досліджувати його зсередини. Ігрове середовище перестало бути статичним фоном — воно перетворилось на **повноцінний художній простір**, у якому кожен об'єкт, текстура, світло чи звук почали працювати на створення цілісного враження та емоційного впливу.

Тривимірна графіка відкрила безліч нових художніх можливостей. Розробники почали застосовувати принципи **кінематографічної композиції** — побудову кадру, ігру зі світлом і тінню, динамічні ракурси та монтажні прийоми. Зображення стало інструментом не лише геймплею, а й **візуального оповідання**, що наближало ігри до кіно як до повноцінного виду мистецтва.

Технічні новації у сфері зображення супроводжувалися активним розвитком **звукових технологій**. Саме в цей період з'являються перші системи об'ємного звучання, що дозволили гравцеві точно визначати напрямок джерела звуку, відчувати просторову глибину, а також реалістичність середовища через реверберацію та ефекти затухання. Це створювало унікальний ефект **присутності** — гравець не просто грав, а перебував у середині подій.

Роль саунд-дизайну в іграх цього періоду суттєво зросла. Звук перестав бути фоновим елементом і почав сприйматися як повноцінна складова художнього образу. Цей процес перегукується з еволюцією звуку в кінематографі, де, за визначенням О. Бут, аудіо стає рівноправним

компонентом образної структури твору, здатним самостійно транслювати зміст [3]. Композитори та звукорежисери отримали змогу працювати з динамічною звуковою сценою, де музика, ефекти й атмосфера реагували на дії гравця, створюючи багатошаровий аудіовізуальний досвід.

Показовими прикладами інтеграції звуку в структуру ігрового процесу стали такі проєкти, як **Half-Life**, **Silent Hill**, **Deus Ex**. У цих іграх саме аудіо виконувало ключову роль у побудові атмосфери, створенні психологічної напруги й навіть формуванні наративу. У Silent Hill, наприклад, звук використовується як елемент страху — гравець чує те, чого ще не бачить, і це породжує тривогу. У Deus Ex аудіо підкреслює футуристичність світу, а в Half-Life — посилює кінематографічний характер сюжету.

Формування естетики сучасної гри

У 2000–2010-х роках відеоігри пережили якісну трансформацію, що визначила їхній сучасний вигляд. Розвиток цифрових технологій, поява потужних графічних карт і розширення можливостей програмного забезпечення призвели до формування цілісної інтегрованої системи створення ігор, де візуальні, звукові, сценарні та інтерактивні елементи злилися в єдине художнє середовище.

Ключову роль у цьому процесі відіграли високорівневі рушії нового покоління — Unity, Unreal Engine 3/4, CryEngine. Вони надали розробникам доступ до інструментів, що раніше вимагали складного програмування, а тепер дозволяли працювати з графікою, фізикою, звуком і сценарною логікою у зручному візуальному середовищі. Така інтеграція зробила можливим створення комплексного художнього продукту, у якому всі компоненти — від освітлення до звукового дизайну — працюють на спільну естетичну мету.

Паралельно з цим активно розвивались middleware-системи для роботи зі звуком, серед яких особливо виділяються FMOD Studio та Wwise. Вони стали основним інструментом професійних саунд-дизайнерів, відкривши шлях до інтерактивного аудіо, що змінюється у режимі реального часу залежно від дій гравця, стану персонажа чи ситуації у грі. Якщо раніше музика

звучала як фоновий супровід, то тепер вона перетворилась на динамічний елемент наративу, що підсилює емоційний вплив і створює відчуття живого, мінливого світу.

Завдяки цим технологіям художні засоби відеоігор перестали бути статичними. Вони набули адаптивного характеру — кожен звуковий чи візуальний елемент тепер міг трансформуватись залежно від контексту. Це стало основою нової форми сприйняття — ефекту присутності, коли гравець відчуває не лише залученість, а й співтворчість у процесі.

Особливої ваги набув кінематографічний підхід до режисури сцени. Починаючи з таких проєктів, як *Metal Gear Solid*, *Half-Life 2*, *Mass Effect* чи *Uncharted*, ігри почали використовувати прийоми, властиві кіно: сценарну побудову, акторську гру, міміку, крупні плани, систему діалогів і постановку світла. Це дозволило ігровим світам передавати емоції й психологічну глибину, наближаючи їх до драматургії фільму.

Проте на відміну від кіно, у відеоіграх гравець виступає не пасивним спостерігачем, а активним співтворцем. Саме від його дій залежить розвиток сюжету, поведінка персонажів і навіть звуковий ландшафт. Така особливість породила новий підхід до художнього проєктування — інтерактивну драматургію, у якій геймплей, музика, звук і візуал утворюють єдину систему взаємодії.

Сучасний етап: інтерактивне мистецтво

Сучасна індустрія відеоігор увійшла у нову фазу свого розвитку, де технологічні інновації тісно поєднуються з художніми експериментами. У 2010–2020-х роках ігри перестають бути виключно продуктом індустрії розваг і дедалі частіше сприймаються як повноцінна форма цифрового мистецтва. Як зазначає Н. Горовіц, відеоігри набули статусу культурних артефактів, що потребують кураторського підходу, збереження та осмислення на рівні з живописом чи кіно [23].

На цьому етапі спостерігається перехід до кінематографічного реалізму, який охоплює не лише графічну якість, але й режисуру, акторську гру, монтаж та звукове оформлення. Такі проєкти, як The Last of Us Part II, God of War Ragnarök, Cyberpunk 2077, Red Dead Redemption 2, демонструють неймовірну глибину емоційного впливу, де візуальна естетика, звук і драматургія зливаються в єдину структуру. Гравець не просто виконує дії, а проживає історію, відчуває внутрішній світ персонажів і занурюється у складно вибудовану емоційну реальність.

Водночас із розвитком великих ААА-проєктів активно формується незалежна (інді) сцена, що протиставляє масштабності корпоративних студій індивідуальне бачення та експериментальний підхід. Ігри на кшталт Journey, Inside, Gris, Limbo, Hollow Knight або Cuphead демонструють, що навіть із мінімальними ресурсами можна створити унікальний художній світ, у якому звук, музика та візуальна стилізація працюють як єдина поетична мова. Саме інді-сцена стала простором для народження нових естетичних тенденцій — символізму, мінімалізму, метафоричності та емоційної глибини.

Розвиток аудіотехнологій суттєво розширив можливості сприйняття ігрового світу. Сучасні системи 3D Audio, Dolby Atmos та технології просторового моделювання звуку (HRTF — Head Related Transfer Function) дозволяють створювати повноцінне відчуття простору, де кожен звук має свою локалізацію й рух у тривимірному середовищі. Гравець може орієнтуватися за слухом, реагувати на джерела звуку, сприймати відстань, напрям і навіть тип поверхні. Усе це формує нову якість занурення, де слухове сприйняття стає рівнозначним зоровому.

Особливої уваги заслуговує розвиток інтерактивної музики, яка більше не є фоновим супроводом, а виступає активним учасником ігрового процесу. За допомогою систем на кшталт FMOD Studio або Wwise, музичні теми можуть змінюватися залежно від динаміки бою, емоційного стану персонажа

чи навіть вибору гравця. Таким чином, музика стає “живою” системою, що реагує на дії користувача, підтримуючи ідею інтерактивного мистецтва.

Цей етап розвитку ігор характеризується повною інтеграцією художнього задуму з технічними інструментами. Саунд-дизайнер, композитор, художник і програміст працюють у тісній взаємодії, створюючи єдиний синтетичний продукт, де кожен елемент має як естетичне, так і функціональне значення. Звук тут уже не лише підтримує атмосферу, а й стає важливим носієм сенсу, частиною наративу та виразником емоцій.

Роль звукорежисера у життєвому циклі розробки гри

Впровадження аудіо вимагає від звукорежисера розуміння логіки ігрового рушія. Як зазначає Г. Таргетт, сучасний фахівець часто виступає мостом між творчою командою та програмістами, самостійно вирішуючи питання логіки відтворення звуку [42]. Його головна мета — забезпечити, щоб аудіо створювало необхідний настрій, занурювало гравця, було динамічним та захоплюючим, спонукаючи його повертатися до гри знову і знову. Для досягнення цієї мети звукорежисер тісно співпрацює з розробниками та дизайнерами.

Основні обов'язки звукорежисера охоплюють широкий спектр завдань:

Звуковий дизайн: Створення звукових ефектів, що відповідають діям та подіям у грі, включаючи звуки оточення, кроків, ефектів навколишнього середовища, зброї, вибухів тощо.

Музична композиція: Створення оригінальних музичних партитур, що посилюють настрій та емоційний вплив гри, доповнюючи геймплей, наратив та загальну тему.

Впровадження аудіо: Інтеграція звукових ресурсів у ігровий рушій, забезпечення належної синхронізації з внутрішньоігровими подіями та діями гравця. Це часто вимагає роботи з мовами програмування та скриптів для створення інтерактивних аудіосистем.

Просторове аудіо та 3D-звук: Використання технік просторового аудіо для створення відчуття напрямку та відстані для звуків у грі, що підвищує занурення гравця.

Редагування діалогів та озвучення: Обробка записів голосових акторів для забезпечення чіткого та якісного діалогу, а також синхронізація губ з анімацією персонажів.

Адаптивне та динамічне аудіо: Розробка та впровадження систем, де музика та звукові ефекти змінюються залежно від дій гравця або розвитку сюжету.

Забезпечення якості (QA): Участь у тестуванні гри для виявлення проблем, помилок або невідповідностей в аудіо та співпраця з командою для їх усунення.

Співпраця та комунікація: Ефективна взаємодія з іншими членами команди розробки (гейм-дизайнерами, художниками, програмістами) для узгодження аудіоелементів із загальним творчим баченням гри.

Інтеграція звукового дизайну на різних етапах розробки гри

Звук у відеоіграх відіграє ключову роль, яка часто залишається непоміченою, але саме він створює атмосферу, посилює занурення та спрямовує наратив. Звуковий дизайн — це не лише технічна задача, а й мистецька форма, що вимагає як глибокої технічної експертизи, так і тонкого художнього чуття. Аудіо становить половину того, як гравець сприймає історію: воно допомагає передати емоції, які мають виникати в конкретні моменти, а також робить зрозумілішими ключові події на екрані. Крім того, звук додає глибини та складності навіть тим елементам ігрового світу, які лишаються невидимими, формуючи враження від часу та простору.

Робота зі звуком починається ще на етапі передпродюсування. Саме тут формується аудіо-візія гри, і звукорежисери стають частиною команди з самого початку проєкту. Аудіо-директор розробляє концепцію для всіх аудіо-

дисциплін — голосів, звукового дизайну та музики, спираючись на тему й художні принципи майбутньої гри. Ця концепція оформлюється в документ, який стає своєрідним стилістичним та технічним орієнтиром. Технічні саунд-дизайнери в цей час вибудовують основу для подальшої роботи, тоді як композитори співпрацюють із командою розробників над створенням системи динамічної музики. Раннє залучення аудіо-фахівців дозволяє уникнути технічних обмежень у майбутньому та навіть надихає суміжні команди — художників чи аніматорів.

Під час етапу продюсування починається активна інтеграція та тестування звуку. Саунд-дизайнери створюють звукові ландшафти, амбієнти, унікальні ефекти, музику та діалоги, які стають частиною ігрового світу. Особливе місце займає робота у студії Фолі, де записуються реалістичні звуки — кроки, шум одягу, брязкіт зброї. Озвучення персонажів потребує окремої підготовки: написання сценаріїв, робота з режисером дубляжу, запис голосів акторів. На цьому етапі звукорежисер уже безпосередньо працює з усіма елементами гри, адаптуючи їх до геймплею.

Постпродюсування зосереджується на доопрацюванні та доведенні звуку до фінального вигляду. Важливими завданнями стають зведення та балансування всіх аудіо-елементів. Використовуються сучасні техніки імерсивного звучання — 3D-аудіо, моделювання поширення звуку, системи динамічного мікшування. Робота здійснюється у професійних програмах, таких як Pro Tools, Wwise чи FMOD. Водночас саме тут можуть проявлятися типові проблеми: спотворення, дисбаланс або технічні збої, які необхідно оперативно усунути.

На етапах альфа- й бета-тестування аудіо проходить ретельну перевірку. Тестери оцінюють баланс, якість звучання та стабільність відтворення на різних пристроях і конфігураціях. Тут важливим стає зворотний зв’язок: завдяки йому звукова команда може виправляти недоліки, регулювати гучність, оптимізувати ефекти чи навіть перезаписувати частину матеріалу.

Для аналітики застосовуються спеціалізовані інструменти — Wwise Profiler, FMOD Profiler, Unity Profiler та Unreal Engine Profiler.

Завершальним етапом є запуск гри, але навіть після релізу робота над звуком не припиняється. Команда моніторить якість аудіо, оперативно виправляє помилки та реагує на відгуки користувачів. Часто після виходу гри з'являються оновлення, які містять покращені аудіо-рішення, додаткові ефекти чи навіть нові музичні композиції. Створення DLC (додаткового завантажуваного контенту) потребує інтеграції нових звукових ресурсів, що фактично перетворює пострелізну фазу на продовження основного продюсування.

Етапи розвитку комп'ютерних ігор та їх техніко-художні трансформації

Еволюція комп'ютерних ігор є одним із найбільш показових прикладів того, як технічний прогрес — експоненційне збільшення обчислювальної потужності та постійне вдосконалення візуальних технологій — неминуче стимулює глибокі **художні трансформації**. На кожному етапі розвитку індустрії, звукорежисура не лише адаптувалася до нових технологічних стандартів, але й відігравала критично важливу роль у формуванні ігрового досвіду, особливо в нерозривній **синергії** з візуальним рядом та інтерактивними механіками, які становлять основу геймплею.

Ці трансформації започатковуються в **Аркадну та 8-бітну еру**, де домінувала **піксельна, абстрактна графіка**. У цей період звук виконував функцію, що насамперед **компенсувала візуальну обмеженість**. Оскільки примітивні звукові чіпи (засновані на хвильовому, або FM-синтезі, відомі як Chiptune) та мізерна пам'ять не дозволяли відтворювати реалістичні чи складні звукові пейзажі, звукорежисери мусили покладатися на високо стилізовану, часто **не-дієгетичну** музику. Вона не тільки створювала необхідну атмосферу (наприклад, відчуття швидкості у гонках або героїзму у платформерах), але й була критично важливою для забезпечення **геймплейної ясності**. Звуковий

сигнал про отримання шкоди, стрибок, збір бонусу або активацію здібності мав бути миттєво розпізнаваний. Через низьку роздільну здатність та обмежену візуальну інформативність, звук став первинним **носієм функціональної інформації**, перетворюючись на своєрідну аудіальну "іконку", що сигналізує про стан системи. Художнім завданням було створення лаконічних, але емоційно насичених звуків, які б міцно асоціювалися з діями персонажа, формуючи таким чином невід'ємну частину ідентичності гри.

Наступний, **3D-перехідний етап**, ознаменований появою консолей з носіями **CD-ROM**, кардинально змінив вимоги до звукового дизайну. Технологія CD дозволила інтегрувати високоякісне аудіо без значних обмежень за обсягом, що було неможливо на картриджах. З появою полігональних 3D-моделей та зростанням візуальної достовірності, звук мав відповідати новій реальності, фокусуючись на **побудові глибшої імерсії**. Синтетичні звуки почали замінюватися **записаними аудіофайлами** високої якості. Це дозволило реалізувати повноцінне **голосове озвучення** та впровадження **оркестрових партитур**, які до того були прерогативою кіно. Звук на цьому етапі перестав бути лише супроводом і став інструментом для **емоційного маніпулювання** гравцем, поглиблюючи наративну складову ігор. Саме тут вперше активно використовується **просторове аудіо** для позиціювання джерел звуку у 3D-сцені. Звукорежисура почала активно будувати **атмосферу** (наприклад, використання тиші та несподіваних звукових ефектів у жанрі Survival Horror), працюючи у прямому зв'язку з кат-сценами та сюжетними поворотами.

У **HD-еру та на сучасному етапі**, завдяки екстремальній обчислювальній потужності сучасних систем та спеціалізованому проміжному програмному забезпеченню (як-от **FMOD** або Wwise), звук перетворився на повністю **інтерактивну систему та геймплейний механізм**. Сьогодні від звукового пейзажу вимагається, щоб він не був заздалегідь записаним, лінійним треком, а **реагував у реальному часі** на кожен аспект стану

геймплею. Це реалізується через **адаптивне аудіо (Adaptive Audio)**: музичні композиції розкладаються на окремі шари (*stems*), які динамічно мікшуються залежно від швидкості руху гравця, рівня здоров'я, або близькості ворогів. Наприклад, при вході в бій система додає агресивні перкусії та баси, а під час дослідження – приглушує їх. Крім того, використовуються складні **системні параметри** для управління звуковими ефектами. Сюди відноситься: імітація деталізованої **акустики** (різні типи реверберації залежно від розміру приміщення), **оклюзія** (реалістичне затухання звуку, що проходить крізь стіни чи перешкоди), а також динамічне змішування звуків кроків залежно від типу поверхні. У багатокористувацьких та стелс-іграх, звук перейшов із категорії художнього доповнення до категорії **стратегічного ресурсу**. Здатність гравця точно визначити місцезнаходження візуально прихованого ворога лише за звуками кроків чи перезарядки, завдяки впровадженню **3D-позиціювання звуку**, є критичною перевагою. Таким чином, сучасний звуковий дизайн — це не просто супровід графіки, а складна, інтерактивна, **системно-орієнтована** структура, що безпосередньо впливає на ігровий процес та глибину занурення.

Процес звукового оформлення гри

Процес створення звукового оформлення у відеоіграх є складним і багаторівневим. Він охоплює широкий спектр завдань і вимагає постійної співпраці між різними фахівцями: композиторами, звукорежисерами, технічними саунд-дизайнерами, програмістами та навіть сценаристами. Саме ця командна робота дозволяє сформувати цілісний аудіовізуальний досвід, де звук не існує окремо, а органічно переплітається з усіма іншими складовими гри.

Першим етапом стає концептуалізація. Це фундамент, на якому вибудовується уся подальша аудіо-стратегія. На цьому етапі розробляються ключові музичні теми, гармонії, визначається стиль та загальний настрій гри. Дуже часто композитори й гейм-дизайнери проводять спільні воркшопи та сесії мозкового штурму, де обговорюють візуальну стилістику, сюжетні

акценти й навіть психологічний вплив звукових рішень. Використовуються референси з інших медіа — від фільмів до попередніх ігор, — щоб краще окреслити бажану атмосферу. У результаті формується початкова аудіо-візія, яка задає напрямок для всієї подальшої роботи.

Після цього розпочинається етап запису та редагування. Тут використовуються різні техніки залежно від потреб гри. Наприклад, студія Фолі дозволяє відтворювати повсякденні звуки — кроки, шелест тканини, металевий брязкіт — і робити їх максимально реалістичними. Польовий запис відкриває можливість захоплення живих звуків навколишнього середовища: шуму міста, співу птахів, звуків природи. Для створення унікальних ефектів активно застосовується синтез звуку за допомогою програмних плагінів. На цьому ж етапі всі записи проходять ретельну обробку: очищення від шумів, редагування, еквалізацію, компресію та додавання реверберації. Використання цифрових аудіо-робочих станцій (DAW) дає змогу досягти максимальної якості звучання, яка згодом визначатиме переконливість усього ігрового світу.

Коли основний аудіо-матеріал готовий, настає час інтеграції у сам ігровий рушій. Цей процес набагато складніший, ніж просте завантаження звукових файлів. Завданням є зробити звук частиною геймплею, щоб він реагував на події у грі. Наприклад, кроки персонажа повинні звучати інакше залежно від типу поверхні, а музика — змінюватися відповідно до рівня небезпеки чи спокою. Для цього застосовуються спеціалізовані аудіо-мідлвери, такі як Wwise чи FMOD, які дозволяють створювати адаптивні звукові системи. Тут же налаштовуються події, тригери та поведінка аудіо у відповідь на внутрішньоігрові змінні. Завдяки цьому звук стає динамічним і живим, а ігровий досвід — набагато реалістичнішим.

Однак інтеграція не завершує процес. Дуже важливим етапом є оптимізація та налагодження. Звукове оформлення повинно бути не лише красивим і якісним, але й технічно стабільним, адже перевантаження системи великою кількістю аудіо-ресурсів може призвести до затримок або

спотворень. Тому на цьому етапі проводиться стиснення файлів, налаштовується потокове відтворення, тестується продуктивність на різних апаратних платформах. Проблеми синхронізації, дисбаланс гучності чи небажані артефакти виправляються за допомогою профілювання. Спеціалізовані інструменти, такі як Wwise Profiler чи Unity Profiler, дозволяють детально відстежити, як поводиться звук під час гри.

Окремо варто згадати роль систем контролю версій (Git, Perforce, Plastic SCM), які забезпечують організовану роботу з великими аудіо-бібліотеками. Завдяки їм команда може ефективно співпрацювати, синхронізуючи всі зміни та уникати втрати даних.

РОЗДІЛ 2. САУНД-ДИЗАЙН У КОМ'ЮТЕРНИХ ІГРАХ

2.1. Основні функції звукового оформлення ігрового середовища

Звукове оформлення у відеоіграх виконує не лише роль супроводу, але й виступає повноцінним інструментом формування ігрового досвіду. Звукове оформлення здатне не тільки підсилювати візуальний ряд, створюючи ефект, який М. Шион визначає як «додана вартість» (added value) — коли звук збагачує зображення інформацією, якої немає у самій картинці [13], а й самостійно формувати атмосферу.

Функції саунд-дизайну в іграх можна поділити на кілька основних напрямів. **Атмосферна функція** забезпечує створення емоційного фону та підсилює художній стиль гри (sci-fi, хорор, фентезі). Як зазначає В. Віттінгтон, саме звуковий дизайн часто виступає головним інструментом легітимізації вигаданого всесвіту, надаючи фантастичним елементам фізичної ваги та реалістичності [46]. **Ідентифікаційна функція** дозволяє закріплювати за персонажами, локаціями чи фракціями власні звукові теми, а також формує музичне «обличчя» бренду (як у серіях *Halo* чи *The Elder Scrolls*). **Інформаційно-комунікативна функція** допомагає гравцеві орієнтуватися у просторі та отримувати важливі сигнали про ігрові події. **Інтерактивна функція** проявляється у здатності аудіо реагувати на дії гравця та адаптуватися до темпу геймплею. **Імммерсійна функція** занурює у віртуальний світ через реалістичні звукові ефекти та створює відчуття «присутності». **Психологічна функція** спрямована на викликання необхідних емоцій — від страху до драйву. **Ігрово-механічна функція** робить звук частиною самого геймплею, а **навчальна** — допомагає інтуїтивно опановувати ігрові механіки за допомогою аудіо-підказок.

Атмосферна функція звукового оформлення відеоігор

Однією з ключових функцій саунд-дизайну у відеоіграх є створення емоційного фону, який формує атмосферу ігрового світу та безпосередньо

впливає на сприйняття подій гравцем. Звукове оформлення у сучасних іграх охоплює набагато більше, ніж просто музичний супровід: воно включає ефекти довкілля, такі як шум вітру, дощу, шелест листя чи відлуння кроків у порожніх приміщеннях; звуки персонажів, їхніх дій, реакцій та взаємодії з об'єктами; а також різноманітні події в ігровому світі, від активації пасток до магічних ефектів. Всі ці елементи разом утворюють цілісну аудіальну картину, де кожен звук виконує певну роль і взаємодіє з іншими складовими.

Створення емоційного фону

Аудіо виступає важливим елементом у створенні загального художнього образу гри. Музика і звукові ефекти здатні викликати у гравця відчуття страху, напруги, захоплення чи меланхолії, які не завжди можливо передати лише візуальними засобами. За словами К. Коллінза (*Collins, 2008*), «музика у відеоіграх є не лише прикрасою, але й активним інструментом впливу на настрій та емоційний стан гравця» [26].

Наприклад, у жанрі survival horror музика з мінімалістичними мотивами, використанням дисонансів та низькочастотних звуків сприяє виникненню відчуття тривожності та загрози (*Silent Hill 2*, композитор Акіра Ямаока). Натомість у фентезійних RPG (наприклад, *The Elder Scrolls V: Skyrim*) оркестрова музика з епічними хоральними партіями створює величну та героїчну атмосферу.

Звук як засіб занурення в ігровий простір

Важливим аспектом створення повного ефекту занурення у відеоігри є не лише музичний супровід, а й звукові ефекти довкілля (ambient sound design), які, за визначенням фахівців індустрії [41], формують аудіальне середовище, що дозволяє гравцеві відчути простір гри більш реалістично.. Саме вони формують аудіальне середовище, яке дозволяє гравцеві відчути простір гри більш реалістично та багатогранно. Звуки дощу, шелест листя, шум вітру, віддалені кроки чи голоси тварин створюють невидиму аудіальну текстуру, що взаємодіє з візуальними ефектами, посилюючи відчуття присутності у віртуальному світі.

Як зазначає Р. Серрі (*Surry, 2016*), «атмосферний саунд-дизайн формує невидимий каркас ігрового світу, який гравець не завжди усвідомлює, але постійно відчуває» [16]. Тобто звуки довкілля виконують не лише декоративну функцію, а й слугують психологічним орієнтиром для гравця, дозволяючи йому відчути масштаб, глибину та характер простору, в якому він знаходиться. Вони можуть сигналізувати про небезпеку, відстань до об'єктів, наявність інших персонажів або навіть змінювати сприйняття настрою сцени.

Наприклад, у грі The Last of Us Part II використання детально прописаних ambient-звуків — пташиний спів, відлуння кроків у порожніх будівлях, шелест листя на вітрі, слабке дзюрчання води — допомагає створити меланхолічну та постапокаліптичну атмосферу, що підсилює драматичну напругу сюжету. Кожен звук працює у взаємодії з іншими елементами аудіо, підкреслюючи ізольованість, небезпеку та емоційний стан персонажів. Крім того, такий підхід дозволяє гравцеві неусвідомлено сприймати світ гри як живий та динамічний, що значно підвищує рівень занурення та емоційної залученості.

Емоційна динаміка та адаптивність

Особливістю відеоігор є те, що звукова атмосфера може змінюватися динамічно у залежності від дій гравця. За словами М. Грімшоу (*Grimshaw, 2011*), «аудіо у відеоіграх функціонує як інтерактивний інструмент, що здатен підлаштовуватися під темп і характер геймплею» [48]. Це означає, що музика може переходити від спокійних фонових мотивів до напружених і драматичних тем у моменти бою чи небезпеки.

Наприклад, у *DOOM Eternal* (композитор Мік Гордон) heavy-metal саундтрек стає інтенсивнішим у міру загострення бою, що створює відчуття драйву та екстремальної дії. У свою чергу, у *Resident Evil Village* перехід від тиші до різких звукових акцентів використовується як засіб лякання та підсилення хорор-ефекту.

2.2 Характерні особливості інтерактивної музики у відеоіграх

Інтерактивна музика є однією з найважливіших складових сучасного аудіодизайну у відеоіграх. Її сутність полягає у здатності музичного супроводу **реагувати на дії гравця**, зміни ігрового середовища або розвиток сюжету. Як зазначає В. Філліпс, на відміну від традиційної лінійної музики, інтерактивна композиція функціонує як гнучка система фрагментів та шарів, що дозволяє їй адаптуватися до непередбачуваних дій гравця [33].

У класичних аудіовізуальних творах, таких як кіно, телесеріали чи театральні постановки, музика зазвичай має заздалегідь визначений темп, форму, гармонічний та ритмічний розвиток, який залишається незмінним під час кожного перегляду. Композиція у цих випадках побудована таким чином, щоб підтримувати драматургію сюжету і викликати у глядача певні емоції, але її структура завжди лінійна та передбачувана. Кожен перегляд фільму забезпечує однакове звучання, що дозволяє режисеру та композитору точно контролювати сприйняття сцени.

У відеоіграх ситуація принципово відрізняється. Гравець має свободу дій, тому події у віртуальному світі розгортаються непередбачувано й можуть змінюватися від одного проходження до іншого. Через це музика не може залишатися статичною: вона повинна бути адаптивною, тобто здатною змінювати свій характер, темп, динаміку, гармонію та інструментальне забарвлення залежно від того, що відбувається на екрані в реальному часі. Саме ця властивість робить інтерактивну музику відмінною від традиційних саундтреків і перетворює її на живий, динамічний компонент ігрової реальності, який реагує на дії користувача.

Інтерактивна музика здатна реагувати на широкий спектр параметрів, що визначають стан ігрового світу: місцезнаходження персонажа, рівень небезпеки, інтенсивність бою, кількість ворогів, швидкість руху, стан здоров’я героя, а іноді навіть на емоційні аспекти сюжету. Завдяки цьому вона може

виконувати низку функцій: підсилювати драматургію, передавати атмосферу, орієнтувати гравця у просторі, а також формувати загальний настрій гри.

Яскравим прикладом адаптивного підходу до створення інтерактивної музики є гра The Elder Scrolls V: Skyrim (Bethesda Softworks, 2011). У цьому масштабному рольовому проєкті музика виступає не лише тлом, а й активним елементом ігрового процесу, який безпосередньо реагує на стан гравця, його дії та події у світі гри. Музичне оформлення створює відчуття живого, дихаючого простору, що постійно змінюється — воно ніби «співіснує» з гравцем, реагуючи на кожен його крок. Під час дослідження відкритого світу звучать спокійні оркестрові композиції з плавними, протяжними мелодіями та м’якою гармонією. Такі музичні теми формують атмосферу величі, свободи та спокою, що ідеально відображає епічний масштаб фентезійного всесвіту Тамріелю. Вони допомагають гравцеві зануритися у світ, викликаючи відчуття гармонії з природою та відпочинку від бойових пригод.

Коли ж гравець вступає у бій, система аудіо у реальному часі змінює музичний шар — у композиції додаються ударні інструменти, драматичний хор та інтенсивні ритмічні елементи. Це створює відчуття напруги, небезпеки й адреналіну, стимулюючи гравця діяти швидше й рішучіше. Зв'язок між ігровими змінними (Game Parameters) та музичними характеристиками було налаштовано за методикою М. Світа [49], де параметри гри (наприклад, "відстань до ворога") безпосередньо керують гучністю окремих інструментальних груп (stems) у мікшері FMOD. Після завершення сутички музика поступово «заспокоюється»: гучність зменшується, темп уповільнюється, зникають агресивні ударні, і звучання плавно переходить назад у мирну атмосферу дослідження. Цей перехід виконується так делікатно, що слухач майже не помічає моменту зміни. Завдяки такій динамічній побудові, музичний супровід у Skyrim не сприймається як щось зовнішнє — він стає частиною ігрового світу, підтримує емоційний стан гравця й підсилює занурення у події.

Не менш показовим прикладом інтерактивного підходу до створення музики є гра Journey (*Thatgamecompany, 2012*), у якій музика виконує надзвичайно важливу роль — вона не лише підсилює емоційне сприйняття, а й частково замінює вербальні засоби виразності, тобто сюжетні діалоги чи голосові підказки. У цьому проєкті гравець не отримує звичних реплік чи текстових інструкцій, натомість саме музика спрямовує його емоційний стан і допомагає зрозуміти сенс подій, що відбуваються. Композитор Остін Вінторі (Austin Wintory) розробив надзвичайно гнучку систему музичних переходів, де кожна частина гри має власний тематичний розвиток, але водночас органічно поєднана з іншими сегментами. Його музика не просто супроводжує події, а взаємодіє з ними, реагуючи на поведінку гравця, швидкість пересування, зміни в ігровому середовищі та навіть на моменти взаємодії з іншими мандрівниками у світі гри.

Особливістю підходу Вінторі є динамічне оркестрове забарвлення: коли гравець рухається спокійно або зупиняється, музика набуває мінімалістичного, камерного звучання — тихі струнні інструменти, плавні гармонії, відчуття самотності й медитативного спокою. Проте у моменти емоційного піднесення або під час взаємодії з іншими гравцями оркестровий склад розширюється, з’являються мідні духові, хор, наростає динаміка — музика «дихає» разом із гравцем, підкреслюючи кожен його крок.

Завдяки такій побудові саундтреку, Journey стає прикладом того, як музика може виконувати роль “емоційного оповідача”, який веде гравця крізь історію без жодного слова. Це не просто фоновий супровід, а повноцінний елемент наративу, який допомагає сформувати унікальний емоційний досвід. У певному сенсі, музика у Journey є “співрозмовником” гравця — вона відчуває його стан, реагує на нього, підтримує у моменти тривоги й підносить у миті натхнення.

На відміну від лінійного підходу, що характерний для кіно або традиційних мультимедійних форм, де музика є статичною й відтворюється однаково під час кожного перегляду, інтерактивна музика у відеоіграх

пропонує абсолютно інший принцип побудови. У такому випадку музика перестає бути лише фіксованим фоновим елементом і перетворюється на динамічну систему, здатну змінюватися в реальному часі залежно від ігрової ситуації. Один і той самий музичний твір може звучати по-різному під час кожного проходження гри, адже на його структуру впливають безліч факторів: дії гравця, його вибір, швидкість пересування, перемоги чи поразки у боях, зміна локацій або навіть настрій сцени. У результаті створюється унікальний музичний досвід, який є неповторним для кожного користувача. Це додає грі глибини та забезпечує більш природну емоційну реакцію, оскільки музика ніби "співпереживає" подіям разом із гравцем. Подібна варіативність звучання дозволяє уникнути монотонності, підтримує зацікавленість користувача протягом усього ігрового процесу та сприяє більш глибокому зануренню у віртуальний простір. Завдяки цьому інтерактивна музика стає не просто частиною звукового оформлення, а повноцінним елементом ігрового дизайну, що формує емоційний ритм і динаміку ігрового світу.

РОЗДІЛ 3. ІМПЛЕМЕНТАЦІЯ ЗВУКОВОГО ДИЗАЙНУ ТА ІНТЕРАКТИВНОЇ МУЗИКИ У ІГРОВИЙ ПРОТОТИП

3.1. Розробка ігрового прототипу в середовищі Unity

Ігровий рушій Unity: архітектура та можливості

Unity є одним із найпопулярніших багатоплатформних ігрових рушіїв сучасності, що надає розробникам потужний інструментарій для створення інтерактивних додатків різної складності. Рушій був вперше представлений у 2005 році компанією Unity Technologies і з того часу став стандартом індустрії для розробки як інді-проєктів, так і AAA-ігор.

Архітектурні особливості Unity

Unity побудований на компонентно-орієнтованій архітектурі, що дозволяє розробникам гнучко конструювати ігрові об'єкти через комбінування різноманітних компонентів. Основою цієї системи є GameObject – базова сутність, яка сама по собі є порожнім контейнером, але набуває функціональності через приєднані до неї компоненти. Такий підхід забезпечує модульність коду, спрощує повторне використання функціоналу та полегшує підтримку проєкту.

Система скриптування та логіка гри

Unity підтримує скриптування на мові C#, що надає розробникам повний контроль над логікою гри. Скрипти можуть бути приєднані до GameObject'ів як компоненти і взаємодіяти з іншими компонентами через добре документоване API. Система життєвого циклу скриптів включає методи на кшталт Start(), Update(), FixedUpdate(), LateUpdate(), що дозволяє точно контролювати момент виконання коду.

Вбудована аудіосистема Unity

Архітектура аудіосистеми

Аудіосистема Unity побудована на базі потужного Audio Engine, що забезпечує низькорівневу обробку звуку в реальному часі. Система

інтегрована безпосередньо в ігровий цикл і оптимізована для роботи на різних платформах, автоматично адаптуючись до апаратних можливостей цільового пристрою. Основою системи є декілька ключових компонентів, які працюють у тісній взаємодії для створення комплексного звукового середовища.

Обмеження вбудованої системи

Попри свою доступність, вбудована аудіосистема Unity має серйозні обмеження для складних проєктів. Вона не підтримує адаптивну музику, а також горизонтальне й вертикальне ремікшування, що ускладнює створення інтерактивних музичних переходів. Параметризація звуку обмежена: важко створювати системи з численними варіаціями та залежностями. Немає зручної системи подій для синхронізації звуку з анімацією та логікою, а робота з великими бібліотеками стає незручною через нестачу інструментів організації.

Через ці обмеження часто потрібна інтеграція middleware-рішень на кшталт FMOD або Wwise, які надають професійні інструменти для складних інтерактивних звукових систем.

Middleware-рішення FMOD Studio

Інтеграція FMOD та Unity: взаємодія систем

Інтеграція FMOD в Unity здійснюється через офіційний FMOD Unity Integration Package, який забезпечує двосторонній зв’язок між Unity та FMOD Engine з мінімальними витратами ресурсів. Пакет містить готові компоненти, Editor-інструменти, скрипти та повноцінне C# API, що дозволяє використовувати можливості FMOD без роботи з низькорівневими системами. Архітектура інтеграції складається з трьох рівнів: на верхньому рівні розташовані Unity-компоненти, які можна використовувати без коду; середній рівень представлений C# wrapper API, що спрощує взаємодію з FMOD; нижній рівень надає доступ до FMOD Core API для точного контролю. Система автоматично керує життєвим циклом FMOD Engine, ініціалізуючи його при запуску застосунку та коректно завершуючи роботу при виході. Runtime

Manager забезпечує централізований доступ до FMOD system, керує банками та синхронізує FMOD із Unity Update. Інтеграція також включає Editor-інструменти для перегляду структури FMOD-проєкту, перевірки посилань та виявлення помилок ще на етапі розробки..

FMOD Studio Listener – аудіальна перспектива

FMOD Studio Listener — це ключовий компонент, що замінює стандартний Unity Audio Listener і визначає позицію та орієнтацію слухача в 3D-просторі для FMOD. Він передає двигуну точну позицію, орієнтацію, швидкість для коректного ефекту Доплера та додаткові параметри камери чи персонажа. На відміну від Unity Audio Listener, FMOD дозволяє використовувати декілька слухачів одночасно, що важливо для ігор із розділеним екраном або складними аудіосценаріями. Система інтерполює рух слухача між кадрами, забезпечуючи плавний просторовий звук. Через attenuation range можна регулювати дальність обробки джерел, а Rigidbody velocity tracking дозволяє FMOD точно визначати швидкість руху для фізично коректного Доплера. У сценаріях з кількома слухачами використовується weighted blending — комбінування перспектив із різними вагами, що дає можливість плавно переключати камери або створювати збалансований звук у кооперативі. Listener masks дозволяють визначати, які джерела звучатимуть для конкретного слухача, створюючи різні звукові перспективи чи асиметричні аудіальні механіки.

Studio Event Emitter – джерела FMOD-подій

Studio Event Emitter — основний компонент для відтворення FMOD Events у Unity, який слугує зв’язком між GameObject та звуковою подією з FMOD Studio. Він під’єднується до будь-якого об’єкта сцени та дозволяє вибрати потрібний Event через Inspector, автоматично отримуючи його з FMOD-проєкту. Компонент постійно оновлює позицію, швидкість і орієнтацію об’єкта, забезпечуючи коректне 3D-позиціонування, доплерівський ефект і природну зміну звучання при русі джерела або слухача. Emitter підтримує різні режими запуску: автоматичний старт при активації об’єкта, роботу через

тригерні події для створення звукових зон, а також повний програмний контроль для складних ігрових сценаріїв, де поведінка звуку залежить від стану персонажа або логіки геймплею.

Параметр Allow Fadeout дозволяє завершувати відтворення плавно, використовуючи вбудовані fade-криві Event, що робить зупинку звуку музично коректною та запобігає раптовим обривам. Preload Event Data забезпечує завчасне завантаження звукових банків і метаданих, усуваючи затримку при першому запуску, що особливо важливо для миттєвих звуків, таких як постріли чи UI-ефекти. Через Override Attenuation можна перевизначити параметри загасання без зміни вихідного Event, налаштовуючи дальність або форму спадання для конкретного екземпляра. Event Instance Management надає доступ до операцій з екземпляром — повторний запуск, пауза, продовження чи перемотування — а Initial Parameter Values дозволяють зручно задавати стартові значення FMOD-параметрів прямо в Inspector, забезпечуючи варіативність звуку без додаткових скриптів.

Передача параметрів між Unity та FMOD

Система параметрів у FMOD є основним механізмом зв’язку між ігровою логікою Unity та аудіосистемою, дозволяючи створювати адаптивний звук, що миттєво реагує на зміни стану гри. Через інтеграційний C# API розробник може змінювати як локальні параметри конкретних Event instances, так і глобальні параметри, які впливають на весь мікс. Локальні значення встановлюються через SetParameter на екземплярі Studio Event Emitter, використовуючи ім’я параметра або константи, при цьому FMOD автоматично інтерполює зміни на основі seek speed та automation-кривих. Параметри можуть оновлюватися щокадру або за подіями, забезпечуючи плавні й природні переходи без артефактів. Це дозволяє будувати складні адаптивні системи: звук двигуна може змінювати тон, гучність і тембр відповідно до параметра Engine RPM; музика може динамічно посилюватися через керування параметром Tension, реагуючи на небезпеку; атмосферні шари можуть змінюватися при переході між частинами доби через глобальний

параметр Time of Day. Такий підхід забезпечує глибоку інтеграцію звуку з геймплеєм та створює живе, контекстно чутливе аудіосередовище.Глобальні параметри встановлюються через RuntimeManager API і доступні всім Events одночасно, що робить їх ідеальним інструментом для координації звукової поведінки на рівні всієї гри. Типові застосування глобальних параметрів включають погодні умови, які впливають на реверберацію, фільтрацію та щільність амбієнту одночасно для всіх звукових джерел, час доби, який контролює баланс між денними та нічними звуковими шарами глобально, рівень інтенсивності геймплею, який модулює музичну структуру та щільність звукових ефектів, стан здоров'я або ресурсів гравця, що впливає на характер музики та додає distortion або low-pass фільтрацію до всіх звуків при критичному стані. Система автоматично інтерполює значення параметрів згідно з налаштуваннями в FMOD Studio, забезпечуючи плавні переходи без різких стрибків навіть при значних змінах цільових значень.

Для оптимізації продуктивності параметри можуть бути кешовані та оновлюватися лише при реальній зміні значення, уникаючи непотрібних API викликів та обробки на стороні FMOD. Label-based parameters дозволяють використовувати осмислені текстові мітки замість числових значень для дискретних параметрів, що робить код більш читабельним та самодокументованим. Parameter automation in FMOD Studio означає, що складна багатомірна реакція звуку на зміну параметра визначається звуковим дизайнером без необхідності програмувати кожен аспект поведінки окремо, розділяючи відповідальність між творчою та технічною командами.

Studio Global Parameter Trigger – автоматизація параметрів

Studio Global Parameter Trigger представляє собою потужний компонент для автоматизації поведінки глобальних параметрів без необхідності написання скриптового коду, що значно спрощує створення динамічних звукових зон, інтерактивних областей та контекстно-залежних звукових трансформацій. Компонент дозволяє автоматично змінювати значення одного або декількох глобальних параметрів при виникненні певних умов або подій

Unity, таких як вхід об'єкта в тригерну зону через фізичний Collider, активація або деактивація GameObject, досягнення певної точки в анімації, або інші стандартні Unity callbacks. Наприклад, можна налаштувати тригер на вході персонажа в певну географічну зону через Trigger Collider, що автоматично змінить глобальний параметр Location з значення "Outdoor" на "Cave", що в свою чергу спричинить комплексну трансформацію всього звукового середовища: зміну амбієнтного фону з лісових звуків на печерну атмосферу з краплями води та відлунням, активацію іншого набору випадкових звукових подій через Scatterer Instruments, перемикання реверберації на preset з характеристиками замкнутого кам'яного простору, та модуляцію частотного балансу всіх звуків для відображення акустики підземелля.

Це значно спрощує створення складних звукових зон та акустичних областей без необхідності писати додатковий спеціалізований код, дозволяючи геймдизайнерам та левел-дизайнерам самостійно визначати звукову поведінку різних частин рівня безпосередньо в Unity Editor через візуальні інструменти. Компонент підтримує множинні параметри одночасно, дозволяючи координовано змінювати декілька аспектів звукової системи в відповідь на одну подію, створюючи узгоджені та комплексні звукові трансформації. Transition curves визначають характер зміни параметра від початкового до цільового значення, підтримуючи лінійні, експоненційні, логарифмічні або довільні bezier криві для створення музично та перцептивно приємних переходів.

Tag filtering дозволяє обмежити спрацювання тригера лише для об'єктів з певними Unity Tags, що надає точний контроль над тим, які сутності можуть активувати звукову трансформацію. Наприклад, тригер підводної акустики може реагувати лише на об'єкт з тегом "Player", ігноруючи інших персонажів або physics objects, які також можуть потрапляти у воду. Condition-based triggering дозволяє визначити додаткові умови активації через прості scripted expressions або посилання на інші компоненти, створюючи складніші логічні схеми без повноцінного програмування. Reset on exit опція автоматично

повертає параметр до попереднього значення при виході з тригерної зони, забезпечуючи коректну звукову поведінку при переміщенні між різними акустичними областями.

FMOD Studio Banks Manager автоматично інтегрується в Unity як singleton система, яка централізовано управляє завантаженням, вивантаженням та життєвим циклом FMOD банків протягом всього часу роботи програми, забезпечуючи оптимальне використання пам'яті та мінімізацію часу завантаження. За замовчуванням Master Bank, який містить глобальні налаштування аудіосистеми, структуру мікшера з усіма групами та шинами, та часто використовувані загальносистемні Events, завантажується автоматично при старті гри під час ініціалізації FMOD runtime, забезпечуючи доступність базової аудіоінфраструктури з самого початку роботи програми. Додаткові тематичні або рівень-специфічні банки можуть завантажуватися динамічно за потребою через простий API, що дозволяє ефективно керувати пам'яттю та завантажувати лише той контент, який необхідний для поточної ігрової ситуації, рівня або сцени.

Програмне завантаження банків здійснюється через RuntimeManager з підтримкою як синхронного, так і асинхронного завантаження залежно від вимог до продуктивності та допустимості блокування основного потоку гри. Синхронне завантаження блокує виконання до повного завантаження банку в пам'ять, що гарантує миттєву доступність всього вмісту банку після завершення операції, але може викликати помітне заморожування на великих банках. Асинхронне завантаження виконується в фоновому потоці без блокування основного game loop, дозволяючи грі продовжувати роботу та рендеринг під час завантаження звукових даних, що критично важливо для підтримання плавності та відгуку інтерфейсу, особливо на платформах з повільними накопичувачами або при завантаженні великих обсягів даних.

Система автоматично відстежує залежності між банками, забезпечуючи завантаження всіх необхідних dependency banks перед завантаженням основного банку, оскільки Events в одному банку можуть посилатися на

звукові дані або інші Events з інших банків через систему cross-bank references. Bank reference counting автоматично управляє lifetime банків, дозволяючи безпечно запитувати завантаження одного банку з декількох місць коду без ризику передчасного вивантаження, поки існує хоча б одне активне посилання. Вивантаження банку звільняє всю пам'ять, зайняту його звуковими даними, метаданими та event descriptions, але коректно обробляє ситуації, коли деякі event instances з цього банку ще активні, дозволяючи їм коректно завершитися перед фінальним звільненням ресурсів.

Studio Bank Loader – динамічне завантаження звуку

Studio Bank Loader є спеціалізованим компонентом, який значно спрощує управління завантаженням банків на рівні Unity сцен через декларативний підхід та автоматичну інтеграцію з Unity Scene Management системою. Можна прикріпити компонент до будь-якого GameObject в сцені, часто до порожнього об'єкта-менеджера або кореневого об'єкта сцени, і вказати через Inspector interface, які банки повинні завантажуватися автоматично при активації цього об'єкта та відповідно при завантаженні сцени, що містить цей об'єкт. При використанні стандартної Unity Scene Management системи для переходів між рівнями або областями гри, Bank Loader автоматично інтегрується в lifecycle сцен, завантажуючи необхідні банки синхронно з завантаженням геометрії, освітлення та інших активів сцени, та вивантажуючи їх при вивантаженні або заміні сцени, забезпечуючи автоматичну та безпомилкову синхронізацію звукових ресурсів з візуальним контентом без необхідності ручного управління з боку розробника.

Параметр Load Timing визначає точний момент завантаження банку в lifecycle GameObject: режим Explicit вимагає явного виклику методу Load через скрипт, надаючи повний контроль над моментом завантаження для складних сценаріїв з специфічними умовами; режим On Enable завантажує банк в момент активації компонента через OnEnable callback, що відбувається при активації GameObject після створення або повторної активації після деактивації; режим On Awake завантажує банк якомога раніше в lifecycle

об'єкта під час Awake phase, що гарантує доступність звукових ресурсів ще до виконання Start методів інших компонентів. Параметр Unload Timing контролює поведінку вивантаження: автоматичне вивантаження при деактивації об'єкта звільняє пам'ять негайно при виході зі сцени або деактивації контролера, тоді як ручне вивантаження дозволяє тримати банк в пам'яті довше для випадків, коли той самий контент може знадобитися найближчим часом.

Collision detection дозволяє Bank Loader виявляти ситуації, коли декілька активних loader instances намагаються керувати одним банком одночасно, попереджаючи про потенційні проблеми з управлінням ресурсами в Editor через warning messages. Preload Sample Data опція змушує систему повністю декодувати та завантажити в пам'ять всі стиснені звукові дані з банку одразу при завантаженні, замість ледачої декомпресії при першому використанні, що збільшує час та обсяг початкового завантаження, але гарантує нульову латентність при першому відтворенні будь-якого звуку з банку.

Event References та організація коду FMOD та ігрового рушія Unity

FMOD Unity Integration включає розвинену систему Event References, яка представляє собою type-safe механізм для роботи зі звуковими подіями в коді та Unity Inspector, що радикально покращує надійність, підтримуваність та рефакторінг-дружність аудіосистеми. Замість використання текстових рядків з шляхами до Events, які є вразливими до друкарських помилок, чутливими до регістру, ламаються при перейменуванні Events в FMOD Studio без можливості автоматичного оновлення посилань, та не забезпечують перевірки існування події на етапі компіляції, система Event References надає строго типізовані посилання, які валідуються на етапі розробки та автоматично оновлюються при зміні структури FMOD проєкту.

Event Reference поля в Inspector відображаються як зручні dropdown меню, автоматично заповнені всіма доступними Events з поточного FMOD проєкту з можливістю швидкого пошуку та фільтрації, що виключає можливість ручних помилок при введенні шляхів та забезпечує візуальну

навігацію по структурі звукових подій безпосередньо в Unity Editor. Система автоматично генерує статичні константи або enum-подібні структури для всіх Events проєкту, дозволяючи використовувати автодоповнення IDE та compile-time перевірку посилань, що значно знижує ймовірність runtime помилок, пов'язаних з некоректними шляхами до звукових ресурсів. При рефакторингу FMOD проєкту з перейменуванням або переміщенням Events, система автоматично оновлює всі Event References в Unity сценах та prefabs під час reimport операції, забезпечуючи консистентність посилань без необхідності ручного пошуку та заміни в кожному файлі проєкту.

Asset validation tools перевіряють цілісність всіх Event References в проєкті, виявляючи broken links, застарілі посилання на видалені Events, або відсутні банки, що містять необхідні дані, з генерацією детальних звітів про проблеми та їх локалізацію в ієрархії сцен. Batch operations дозволяють масово оновлювати або замінювати Event References по всьому проєкту для глобальних змін звукового дизайну або реорганізації структури Events. Event browser window в Unity надає централізований інтерфейс для перегляду всіх використань конкретного Event в проєкті, допомагаючи розуміти залежності та вплив потенційних змін в звуковому дизайні на різні частини гри.

Профілювання та виправлення помилок

FMOD Unity Integration включає комплексний набір інструментів для моніторингу, аналізу та дебагу аудіосистеми безпосередньо в Unity Editor та під час runtime тестування, що значно спрощує виявлення проблем продуктивності, налагодження звукової логіки та оптимізацію використання ресурсів. Live Update є революційною функцією, яка дозволяє підключити FMOD Studio до запущеної гри в реальному часі через мережевий протокол і безперервно спостерігати стан всієї аудіосистеми: поточні активні Event instances з детальною інформацією про їх параметри, позицію в просторі та стан відтворення, значення всіх глобальних та локальних параметрів з можливістю спостереження їх динаміки в часі, детальні метрики навантаження на процесор з розбивкою по різних підсистемах FMOD Engine,

поточне використання оперативної пам'яті з категоризацією за типами даних, стан streaming системи з інформацією про активні потоки та затримки читання з диска, активність mixer groups та buses з візуалізацією рівнів сигналу та peak meters.

Режим Live Update дозволяє не лише пасивно спостерігати, але й активно втручатися в звукову систему працюючої гри: звукові дизайнери можуть змінювати Events безпосередньо в FMOD Studio, корегувати automation curves, додавати або видаляти ефекти, змінювати параметри мікшера, і миттєво чути результат цих змін в грі без необхідності перезапуску програми, перебудови банків або будь-яких інших проміжних кроків. Це радикально прискорює ітеративний процес звукового дизайну, дозволяючи експериментувати з налаштуваннями безпосередньо в контексті реальної гри з реальною ігровою логікою, складністю звукової сцени та специфічними сценаріями, які важко або неможливо відтворити в ізольованому середовищі FMOD Studio без повної симуляції ігрового контексту.

FMOD Profiler Window в Unity Editor надає детальну візуалізацію метрик продуктивності аудіосистеми через графіки в реальному часі, що дозволяє візуально ідентифікувати піки навантаження, виявляти bottlenecks та аномалії в поведінці звукової системи. CPU Usage Breakdown показує детальну декомпозицію процесорного часу, витраченого на різні операції: mixing and DSP processing для обчислення аудіосигналу та застосування ефектів, geometry and spatialization для розрахунків 3D-позиціонування та акустики, streaming and file I/O для операцій читання з диску та декодування стиснених форматів, system overhead для управління lifecycle звукових об'єктів та синхронізації. Memory Usage Visualization відображає поточне використання RAM з категоризацією за типами даних: sample data для розпакованих звукових хвиль в пам'яті, streaming buffers для тимчасових буферів потокового завантаження, event instances для метаданих активних звукових подій, geometry and reverb для даних просторової акустики.

Voice Activity Monitor показує реальну кількість одночасно активних голосів з розбивкою на реальні та віртуалізовані, дозволяючи оцінити ефективність voice stealing алгоритмів та виявити ситуації voice starvation або надлишкового споживання голосових ресурсів. Event Inspector надає детальний view окремих Event instances з можливістю спостереження їх внутрішнього стану: поточна позиція на Timeline для подій з структурованим контентом, значення всіх параметрів з історією змін, активні instruments та їх індивідуальні гучності, застосовані ефекти з поточними налаштуваннями. Console Logging інтегрує FMOD лог-повідомлення в Unity Console з налаштовуваним рівнем деталізації від критичних помилок до детального debug output, що спрощує діагностику проблем через централізований інтерфейс разом з іншими системами гри.

FMOD надає детальні та зрозумілі повідомлення про помилки з контекстною інформацією, що значно спрощує діагностику проблем: точне місце виникнення помилки в коді, тип операції, яка завершилася невдачею, залучені ресурси та їх стан, можливі причини проблеми та рекомендації щодо усунення. Error callback system дозволяє централізовано перехоплювати всі помилки FMOD через єдиний handler для логування, аналітики або користувацьких повідомлень без необхідності обробляти кожен API виклик окремо. Graceful degradation strategies дозволяють системі продовжувати роботу при часткових відмовах: якщо не вдається завантажити специфічний звуковий ефект, гра продовжує роботу мовчки замість краху, з опціональним fallback на placeholder звук або просто без звукового супроводу для конкретної події, зберігаючи gameplay experience максимально непорушеним.

Validation utilities перевіряють цілісність аудіоконфігурації при старті гри: наявність всіх необхідних банків у build, коректність посилань на Events, достатність виділених ресурсів для запланованого навантаження, з генерацією детальних warning або error messages при виявленні потенційних проблем. Assert and debug builds можуть включати додаткову перевірку інваріантів та

стану системи для раннього виявлення проблем під час розробки, яка автоматично вимикається в release builds для уникнення overhead.

Оптимізація продуктивності інтеграції

Інтеграція FMOD та Unity вимагає ретельної уваги до питань продуктивності для забезпечення стабільної роботи аудіосистеми без негативного впливу на загальну частоту кадрів та відгук гри, особливо на платформах з обмеженими ресурсами на кшталт мобільних пристроїв або консолей попереднього покоління. Ключові принципи оптимізації включають ефективне управління lifecycle звукових об'єктів, мінімізацію накладних витрат на взаємодію між Unity managed code та native FMOD Engine, розумне використання пам'яті та streaming ресурсів.

Пулінг Event Instances представляє паттерн об'єктного пулу для звукових подій, де замість створення нового event instance при кожному відтворенні часто використовуваного звуку, система підтримує пул попередньо створених та повторно використовуваних об'єктів, які переробляються між відтвореннями. Це значно зменшує навантаження на managed heap та garbage collector Unity, оскільки уникає постійної allocation та deallocation wrapper об'єктів навколо native FMOD handles, а також прискорює відтворення звуків шляхом усунення overhead створення нового instance, що може бути особливо помітним для звуків, які спрацьовують дуже часто на кшталт звуків пострілів в інтенсивних бойових сценах або системних UI sounds при швидкій навігації меню.

Віртуалізація голосів є автоматичним механізмом FMOD, який інтелектуально "віртуалізує" звукові джерела, що є далекими від слухача, занадто тихими для сприйняття, або мають низький пріоритет відносно інших активних звуків, припиняючи їх реальне відтворення та звільняючи process resources, але продовжуючи відстежувати їх логічний стан, позицію та параметри. Коли віртуалізований звук знову стає важливим через наближення слухача, збільшення пріоритету або зміну акустичних умов, він плавно та непомітно повертається до реального відтворення з коректним станом, як ніби

він відтворювався весь час. Це дозволяє підтримувати сотні або тисячі потенційних звукових джерел в великих відкритих світах, де реально відтворюються лише найрелевантніші звуки в безпосередній близькості або важливі для gameplay events незалежно від відстані.

Просторова оптимізація через Distance Culling автоматично зупиняє або віртуалізує Events, позиція яких знаходиться занадто далеко від будь-якого активного слухача, на основі налаштованого Max Distance параметра, який визначає межу слухової значимості для кожного типу звуку індивідуально. Occlusion optimization використовує спрощені розрахунки для звуків, що знаходяться за геометричними перешкодами, оскільки детальна просторова обробка менш критична для сильно приглушених джерел. Update rate reduction дозволяє зменшити частоту оновлення просторових параметрів для далеких джерел, оскільки зміни їх позиції менш помітні на відстані.

Асинхронне завантаження банків є критичною оптимізацією для уникнення заморожування основного потоку гри під час завантаження великих обсягів звукових даних, особливо на платформах з повільними накопичувачами. Асинхронні операції виконуються в окремих worker threads, дозволяючи грі продовжувати rendering, обробку вводу та виконання gameplay logic під час завантаження, з callback механізмом для сповіщення про завершення операції та готовність ресурсів до використання. Priority-based loading дозволяє завантажувати критичні банки з вищим пріоритетом раніше, забезпечуючи доступність найважливіших звуків першими при обмеженій пропускній здатності I/O системи.

Багатоплатформність та адаптація

FMOD автоматично адаптується до особливостей та обмежень різних апаратних платформ, забезпечуючи оптимальну роботу на широкому спектрі пристроїв від потужних настільних PC до обмежених мобільних пристроїв. При створенні build в Unity система автоматично включає відповідні platform-specific бінарні файли FMOD Engine, які компільовані та оптимізовані для конкретної цільової архітектури процесора та операційної системи,

забезпечуючи максимальну продуктивність нативного коду на кожній платформі. Система підтримує широкий спектр налаштувань, які можуть бути сконфігуровані індивідуально для кожної платформи без змін в єдиній кодовій базі: налаштування компресії аудіоданих від lossless форматів для платформ з достатніми ресурсами до агресивної компресії для мобільних пристроїв, sample rate від студійних 48kHz для консолей до зниженого 22kHz для low-end мобільних пристроїв, кількість одночасних каналів від 128+ для PC до обмежених 32 для слабких платформ, розмір буферів для балансу між латентністю та стабільністю на різному залізі.

Platform-specific encoding налаштовується безпосередньо в FMOD Studio через platform override система, де можна вказати індивідуальні параметри компресії та формату для кожної підтримуваної платформи: Vorbis з високим bitrate для PC та консолей для балансу між якістю та розміром, FADPCM для мобільних пристроїв для мінімального CPU overhead при декодуванні, platform-native formats на кшталт ATRAC9 для PlayStation або XMA для Xbox для максимальної інтеграції з апаратними декодерами, різні рівні агресивності компресії залежно від доступних ресурсів. Automatic platform detection під час build процесу обирає правильні audio assets та налаштування без необхідності ручної конфігурації для кожного build target.

Hardware-specific optimizations використовують унікальні можливості різних платформ: апаратні аудіопроцесори на консолях для offloading DSP обробки з основного CPU, специфічні SIMD інструкції процесорів для векторизації обчислень, низькорівневі audio APIs на кшталт WASAPI Exclusive Mode на Windows або Core Audio на macOS для мінімальної латентності. Memory management strategies адаптуються до обмежень платформи: агресивне використання streaming на консолях з швидкими SSD, більша залежність на in-memory банки на PC з достатнім RAM, ретельна оптимізація використання пам'яті на мобільних пристроях через compressed formats та minimalistic sound sets.

Взаємодія з іншими системами Unity

FMOD інтегрується не лише зі звуковою системою Unity, але й з широким спектром інших підсистем рушія, забезпечуючи глибоку інтеграцію в загальний workflow розробки та синергію з іншими аспектами створення гри. Timeline Integration надає можливість розміщувати FMOD Events безпосередньо на Unity Timeline для точної синхронізації звуку з кінематографічними послідовностями, складними анімаціями, подійними тригерами та іншими timeline-based системами. FMOD Event Track представляє спеціалізований тип доріжки в Timeline Editor, а FMOD Event Playable Assets є playable clips, які можна розташовувати, розтягувати та редагувати вздовж часової шкали візуально, дозволяючи дизайнерам кат-сцен створювати складну звукову синхронізацію без програмування. Parameter automation на Timeline дозволяє створювати анімаційні криві для параметрів FMOD Events безпосередньо в Timeline, синхронізуючи динамічну модуляцію звуку з візуальною дією кадр за кадром.

Animator Integration дозволяє викликати FMOD Events з Animation Events в Unity Mecanim Animator System, що забезпечує кадрово-точну синхронізацію звуків кроків, ударів, змін зброї, озвучування та інших звукових подій з відповідними ключовими кадрами анімації персонажів або об'єктів. Це критично важливо для підтримання відчуття ваги, імпакту та правдоподібності дій персонажів, де навіть невеликі десинхронізації між візуальним ударом та його звуком можуть руйнувати імерсію та сприйняття реалістичності. State Machine Parameters можуть бути зв'язані з FMOD параметрами, дозволяючи звуку автоматично адаптуватися до поточного анімаційного стану персонажа: звуки дихання змінюються між idle, walk, run, sprint states, озвучування бойових рухів підбирається відповідно до типу атаки з animator state machine.

Physics Integration реалізує автоматичну генерацію звуків від фізичних взаємодій між об'єктами в Unity Physics System через систему callbacks та event triggers. OnCollisionEnter/Stay/Exit events можуть автоматично спрацьовувати відповідні FMOD Events з параметрами, що відображають характеристики зіткнення: сила удару визначає гучність та інтенсивність звуку через mapping

collision magnitude на impact parameters, матеріали об'єктів визначають тип звуку через PhysicMaterial to SoundType mapping, швидкість та маса об'єктів модулюють частотний баланс та тембр. Surface Type System дозволяє асоціювати різні типи поверхонь з різними наборами звуків: кроки по дереву, металу, каменю, склу, воді автоматично відтворюють правильні звуки на основі матеріалу поверхні під персонажем через raycast detection та material queries.

Particle System Integration дозволяє синхронізувати звукові події з particle systems для створення узгоджених аудіовізуальних ефектів: звуки вогню, вибухів, магічних ефектів можуть модулюватися відповідно до інтенсивності та lifecycle particle emitters. FMOD Events можуть бути прив'язані до sub-emitters або конкретних фаз particle lifecycle, створюючи складні багатошарові аудіовізуальні композиції.

Workflow та організація роботи

Інтеграція FMOD та Unity створює високоефективний та паралелізований workflow для багатодисциплінарної команди розробників, де кожна спеціалізація може працювати в своїй області компетенції з мінімальними залежностями та блокуючими взаємодіями. Звукові дизайнери працюють переважно в FMOD Studio як в своєму primary інструменті, створюючи Events з повною аудіальною логікою, налаштовуючи складні параметричні системи, дизайнячи мікс-структуру та ефектні ланцюги, композуючи адаптивну музику з всіма transition rules та layering логікою. Вони можуть повноцінно тестувати свої звуки безпосередньо в контексті працюючої гри через Live Update функціональність без залучення програмістів для кожної ітерації, що радикально прискорює creative процес та дозволяє швидко експериментувати з різними підходами до звукового дизайну конкретних ігрових ситуацій.

Програмісти інтегрують готові Events в ігрову логіку через Unity, фокусуючись на технічній стороні: визначаючи умови та тригери для запуску звукових подій, передаючи релевантні параметри з ігрового стану до

аудіосистеми через параметричні API, керуючи життєвим циклом складних звукових об'єктів через програмне управління event instances, оптимізуючи продуктивність через правильне використання пулінгу та ресурсного менеджменту. Вони використовують Event References для безпечного та надійного посилання на звукові події з compile-time перевіркою та IDE autocomplete, що мінімізує ймовірність помилок та прискорює написання інтеграційного коду. Чітке розділення відповідальності означає, що звукові дизайнери можуть радикально змінювати внутрішню реалізацію Event, його структуру, використовувані семпли та ефекти без необхідності змін в програмному коді, доки зберігається контракт interface у вигляді імені події та її параметрів.

Геймдизайнери та левел-дизайнери можуть безпосередньо працювати зі звуком на високому рівні абстракції, використовуючи готові Studio Event Emitter компоненти для додавання звукових джерел до об'єктів, розміщуючи trigger zones для акустичних областей, налаштовуючи sound occlusion geometry без написання коду або глибоких технічних знань про аудіосистему. Параметри Events можуть контролюватися через візуальний Inspector interface або через прості, доступні для non-programmers скрипти з мінімальною технічною складністю. Це демократизує роботу зі звуком, дозволяючи всій команді контрибютити в аудіальний досвід гри відповідно до своїх компетенцій та бачення.

Система контролю версій є критичною частиною collaborative workflow, і FMOD Studio проєкти спроектовані з урахуванням VCS best practices: всі файли проєкту зберігаються як text-based XML з читабельною структурою, що забезпечує ефективну роботу з Git, Perforce, SVN або іншими системами контролю версій, дозволяє використовувати diff tools для перегляду змін між версіями, підтримує merge operations для паралельної роботи декількох звукових дизайнерів над різними частинами проєкту, та створює зрозумілу історію змін для аудіту та rollback при необхідності. Unity автоматично виявляє зміни в FMOD проєкті через file system watchers та автоматично

ініціює перебудову банків при імпорті нових версій .fspro файлів, забезпечуючи синхронізацію між source FMOD контентом та built банками в Unity проєкті без ручних операцій.

Build pipeline integration автоматизує підготовку аудіоконтенту для різних платформ: FMOD Studio Command Line Tools можуть бути інтегровані в automated build systems для batch processing великих обсягів контенту, validation scripts перевіряють цілісність аудіопроєкту перед build для раннього виявлення проблем, platform-specific builds автоматично генерують оптимізовані версії банків для кожної target platform з відповідними налаштуваннями компресії та формату. Asset streaming configuration визначає, які банки повинні бути вбудовані безпосередньо в application bundle, а які можуть бути завантажені динамічно через content delivery networks або downloadable content packages для зменшення розміру початкового завантаження.

Переваги інтегрованого підходу

Комбінація Unity як гнучкого та доступного ігрового рушія загального призначення та FMOD як спеціалізованого професійного middleware для інтерактивного аудіо створює потужну та збалансовану екосистему для створення сучасного ігрового звуку з високою якістю та складністю, доступну для проєктів будь-якого масштабу від невеликих інді-ігор до масивних AAA продукцій. Розділення відповідальності між візуально-ігровою частиною в Unity та аудіальною логікою в FMOD дозволяє кожній команді працювати в своїй області експертизи максимально ефективно та незалежно: звуковий дизайн повністю відокремлений від ігрової логіки на рівні імплементації, що дозволяє звуковим дизайнерам створювати складні інтерактивні звукові системи, багатошарову адаптивну музику, параметрично-контрольовані ефекти та складні мікс-конфігурації без необхідності розуміння програмування, C#, Unity scripting API або внутрішньої архітектури гри, фокусуючись виключно на творчих та акустичних аспектах своєї роботи.

Ітеративний розвиток стає природним та ефективним завдяки Live Update функціональності та швидкому циклу внесення змін: звукові дизайнери можуть експериментувати зі звуком безпосередньо в запущеній грі з реальним геймплеєм, складністю звукової сцени, специфічними ігровими ситуаціями та контекстом, який важко або неможливо повністю відтворити в ізольованому середовищі FMOD Studio, миттєво чуючи результат своїх налаштувань та змін без будь-яких проміжних кроків на кшталт експорту, імпорту, перебудови або перезапуску. Це радикально скорочує час від творчої ідеї до її реалізації та тестування в реальних умовах, дозволяючи швидко ітерувати над звуковим дизайном, пробувати різні підходи та налаштування, та знаходити оптимальні рішення для кожної специфічної ігрової ситуації через практичне експериментування замість теоретичного планування.

Професійні інструменти студійного рівня стають доступними навіть для невеликих команд та інді-розробників: FMOD надає інструментарій, функціональність та можливості, які використовуються в AAA-студіях з багатомільйонними бюджетами та командами десятків звукових спеціалістів, але доступний безкоштовно для інді-проєктів під певним threshold revenue та за розумну ціну для комерційних продукцій, демократизуючи доступ до передових технологій інтерактивного аудіо. Це дозволяє малим студіям створювати звуковий досвід, який конкурує за якістю та складністю з найбільшими продукціями, маючи лише обмежені ресурси та персонал, оскільки інструменти автоматизують багато технічних аспектів та надають готові рішення для типових задач.

Масштабованість системи підтримує весь спектр проєктів від простих мобільних ігор з десятками звуків та мінімальною складністю до масивних відкритих світів з тисячами унікальних інтерактивних аудіоелементів, складними системами адаптивної музики з десятками інтерактивних треків, просторовою акустикою з детальною геометрією оклюзії, та багатошаровими параметричними системами з сотнями керованих параметрів. Архітектура FMOD та його інтеграція з Unity побудовані з урахуванням потреб різних

масштабів проєктів: можна почати з простих non-параметричних звуків та поступово нарощувати складність, додаючи параметричний контроль, адаптивну музику, просторову обробку за необхідності, або одразу будувати складну багаторівневу систему для амбітного проєкту з повним використанням всіх можливостей middleware.

Продуктивність FMOD Engine є результатом багаторічної оптимізації спеціально для real-time інтерактивного аудіо в іграх: низькорівневий native code написаний на C++ з урахуванням специфіки різних апаратних платформ забезпечує мінімальний overhead та максимальну швидкість обробки, використання SIMD інструкцій та векторизації для паралельної обробки множинних аудіопотоків одночасно, ефективні алгоритми мікшування та ефектів, оптимізовані для real-time обробки з жорсткими обмеженнями на латентність, інтелектуальне управління ресурсами через voice stealing та віртуалізацію дозволяє підтримувати складні звукові сцени з сотнями джерел без деградації продуктивності. Це забезпечує, що навіть складні аудіосистеми споживають лише малу частку доступного CPU budget, залишаючи основні ресурси для gameplay логіки, рендерингу, фізики та AI.

Кросплатформна підтримка означає write once, deploy everywhere підхід для аудіосистеми: одна і та ж FMOD проєкт та інтеграційний код в Unity працює на всіх підтримуваних платформах від Windows, macOS, Linux на desktop через PlayStation, Xbox, Nintendo Switch на консолях до iOS та Android на мобільних пристроях, з автоматичною адаптацією до специфіки кожної платформи без необхідності підтримувати окремі кодові бази або проєкти. Platform-specific оптимізації застосовуються автоматично, використовуючи унікальні можливості апаратури кожної платформи для максимальної продуктивності та якості, але зберігаючи єдиний source проєкт та консистентну поведінку по всіх платформах.

Майбутня підтримка та розвиток забезпечується активним розвитком FMOD компанією Firelight Technologies з регулярними оновленнями, новими features, підтримкою нових платформ та технологій: нові версії Unity

автоматично підтримуються через оновлення інтеграційного пакету, нові апаратні платформи додаються до списку підтримуваних регулярно, нові аудіотехнології на кшталт spatial audio, Dolby Atmos, object-based audio інтегруються в FMOD і стають доступними для всіх користувачів. Велика спільнота користувачів створює екосистему навчальних матеріалів, tutorials, example projects, та shared knowledge, що спрощує навчання та вирішення специфічних проблем через колективний досвід індустрії.

Таким чином, інтеграція FMOD Studio з Unity представляє собою сучасний індустріальний стандарт професійного звукового дизайну в інтерактивних медіа та іграх, що поєднує спеціалізовану потужність dedicated audio middleware з гнучкістю, доступністю та широкою екосистемою популярного ігрового рушія загального призначення. Ця комбінація технологій дозволяє реалізувати найамбітніші звукові концепції та творчі бачення, створюючи глибоке аудіальне занурення, емоційний резонанс та високий рівень інтерактивності звукового середовища, що є критично важливими факторами для сучасних ігрових проєктів, які конкурують за увагу гравців у насиченому ринку. Система забезпечує оптимальний баланс між потужністю професійних інструментів, доступністю для розробників різного рівня, та продуктивністю для real-time застосунків, дозволяючи створювати звукові досвіди високої якості незалежно від розміру команди, бюджету проєкту або цільової платформи. Розділення відповідальності між творчою та технічною сторонами через middleware підхід створює ефективний workflow, де кожен спеціаліст може працювати в своїй області максимально продуктивно, а результат їх роботи органічно інтегрується в єдиний цілісний продукт через добре спроектовані API та автоматизовані процеси. Практичне значення цієї інтеграції полягає в тому, що вона робить складний адаптивний звук не лише технічно можливим, але й практично досяжним та економічно виправданим для широкого спектру проєктів, від студентських робіт та інді-експериментів до комерційних релізів ААА масштабу. Інструменти, які раніше були доступні лише великим студіям з спеціалізованими audio

programming командами, тепер доступні будь-якому розробнику через зручні готові рішення, що демократизує якісний звуковий дизайн та підвищує загальний рівень аудіальної складності та виразності в індустрії ігор в цілому.

3.2 Розробка та інтеграція звукового дизайну з використанням FMOD Studio

Підготовка проєкту та налаштування робочого середовища

Для практичної демонстрації можливостей інтеграції FMOD Studio та Unity я обрав техно-демо "Viking Village URP" від Unity Technologies як базове середовище для реалізації адаптивної аудіосистеми [Додаток В]. Це рішення було обґрунтованим з декількох причин: проєкт вже містив повністю готову візуальну сцену з детальною геометрією, професійно налаштованим освітленням та великою кількістю готових 3D-об'єктів, текстур та матеріалів. Така готова база дозволила повністю зосередитись на технічних та творчих аспектах звукового дизайну без необхідності витрачати час на створення візуального контенту з нуля.

Оригінальна демо-сцена була призначена для статичного кінематографічного прольоту камери через локацію, реалізованого за допомогою системи Unity Timeline та Cinemachine. Оскільки моєю метою було створення інтерактивного досвіду з можливістю вільного дослідження простору, першим кроком стало повне видалення всієї кінематографічної логіки.

Замість статичної камери я додав новий ігровий об'єкт Player до сцени та оснастив його компонентом CharacterController, який забезпечує фізично коректне пересування по тривимірному середовищу з автоматичною підтримкою колізій, природною реакцією на гравітацію, можливістю підйому по сходах. Додатково я реалізував простий скрипт керування від першої особи, який обробляє ввід з клавіатури для руху та ввід з миші для обертання камери. Це дало можливість вільно досліджувати всю сцену та тестувати просторову поведінку звукових джерел з різних позицій.

Інтеграція FMOD у Unity проєкт здійснювалась через офіційний плагін FMOD for Unity Integration Package, який було завантажено з офіційного сайту Firelight Technologies та імпортовано через Unity Package Manager. Після успішного імпорту в меню Unity з'явились нові пункти FMOD, які надають доступ до налаштувань інтеграції. Через FMOD Settings я вказав шляхи до директорії мого FMOD Studio проєкту, де зберігаються файли .fspro та .bank, налаштував параметри автоматичної збірки банків при зміні FMOD проєкту, та сконфігурував platform-specific налаштування для Windows.

Структура FMOD Studio проєкту була організована з самого початку для зручної навігації. Я створив логічну ієрархію папок: Events розділені на категорії Ambience для фонових звуків, SFX для звукових ефектів, Footsteps для системи кроків. Audio Files організовані відповідно до типу: Recorded містить власні записи, Library містить ліцензовані звуки, Processed містить фінально оброблені версії. Mixer Groups структуровані ієрархічно: Master група на вершині, під нею Music, SFX, Ambience, а всередині SFX додаткові підгрупи для Footsteps, Environment, UI.

Створення та обробка вихідного звукового матеріалу

Перед початком роботи в FMOD Studio необхідно було підготувати якісний вихідний звуковий матеріал. Особливу увагу я приділив створенню реалістичних звуків кроків персонажа, оскільки це один з найбільш частих звуків у грі. Замість готових бібліотечних семплів я вирішив створити власні унікальні записи.

Для запису я використав портативний цифровий рекордер Zoom H1n [Додаток Г], який забезпечує високу якість запису у польових умовах завдяки якісним конденсаторним мікрофонам у стерео конфігурації, низькому рівню власних шумів та широкій частотній характеристиці. Записи проводились у форматі WAV 24-bit/48kHz для максимальної якості без втрат від компресії. Процес запису кроків (Foley) здійснювався з дотриманням професійних методик польового та студійного звукозапису, описаних Р. В'єрсом [45], що

передбачало створення кількох варіацій для кожної поверхні задля уникнення ефекту повторюваності.

Було записано декілька серій кроків по різних типах поверхонь, релевантних для середньовічного поселення. Трава різної густоти створює характерний шурхіт з м'яким хрустом волокон. Дерев'яні дошки настилів мають резонансне звучання з характерним відлунням порожнин під ними, старі дошки скриплять під вагою. Гравій та утрамбована земля дають характерний хруст дрібних камінчиків. Кожна поверхня була записана з мінімум десятьма різними дублями, з дотриманням професійних методик польового та студійного звукозапису, описаних Р. В'єрсом [45], що передбачало створення кількох варіацій для кожної поверхні задля уникнення ефекту повторюваності.

Також я варіював інтенсивність кроків від дуже легких до важких впевнених кроків з повною вагою тіла. Окремо записав звуки приземлення після стрибка, які мають більш ударний характер з акцентованим низькочастотним компонентом.

Подальша обробка записаного матеріалу здійснювалась в PreSonus Studio One 7 Professional. Кожен окремий семпл кроку було імпортовано в DAW та розміщено на окремій доріжці для індивідуальної обробки.

Перший етап обробки – очищення записів від небажаних артефактів та шумів. Я використав спектральний аналізатор для виявлення проблемних частот [Додаток Д]. Я використав спектральний аналізатор для виявлення проблемних частот. Керуючись принципами частотного розділення Д. Гібсона [19], до всіх записів було застосовано High-pass фільтр (80-100 Гц) для усунення низькочастотного конфлікту, а також посилено діапазон 2-5 кГц для кращої читабельності текстур у загальному міксі. Для деяких записів з дерев'яними поверхнями я зменшив резонансні піки в середніх частотах близько 400-600 Гц через parametric EQ.

Додавання чіткості здійснювалось через помірне посилення верхніх середніх частот у діапазоні 2-5 кГц приблизно на 2-3 dB. Саме в цьому

діапазоні знаходяться основні характеристики текстури поверхонь: шурхіт трави, хруст гравію, деталі дерева. Для звуків по траві я також додав легке посилення близько 8-10 кГц для додаткової повітряності.

Компресія застосовувалась для вирівнювання динамічного діапазону. Я використав налаштування з ratio 3:1, attack 5-10 ms для збереження початкового транзієнту удару, release 50-80 ms для природного затухання, та threshold налаштований індивідуально для кожного типу поверхні щоб досягти приблизно 3-4 dB gain reduction. Це забезпечило стабільну гучність кожного кроку незалежно від варіативності первинного запису.

Фінальна стадія включала нормалізацію всіх семплів до -6 dB peak level для залишення headroom під можливі додаткові ефекти в FMOD, trimming silence з початку та кінця кожного файлу, та fade-in/fade-out по 5-10 мілісекунд для уникнення clicks. Оброблені файли були експортовані в форматі WAV 16-bit/48kHz.

Імплементація атмосферних звуків в Unity

Після підготовки базових звукових ресурсів я почав роботу над створенням атмосферного звукового шару локації. Створення переконливого звукового середовища вимагало комплексного підходу, де множина різнотипних звукових елементів працюють разом.

Особливу увагу я приділив створенню ілюзії життя всередині закритих будинків поселення. За зачиненими дверима мало відчуватись святкування, розмови мешканців, брязкання посуду, сміх та приглушена музика.

В FMOD Studio я створив декілька варіантів звукових подій для різних типів будинків. Подія "House_Feast" для великого святкового залу містить чотири основні шари звуку. Базовий шар представляє групову розмову з накладеними записами голосів, який циклюється безперервно через Loop Region з налаштованим crossfade 500 мс.

Другий шар містить звуки переміщення меблів, брязкання посуду, стукання кубків. Ці елементи організовані через Multi Sound модуль з 8

різними варіантами в Random mode. Scatterer Instrument налаштований для випадкового спрацювання з інтервалом від 3 до 8 секунд.

Третій шар - сміх та окремі вигуки. Я використав Multi Sound з 6 варіантами різного сміху з випадковою модуляцією pitch на ±5%. Ці елементи спрацьовують рідше через Scatterer з інтервалом 8-15 секунд.

Четвертий шар - м'яка фонова музика з середньовічними інструментами. Я обробив її через low-pass фільтр з частотою зрізу 4 кГц для ефекту приглушеності крізь стіни, додав легку реверберацію, та знизив загальну гучність до -18 dB.

Для менших житлових будинків я створив подію "House_Living" з більш спокійним характером: тихі розмови 2-3 людей, періодичні звуки побутової активності. Для ремісничих майстерень створив "House_Workshop" зі специфічними звуками: удари молотка для коваля, свист мехів, скрип обробки деревини для столяра.

Імплементація цих звуків в Unity вимагала ретельного розміщення джерел. Я створював порожні GameObject всередині геометрії кожного будинку, часто в центрі найбільшого приміщення або біля каміна. До кожного об'єкта додавався компонент FMOD Studio Event Emitter з посиланням на відповідну звукову подію.

Критично важливою була правильна налаштування 3D атрибутів Emitter для оклюзії. В налаштуваннях Event я встановив Min Distance на 2 метри та Max Distance на 25 метрів. Attenuation curve налаштував на Inverse Tapered для природного загасання звуку [Додаток Є].

Налаштування системи оклюзії (Occlusion) спиралося на фізичні властивості звукоізоляції перегородок [1]: оскільки високочастотні хвилі поглинаються та відбиваються перешкодами значно сильніше за низькочастотні, для імітації звуку "за стіною" було застосовано Low-Pass фільтр із частотою зрізу 1-2 кГц. Це створює характерне приглушене звучання. При наближенні до дверей ефект оклюзії динамічно зменшується.

Додатково FMOD автоматично застосовує ефект Доплера при швидкому русі персонажа. При пробіганні повз будинок відбувається невелике підвищення частоти звуку при наближенні та зниження при віддаленні.

Реалізація системи точкових звукових джерел через Prefab

Наступним етапом стала імплементація звуків смолоскипів, які широко представлені по всій сцені. Viking Village містить більше тридцяти смолоскипів, розміщених біля входів, вздовж доріжок, на стінах веж.

Для кожного смолоскипа я створив складну звукову подію "Torch" в FMOD Studio, яка складається з двох основних компонентів.

Базовий шар "Torch_Loop" представляє безперервне горіння полум'я - постійний звук руху повітря, коливання полум'я, тихий гул горіння. Я використав запис справжнього великого полум'я, записаний у близькій позиції. Оригінальний запис я обробив для створення ідеального безшовного циклу: знайшов точку з найбільш стабільним звучанням, обрізав до 4 секунд, застосував crossfade 200 мілісекунд між кінцем та початком.

Loop Region налаштований для безкінечного відтворення з Sustain Point. До цього шару я додав легку автоматизацію гучності через LFO modulator з частотою 0.3 Hz та глибиною ±1.5 dB, що створює ледь помітні коливання інтенсивності горіння.

Другий шар "Torch_Crackle" додає варіативність через випадкові звуки потріскування. Він організований через Scatterer Instrument з Multi Sound модулем, що містить п'ять різних варіантів. Spawn Interval встановлений від 2 до 6 секунд, Volume Range від -3 до +1 dB, Pitch Range від -8% до +8%.

Для просторового позиціонування події Torch я налаштував 3D Attenuation з Min Distance 1 метр та Max Distance 12 метрів як реалістична межа чутності тихого полум'я.

Технічно імплементувати більше тридцяти смолоскипів індивідуально було б неефективно. Тому я використав систему Prefab Unity. Спочатку створив один еталонний GameObject "Torch_Audio_Source" - порожній об'єкт

без візуальної геометрії. До нього додав компонент FMOD Studio Event Emitter з посиланням на подію "event:/SFX/Environment/Torch" [Додаток Е].

Trigger налаштування встановлені на Object Start для автоматичного запуску при активації. Stop Event на Object Destroy для зупинки при знищенні. Allow Fadeout увімкнений для плавного затихання.

Після повного налаштування я перетягнув об'єкт з Hierarchy у вікно Project в папку Prefabs. Unity автоматично створила Prefab asset. Тепер для додавання звуку до кожного візуального смолоскипа я просто перетягував Prefab з Project на позицію смолоскипа. Весь процес додавання до тридцяти смолоскипів зайняв менше десяти хвилин.

Найбільша перевага Prefab проявилась при коригуванні: коли виникла необхідність змінити Max Distance з 12 на 8 метрів, я просто відкрив оригінальний Prefab, змінив одне значення, та Unity автоматично поширила це оновлення на всі екземпляри миттєво.

Інтеграція динамічних атмосферних елементів

Для додаткових атмосферних деталей я імплементував систему випадкових скрипів дахів та дерев'яних конструкцій. Старі вікінгські будинки природно реагують на зміни вологості, температури, вітер, створюючи періодичні скрипи.

В FMOD Studio було створено подію "RoofCreak" з Multi Sound модулем, що містить сім різних варіантів: від коротких легких скрипів 0.5 секунди до довгих протяжних до 2 секунд, від високочастотних до низьких. Playlist mode встановлений на Shuffle для гарантії що всі варіанти будуть відтворені по одному разу перед повторенням. Кожен звук має випадкову модуляцію pitch ±10%.

Замість безперервного відтворення, ця подія викликається періодично. Я розмістив порожні GameObject під дахами, біля балок, на вежах - всього близько п'ятнадцяти позицій. Кожен Emitter налаштований на випадкове спрацювання через допоміжний скрипт RandomEventTrigger.

Скрипт працює просто: кожні випадкові 15-45 секунд він викликає метод Play на Event Emitter, що запускає одне відтворення події з випадковим семплом. Така нерегулярність створює відчуття живої дерев'яної архітектури, яка постійно реагує на навколишнє середовище.

3.3 Композиція та інтерактивна імплементація звукового супроводу у FMOD Studio

Розробка адаптивної системи звуків персонажа в FMOD Studio

Створення переконливих звуків кроків персонажа є критичним для занурення гравця. Звуки кроків - це найчастіший постійний елемент, з яким взаємодіє гравець протягом всього досвіду.

Оброблені аудіофайли я імпортував в FMOD Studio з чіткою організацією. В папці Audio Files я створив підпапки за типами: Grass містить вісім варіантів кроків по траві, Wood містить десять варіантів по дерев'яних дошках, Gravel містить дев'ять варіантів по гравію.

Центральна звукова подія "event:/Player/Footsteps" була спроектована як складна інтерактивна структура з можливістю динамічної адаптації. Ключова концепція - використання параметричного контролю через FMOD Game Parameter для перемикання між наборами звуків.

В розділі Parameters я створив новий Game Parameter "Terrain" з типом Continuous. Діапазон встановлений від 0.0 до 2.0 [Додаток Ж], де кожне ціле значення відповідає типу поверхні: 0.0 - гравій, 1.0 - дерев'яні дошки, 2.0 - резерв для майбутнього каменю. Initial Value 0.0, Seek Speed 5.0 units per second для швидкого але не миттєвого переходу.

На Timeline події Footsteps я створив складну багатошарову структуру з чотирма паралельними треками.

Перший трек "Gravel_Main" містить Multi Instrument з дев'ятьма варіантами звуків по гравію в Random mode з minimum 2 retriggerings. Гучність контролюється через Automation Curve: при Terrain 0.0 гучність максимальна

0 dB, при 0.5 знижується до -12 dB, при 1.0 та вище повністю приглушений -∞ dB.

Другий трек "Wood_Main" містить Multi Instrument з десятьма варіантами по дереву. Automation Curve має протилежну форму: при 0.0 приглушений -∞ dB, при 0.5 на рівні -12 dB, при 1.0 досягає максимуму 0 dB, при 2.0 знову -∞ dB. Це створює "пік" на значенні 1.0.

Третій та четвертий треки "Gravel_Sub" та "Wood_Sub" - додаткові шари для багатшого звучання. Gravel_Sub містить тихіші варіанти землі та пилу. Wood_Sub містить тихі скрипи дощок та резонанси порожнин.

Особлива увага дерев'яним поверхням через багатошарову композицію. Wood_Main містить основний резонансний звук удару по дошках. Але Wood_Sub додає тихий шар звуків гравію на рівні -18 dB. Чому? Тому що в реальності між дошками настилу є щілини, через які видно землю. Коли нога ступає на дошку, вага трохи стискає грунт під настилом, створюючи додатковий тихий звук. Цей підхід до layering створює значно більш правдоподібне звучання.

Для додаткової варіативності я додав випадкову модуляцію. Pitch Randomization ±3% для кожного відтворення створює ледь помітну тональну різницю. Volume Randomization ±1.5 dB додає природну мінливість інтенсивності кроків.

Програмна реалізація адаптивних кроків через Unity скриптинг

Технічна реалізація вимагала створення скрипта PlayerFootsteps.cs. Цей скрипт служить мостом між фізичним світом Unity та логікою FMOD.

Скрипт прикріплюється до GameObject персонажа з CharacterController. Структура включає кілька ключових змінних: footstepEvent типу EventReference містить посилання на FMOD подію, surfaceParameterName містить ім'я параметра "Terrain", speedThreshold 0.1f визначає мінімальну швидкість для кроків, stepInterval 0.5f визначає базовий інтервал між кроками.

Приватні змінні: characterController для доступу до даних руху, stepTimer для відстеження часу з останнього кроку, currentSurfaceValue для значення параметра Terrain, lastSurfaceTag для оптимізації.

Метод Start отримує посилання на CharacterController через GetComponent для подальшого доступу до velocity та isGrounded без повторних викликів.

Центральна функція CheckSurface виконується кожен кадр з Update і визначає тип поверхні через Physics.Raycast. Raycast - техніка трасування променя: з певної точки в заданому напрямку випускається невидимий промінь, і фізичний рушій перевіряє чи перетинається він з колайдерами.

У моїй реалізації промінь випускається з позиції трохи нижче центру CharacterController через transform.position + Vector3.down * 0.1f, вертикально вниз, на відстань characterController.height * 0.5f + 0.2f для надійної детекції підлоги.

Якщо промінь влучає в колайдер, Physics.Raycast повертає true та заповнює RaycastHit інформацією: hit.point містить координати влучання, hit.collider дає доступ до GameObject.

З GameObject я зчитую Unity Tag через hit.collider.tag. Для системи кроків я створив кастомні теги через Edit > Project Settings > Tags and Layers: "Grass" для трави, "Wood" для дерев'яних настилів, "Stone" для каменю, "Default" для загальних поверхонь.

Різні області локації я промаркував відповідними тегами. Всі plane об'єкти трав'яного покриття отримали Grass. Містки та настили - Wood. Кам'яні сходи - Stone. Цей процес зайняв близько години, але він критично важливий.

Після отримання тегу скрипт використовує switch statement для перетворення текстової мітки в числове значення. case "Grass" встановлює 0.0f, case "Wood" встановлює 1.0f, case "Stone" - 2.0f, default - 0.0f.

Для оптимізації я додав перевірку: якщо виявлений тег ідентичний попередньому в lastSurfaceTag, значення не змінюється. Це зменшує непотрібні API виклики при ходьбі по одній поверхні.

Логіка відтворення кроків в Update перевіряє три умови. Перша - characterController.isGrounded чи персонаж на землі, не в повітрі. Друга - characterController.velocity.magnitude > speedThreshold перевіряє швидкість руху. speedThreshold 0.1f створює мертву зону для уникнення звуків при мікрорухах.

Якщо умови виконуються, активується система таймера. stepTimer акумулює час через додавання Time.deltaTime кожен кадр. Time.deltaTime містить час що пройшов з останнього кадру.

Коли stepTimer >= stepInterval, викликається PlayFootstepSound і таймер скидається до нуля. Я також додав адаптивний елемент: stepInterval динамічно змінюється залежно від швидкості. При повільній ході interval 0.5f, при швидкому бігу зменшується до 0.35f через stepInterval = Mathf.Lerp(0.5f, 0.35f, velocity.magnitude / 6f). Це створює природну різницю в ритмі.

Метод PlayFootstepSound інкапсулює логіку створення екземпляра FMOD. Спочатку створюється екземпляр через RuntimeManager.CreateInstance(footstepEvent). CreateInstance повертає EventInstance - незалежну копію з власним станом. Це дозволяє мати декілька активних екземплярів без конфліктів.

Далі через set3DAttributes прив'язуються просторові атрибути. ATTRIBUTES_3D заповнюється поточною позицією персонажа через RuntimeUtils.To3DAttributes(transform.position), векторами forward та up, та velocity для ефекту Доплера. Це забезпечує що кожен крок звучить точно з позиції ніг.

Критично важливий момент - eventInstance.setParameterByName(surfaceParameterName, currentSurfaceValue). Ця команда передає інформацію про тип поверхні з Unity

в FMOD, де automation curves автоматично активують відповідний набір звуків.

Така реалізація відповідає кібернетичному підходу до геймдизайну [34] звук виступає інформаційним каналом, який повідомляє гравцеві про зміну стану системи (зміна поверхні під ногами), замикаючи контур зворотного зв'язку.

Після налаштування екземпляр запускається методом start(). Одразу викликається release(), який може здаватись парадоксальним але критично важливий. release() НЕ зупиняє відтворення. Він лише повідомляє FMOD що після природного завершення відтворення система може автоматично звільнити пам'ять. Це pattern fire-and-forget, ідеальний для коротких одноразових звуків. Без release() кожен екземпляр залишався б в пам'яті назавжди, призводячи до витоку пам'яті.

Просторове аудіо та створення єдиного акустичного простору

Після імплементації всіх індивідуальних елементів виникла необхідність об'єднати їх в єдине цілісне середовище. Без цього звуки звучали б ізольовано та штучно.

Для створення єдиного акустичного простору я використав систему реверберації в FMOD Studio. Реверберація симулює природне відлуння від архітектурних елементів, відбиття від стін, землі, дерев, що створює "обволікання" звуку простором.

В розділі Mixer я створив спеціальну групу "Reverb Bus" окремо від основних Music, SFX, Ambience. На цю шину додав ефект Convolution Reverb з імпульсною характеристикою середнього розміру відкритого двору оточеного дерев'яними будівлями з бібліотеки Open AIR. Альтернативно можна використати алгоритмічну FMOD Studio Reverb.

Параметри реверберації: Room Size на Medium (30-40 метрів) для простору між будинками, Decay Time 1.2 секунди щоб відлуння не було надто довгим, High Frequency Damping 60% для поглинання високих частот

дерев'яними поверхнями, Wet Level на -12 dB відносно dry сигналу для помірного ефекту.

Для маршрутизації звуків на реверберативну шину я використав механізм send buses в FMOD [Додаток З]. Send bus дозволяє відправляти копію сигналу на додаткову шину паралельно з основним. Оригінальний dry сигнал йде на master output без змін, але його копія одночасно надсилається на Reverb Bus де обробляється та змішується назад.

Я відкрив кожну ключову подію - Footsteps, Torch, всі House амбієнти, RoofCreak - та на їх Master Track додав Send до Reverb Bus. Рівень send налаштував індивідуально для кожного типу.

Кроки персонажа мають send level -20 dB для збереження чіткості звуку близько до слухача. Смолоскипи мають -15 dB як компроміс. Амбієнти будинків мають -10 dB оскільки вони вже приглушені оклюзією і додаткова реверберація посилює відчуття простору. Скрипи дахів мають найбільший send -8 dB як далекі атмосферні елементи.

Така диференціація створює природну звукову перспективу: близькі важливі звуки залишаються чіткими, тоді як далекі елементи сильніше інтегровані в простір. Результат цієї системи надзвичайно відчутний: всі елементи тепер звучать як єдине середовище. При пересуванні можна чути як звуки смолоскипів відлунюють від стін, як кроки мають легке відлуння, як далекі скрипи зливаються з загальною атмосферою.

Організація контенту через групову систему

Ефективна організація через систему банків FMOD критично важлива для оптимізації пам'яті та часу завантаження. Банки дозволяють логічно групувати контент та завантажувати його вибірково.

В моєму проєкті я структурував контент в три основні банки. Master Bank завантажується автоматично при старті і містить всю структуру мікшера з усіма групами Music, SFX, Ambience, Reverb Bus, всі визначення VCA, та глобальні параметри на кшталт Terrain. Master Bank зазвичай невеликий оскільки містить лише метадані без фактичних аудіосемплів.

Банк Environment містить всі інтерактивні звуки що залежать від дій гравця. Сюди входять Footsteps з усіма варіантами кроків, Torch з звуками полум'я, RoofCreak, та потенційні майбутні звуки взаємодій. Для оптимізації використання оперативної пам'яті (RAM) налаштування банків було здійснено згідно з рекомендаціями розробників [18]: короткі звуки (SFX) завантажуються у режимі Decompress on Load, тоді як довгі атмосферні треки використовують режим Streaming для читання з диска [Додаток К].

Це забезпечує абсолютно нульову латентність при першому відтворенні - критично для звуків кроків які повинні спрацьовувати синхронно з рухом без затримки. Недолік - більше споживання RAM, але для відносно коротких ефектів це прийнятний trade-off. Банк Ambs містить всі фонові атмосферні звуки. Сюди входять всі варіанти House подій, Wind якщо буде додано, Ocean, та інші довготривалі циклічні елементи. Ambs Bank налаштований як streaming bank, що означає великі аудіофайли не завантажуються повністю в пам'ять а читаються поступово з диска під час відтворення.

Streaming критично важливий для великих амбієнтів оскільки завантаження всіх варіантів House амбієнтів повністю могло б займати 50-80 мегабайт RAM. З streaming ці самі звуки займають лише 2-3 мегабайти буферів, тоді як основні дані читаються з диска в реальному часі. Вимога - достатня пропускна здатність диска, але на сучасних SSD це не проблема.

В Unity було налаштоване автоматичне завантаження банків. Master Bank завантажується автоматично при ініціалізації FMOD через налаштування в FMOD Settings. Environment Bank я завантажую в Start методі головного менеджера сцени через RuntimeManager.LoadBank("Environment"). Ambs Bank завантажується аналогічно одразу після Environment для забезпечення доступності всіх звуків на момент першого кадру геймплею.

Реалізація динамічного вітру залежно від висоти

Система динамічного вітру представляє складнішу інтерактивну систему, де інтенсивність звуку адаптується до висоти персонажа над базовим рівнем землі. Це створює відчуття посилення вітру при підйомі на вежі або

гірські схили.

В FMOD Studio було створено багатошарову подію Wind з трьома паралельними треками різної інтенсивності. Базовий шар містить м'який низькоінтенсивний вітер, характерний для захищеного поселення на рівні землі - легкий рух повітря, тихий свист через щілини. Середній шар додає більш виражену поривистість для помірних висот - посилений рух повітря, періодичні порив. Верхній шар включає агресивне завивання та свист для максимальних висот, де повітряні потоки не обмежуються перешкодами.

Для керування цими шарами створив глобальний параметр Height з діапазоном від 0 до 20 одиниць, що відповідає вертикальному діапазону локації. Automation curves для гучності кожного шару налаштовані так: при Height близькому до нуля активний лише базовий шар з повною гучністю 0 dB, середній та верхній повністю приглушені -∞ dB.

При поступовому зростанні параметра базовий шар поступово затихає, середній з'являється та досягає максимуму приблизно на середині діапазону близько Height 10, після чого також затихає, поступаючись верхньому шару який домінує при значеннях близьких до 20. Такий підхід до layering з crossfade переходами створює плавну еволюцію характеру вітру без різких стрибків.

Технічна реалізація вимагає скрипта на персонажі, який постійно моніторить вертикальну координату. Я створив простий скрипт HeightParameterController.cs, який прикріплюється до Player об'єкта. В методі Start скрипт визначає базову висоту локації minHeight як найнижчу точку землі та maxHeight як найвищу доступну точку для гравця.

В Update кожен кадр скрипт зчитує transform.position.y, нормалізує її у діапазон 0-20 через просте обчислення: normalizedHeight = Mathf.InverseLerp(minHeight, maxHeight, currentHeight) * 20f, та передає значення до FMOD через RuntimeManager.StudioSystem.setParameterByName("Height", normalizedHeight). Це забезпечує миттєву реакцію звуку на зміни висоти.

Система звучання динамічного океану з розрахунком відстані

Система динамічного океану - найскладніша частина проєкту. Океан займає велику площу вздовж краю локації, і його гучність повинна змінюватись залежно від відстані до берега.

В FMOD Studio я створив багату композицію "Ocean Event" з декількома шарами. Базовий шар - постійний шум прибою з плесканням хвиль об берег як безперервний текстурний фон. Середній шар додає окремі великі хвилі з характерним набіганням води через Scatterer модуль з інтервалом 4-8 секунд для природної нерегулярності. Верхній шар - морський вітер, який специфічно відрізняється від континентального своєю вологістю. Додатковий шар може включати випадкові крики чайок через Scatterer з рідкими інтервалами 15-30 секунд.

Для керування інтенсивністю створено глобальний параметр DistanceOcean з діапазоном від 0 до] одиниць. Нуль відповідає близькості до води або знаходженню всередині акустичної зони, сорок - максимальній відстані за якою океан не чутний. Automation curves налаштовані так: при DistanceOcean рівному нулю всі елементи звучат з максимальною інтенсивністю 0 dB, створюючи потужне звучання на березі. При збільшенні значення загальна гучність поступово зменшується відповідно до реалістичної кривої загасання, яку я налаштував як комбінацію лінійного та експоненційного загасання для художнього контролю. Окремі шари мають різні криві: високочастотні елементи плескоту затихають швидше через більше атмосферне поглинання, тоді як низькочастотний гул залишається чутним довше. Додатково застосовується progressive low-pass фільтрація через автоматизацію Cutoff Frequency фільтру, прив'язану до параметра, яка посилюється при збільшенні відстані, прибираючи високі частоти.

Технічна реалізація є найскладнішою і вимагала створення спеціалізованого скрипта FMODDistanceParameterZone.cs, який вирішує задачу обчислення відстані від персонажа до найближчої точки великої тригерної зони океану.

Після імплементації всіх систем я виконав оптимізацію продуктивності. Завдяки функції **Live Update** було встановлено безпосереднє з'єднання між FMOD Studio та запущеним ігровим прототипом [Додаток Л] дозволило в реальному часі моніторити метрики під час інтенсивних сценаріїв.

Виявилось, що в центрі поселення з багатьма смолоскипами, активними амбієнтами та звуками кроків система досягала 40-50 одночасних голосів, що залишалось в комфортних межах. Налаштування пріоритетів для різних Events дозволило FMOD автоматично віртуалізувати найменш важливі звуки при досягненні ліміту.

Організація банків виявилась ефективною: критичні звуки завантажувались в RAM повністю, великі амбієнти використовували streaming. Загальне використання аудіопам'яті склало 15-20 мегабайт, що прийнятно навіть для обмежених платформ.

Тестування показало високу ефективність адаптивного підходу: звук органічно реагував на всі дії персонажа. Система кроків коректно перемикалась між поверхнями без помітних артефактів, океан плавно посилювався при наближенні, вітер природно змінювався при підйомі, а реверберація об'єднувала все в єдиний простір.

Реалізація адаптивного звукового дизайну в Unity з FMOD Studio на базі Viking Village продемонструвала високу ефективність middleware підходу для створення динамічного та імерсивного звукового середовища. Система параметричного керування через Terrain, Height та DistanceOcean дозволила звуку інтелектуально реагувати на контекст та дії гравця, створюючи відчуття живого світу, який відповідає на присутність персонажа природним чином.

Використання Prefab системи для організації повторюваних звукових елементів значно спростило робочий процес та забезпечило консистентність налаштувань по всій сцені. Технологія Multi Instrument в FMOD забезпечила необхідну варіативність без значного збільшення обсягу роботи, дозволяючи створювати живі та непередбачувані звукові текстури з обмеженого набору семплів. Просторова обробка через Spatializer та система оклюзії створили

реалістичне тривимірне звукове поле, де позиція та відстань джерел природно сприймаються слухом гравця.

Практична робота підтвердила, що комбінація Unity та FMOD надає всі необхідні інструменти для реалізації складних адаптивних аудіосистем професійного рівня, доступних навіть для невеликих команд та індивідуальних розробників. Розділення відповідальності між творчою роботою в FMOD Studio та технічною інтеграцією в Unity створює ефективний workflow, де звуковий дизайнер може фокусуватися на художніх аспектах, тоді як програміст забезпечує технічну основу для реалізації творчого бачення через прості та зрозумілі скриптові рішення.

3.4 Написання та імплементація музичного супроводу

У рамках виконання практичної частини магістерської роботи було здійснено розробку та інтеграцію музичного супроводу до ігрового прототипу. Основною метою цього етапу стало створення тематичної музичної композиції, яка б гармонійно поєднувалася зі звуковим середовищем та сприяла формуванню цілісного аудіовізуального простору гри.

Процес інструментовки та добору тембрів здійснювався відповідно до засад комп'ютерного аранжування, викладених у посібнику В. Грищенка [5]. Для створення автентичного забарвлення використовувалися віртуальні інструменти на основі семплера Native Instruments Kontakt..

Основою оркестрового звучання стали бібліотеки **Spitfire Audio**, зокрема набори струнних та духових інструментів, що відзначаються високим рівнем деталізації, природними артикуляціями та багатошаровою динамікою. Завдяки цьому вдалося досягти живого, динамічного звучання оркестру, яке зберігає автентичний характер і водночас відповідає сучасним вимогам до якості аудіопродукції у відеоіграх [Додаток М].

Особливу увагу було приділено створенню простору звучання. За допомогою реверберaційних ефектів було змодельовано акустичне середовище невеликого кам’яного приміщення, що асоціюється із

середньовічними залами або тавернами. Додатково застосовано легке фільтрування високих частот, яке надає композиції м'якого, «вікового» відтінку, що підсилює атмосферу старовини.

На етапі інтеграції музики у гру було використано звуковий рушій **FMOD Studio**, який забезпечує можливість динамічного керування аудіоподіями у зв'язку з ігровими сценаріями. У межах FMOD створено новий подієвий об'єкт (*event*) під назвою **Music**, який розміщено в окремому банку **Music**. На **Timeline** додано музичний фрагмент із активованою функцією **Loop** [Додаток Н], що забезпечує безшовне циклічне відтворення теми без чутних переходів між повтореннями. Такий підхід дозволяє підтримувати стабільну музичну атмосферу без відчуття повторюваності чи монотонності.

Подальша інтеграція здійснювалася у рушії **Unity**. Для цього було створено порожній об'єкт, якому присвоєно назву **Music**. До нього додано компонент **FMOD Studio Event Emitter**, який відповідає за ініціалізацію та керування аудіоподією в ігровому просторі. У параметрах **Event Play Trigger** обрано опцію **Object Start**, що забезпечує автоматичний запуск музики під час завантаження сцени. У полі **Event Stop Trigger** встановлено значення **None**, завдяки чому відтворення відбувається безперервно протягом усього часу перебування гравця на локації.

Для досягнення збалансованого звучання музичного фону та звукових ефектів середовища в **FMOD Mixer** було реалізовано окремі маршрути (*bus routing*) для музики, атмосферних і дієгетичних звуків. Це дозволило забезпечити правильне співвідношення рівнів гучності, уникнувши конфлікту між музикою та ефектами навколишнього середовища.

У результаті проведених робіт було реалізовано повноцінну інтеграцію музичного супроводу, що гармонійно поєднується з усіма аудіоелементами ігрового проєкту. Створена композиція виконує не лише декоративну, а й емоційно-наративну функцію, формуючи відчуття занурення гравця у середньовічний світ, посилюючи атмосферу гри та підкреслюючи її художню стилістику.

ВИСНОВКИ

Відповідно до поставленої мети та визначених задач результати дослідження дозволяють зробити наступні **висновки**:

1. Під час виконання роботи була опрацьована джерельна база дослідження (всього 48 джерел). Зокрема, проаналізовано праці зарубіжних дослідників з історії та практики ігрового аудіо: К. Коллінз «Game Sound», А. Маркса «The Complete Guide to Game Audio», Р. В'єрса «The Sound Effects Bible», В. Філліпс «A Composer's Guide to Game Music». Розглянуто теоретичні аспекти геймдизайну та соціокультурного значення ігор у працях Дж. Шелла, К. Сален та Е. Циммермана. Також вагому частину дослідження склали праці вітчизняних науковців (К. Станіславської, Н. Бєлявіної, Л. Рязанцева, В. Дьяченка, А. Ананьєва, О. Бут, В. Грищенка), що дозволило розглядати відеогру не лише як технічний продукт, а як синтетичний вид мистецтва.
2. Здійснено історичний огляд еволюції відеоігор та ігрового звуку: від примітивних сигналів епохи аркадних автоматів та 8-бітних консолей до сучасних систем імерсивного аудіо. Виявлено, що технічний прогрес (поява звукових чіпів, CD-носіїв, мідлверу) став каталізатором художніх трансформацій. Встановлено, що сучасний етап розвитку індустрії характеризується злиттям кінематографічних стандартів якості з принципами інтерактивності.
3. Проаналізовано функції звукового дизайну в ігровому середовищі. Виокремлено п'ять ключових функцій: атмосферну, ідентифікаційну, комунікативну, інтерактивну та імерсійну. На основі теоретичних розвідок доведено, що звук у грі виконує роль «доданої вартості», збагачуючи візуальний ряд інформацією та

емоціями. Обґрунтовано важливість психологічного впливу звуку для утримання «магічного кола» гри та забезпечення зворотного зв’язку (Feedback) на дії гравця.

4. Окреслено специфіку інтерактивної музики, яка, на відміну від лінійної, функціонує як гнучка адаптивна система. Розглянуто методи горизонтального перерозподілу та вертикального ремікшування, що дозволяє музичному супроводу динамічно реагувати на зміни ігрового стану (дослідження, бій, діалог), уникаючи ефекту повторюваності та слухової втоми.
5. У практичній частині роботи створено ігровий прототип у середовищі Unity на базі сцени «Viking Village». Реалізовано механіку пересування персонажа та підготовлено візуальне середовище для інтеграції аудіо. Проведено порівняльний аналіз вбудованої аудіосистеми Unity та професійного мідлверу FMOD Studio, обґрунтувавши вибір останнього завдяки можливостям параметричного керування та профілювання ресурсів у реальному часі.
6. Автором здійснено повний цикл розробки та імплементації звукового дизайну. Записано та оброблено оригінальні семпли, реалізовано складні інтерактивні системи: адаптивну систему кроків (зміна звуку залежно від поверхні), динамічну атмосферу (вітер, океан), а також застосовано техніки просторового аудіо та оклюзії для створення реалістичного акустичного простору.
7. Написано та імплементовано авторський музичний супровід у стилістиці середньовічної музики. Композицію створено з використанням віртуальних інструментів та адаптовано для безшовного циклічного відтворення. У середовищі FMOD налаштовано баланс між музикою та ефектами, що забезпечило емоційну виразність прототипу.

8. Узагальнюючи результати роботи, варто зазначити: у ході дослідження було проаналізовано роль звукорежисера в геймдеві. Спираючись на праці Н. Бєлявіної [2], К. Станіславської [9] та В. Дьяченка [6], ми визначили мистецький контекст професії. Практична частина підтвердила, що використання мідлверу FMOD у поєднанні з Unity [43] значно розширює творчі можливості звукорежисера, дозволяючи створювати адаптивні звукові ландшафти, що реагують на дії гравця.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ананьев А. Б. Акустика для звукорежиссеров. Київ : Фенікс, 2012. 256 с.
2. Бєлявіна Н. Д. Методологія та методика викладання фахових мистецьких дисциплін : навч. посіб. Київ : НАКККіМ, 2017. 160 с.
3. Бут О. В. Звук як компонент образної структури фільму. Мистецтвознавчі записки. 2013. Вип. 24. С. 254–261.
4. Гейзінга Й. Homo Ludens. Людина, що грається. Київ : Основи, 1994. 250 с.
5. Грищенко В. І. Композиція та комп'ютерне аранжування : навч.-метод. посіб. Київ : НАКККіМ, 2015. 112 с.
6. Дьяченко В. В. Творча діяльність українських звукорежисерів другої половини XX – початку XXI століття: теорія, історія, практика : дис. … канд. мистецтвознавства. Київ, 2018. 220 с.
7. Лісса З. Естетика кіномузики. Київ : Мистецтво, 1958. 456 с.
8. Рязанцев Л. В. Звукорежисура : навч. посібник. Київ : ДАКККіМ, 2009. 144 с.
9. Станіславська К. І. Мистецько-видовищні форми сучасної культури : монографія. Київ : НАКККіМ, 2016. 352 с.
10. Baer R. Videogames: In the Beginning. Union : Rolenta Press, 2005. 280 p.
11. Bridgett R. Dynamic Range: Subtlety and Silence in Video Game Sound. From Pac-Man to Pop Music: Interactive Audio in Games and New Media. Farnham : Ashgate Publishing, 2008.
12. Carlsson A. Chip Music: Low-Tech Data Music Sharing. From Pac-Man to Pop Music: Interactive Audio in Games and New Media. Farnham : Ashgate Publishing, 2008. P. 15–23.
13. Chion M. Audio-Vision: Sound on Screen. New York : Columbia University Press, 1994. 268 p.
14. Collins K. Game Sound: An Introduction to the History, Theory, and Practice of Video Game Music and Sound Design. Cambridge : The MIT Press, 2008. 216 p.
15. Dolby Laboratories. Dolby Atmos for Gaming: Technical Overview. Dolby White Paper Series. 2020.
16. Donovan T. Replay: The History of Video Games. East Sussex : Yellow Ant Media, 2010. 501 p.
17. Farnell A. Designing Sound. Cambridge : The MIT Press, 2010. 664 p.
18. Firelight Technologies. FMOD Studio User Manual [Електронний ресурс]. URL: https://www.fmod.com/docs (дата звернення: 21.05.2020).
19. Gibson D. The Art of Mixing: A Visual Guide to Recording, Engineering, and Production. Boston : Course Technology, 2005. 279 p.
20. Grimshaw M. Sound and Immersion in the First-Person Shooter. International Journal of Intelligent Games & Simulation. 2008. Vol. 5, No. 1. P. 119–124.
21. Harris B. Console Wars: Sega, Nintendo, and the Battle that Defined a Generation. London : Atlantic Books, 2014. 576 p.

22. Herman L. Phoenix: The Rise and Fall of Video Games. Union : Rolenta Press, 1997. 312 p.
23. Horowitz N. The Art of Video Game Curating. San Francisco : No Starch Press, 2020.
24. Izhakevych S. The World of Computer Gaming. Kyiv : NaUKMA, 2019.
25. Jørgensen K. "What are Those Grunts and Growls Over There?" Computer Game Audio and Player Action : PhD dissertation. Copenhagen : Copenhagen University, 2007.
26. Kent S. L. The Ultimate History of Video Games: From Pong to Pokémon and Beyond. New York : Crown Publishing Group, 2001. 624 p.
27. Kocurek C. Coin-Operated Americans: Reimagining Boyhood with the Video Game. Minneapolis : University of Minnesota Press, 2015. 288 p.
28. Kohler C. Power-Up: How Japanese Video Games Gave the World an Extra Life. Indianapolis : BradyGames, 2016. 336 p.
29. Loguidice B., Barton M. Vintage Games: An Insider Look at the History of Grand Theft Auto, Super Mario, and the Most Influential Games of All Time. Boston : Focal Press, 2009. 408 p.
30. Marks A. The Complete Guide to Game Audio: For Composers, Musicians, Sound Designers, and Game Developers. 2nd ed. Burlington : Focal Press, 2008. 390 p.
31. Montfort N., Bogost I. Racing the Beam: The Atari Video Computer System. Cambridge : The MIT Press, 2009. 192 p.
32. O'Donnell C. Developer's Dilemma: The Secret World of Videogame Creators. Cambridge : The MIT Press, 2014.
33. Phillips W. A Composer's Guide to Game Music. Cambridge : The MIT Press, 2014. 296 p.
34. Salen K., Zimmerman E. Rules of Play: Game Design Fundamentals. Cambridge : The MIT Press, 2004.
35. Schell J. The Art of Game Design: A Book of Lenses. Boca Raton : A K Peters/CRC Press, 2008. 600 p.
36. Schnyder D. The Sound of Super Nintendo. The Music and Sound of Experimental Film. Oxford : Oxford University Press, 2008.
37. Schreier J. Blood, Sweat, and Pixels: The Triumphant, Turbulent Stories Behind How Video Games Are Made. New York : HarperCollins, 2017. 304 p.
38. Sheff D. Game Over: How Nintendo Conquered the World. New York : Random House, 1993. 451 p.
39. Stevens R., Raybould D. The Game Audio Tutorial: A Practical Guide to Sound and Music for Interactive Games. Boston : Focal Press, 2011. 400 p.
40. Summers T. Understanding Video Game Music. Cambridge : Cambridge University Press, 2016. 250 p.
41. Surry R. Ambient Sound Design in Video Games. Gamasutra. 2016.
42. Targett G. Game Audio: Tales of a Technical Sound Designer. London : Routledge, 2018.

43. Unity Technologies. Unity User Manual [Електронний ресурс]. URL: https://docs.unity3d.com/Manual/index.html (дата звернення: 20.05.2020).
44. Vendel K., Goldberg M. Atari Inc.: Business Is Fun. Carmel : Syzygy Press, 2012. 800 p.
45. Viers R. The Sound Effects Bible: How to Create and Record Hollywood Style Sound Effects. Studio City : Michael Wiese Productions, 2008. 368 p.
46. Whittington W. Sound Design and Science Fiction. Austin : University of Texas Press, 2007. 296 p.
47. Wolf M. J. P. Encyclopedia of Video Games: The Culture, Technology, and Art of Gaming. Santa Barbara : Greenwood Press, 2012. 789 p.
48. Wolf M. J. P. The Video Game Explosion: A History from PONG to PlayStation and Beyond. Westport : Greenwood Press, 2008. 380 p.

ДОДАТКИ

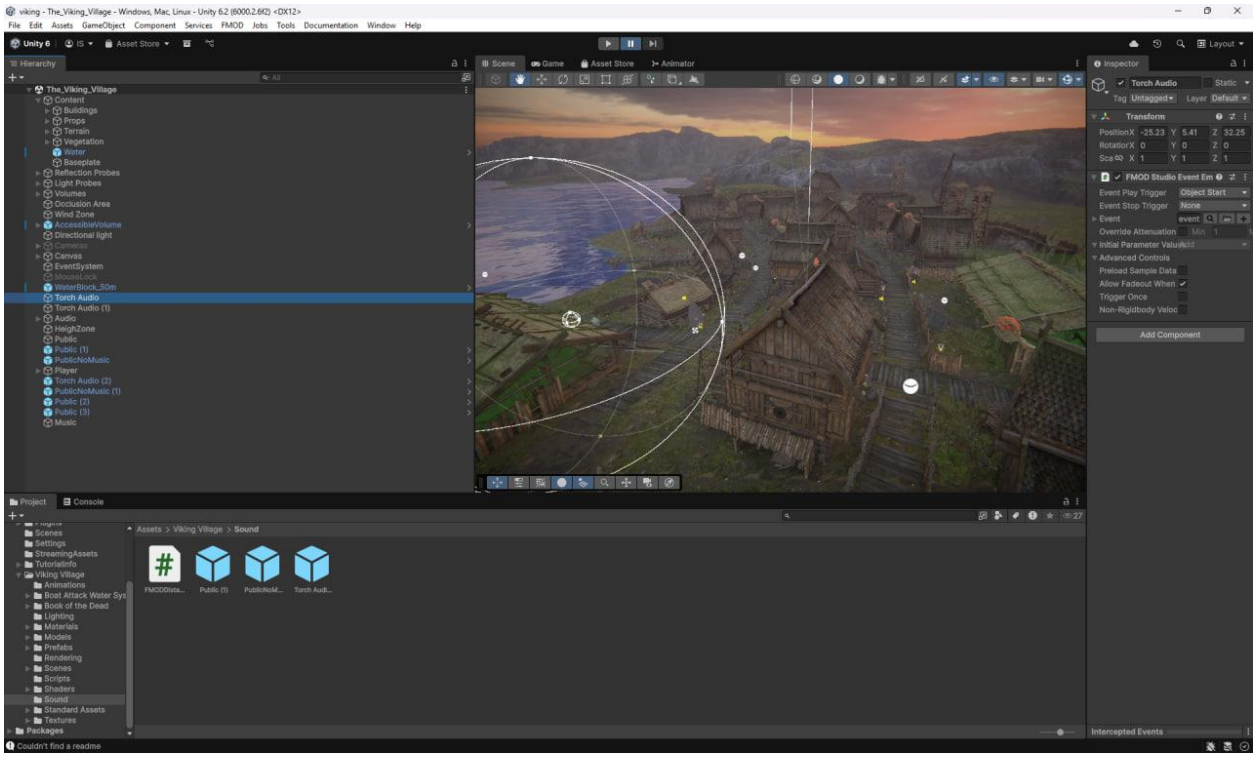
Додаток А

Екран осцилографа з грою «Tennis for Two» (1958).

Додаток Б

Ігрова консоль Atari 2600 як приклад домашньої ігрової системи.

Додаток В



Візуалізація авторського ігрового прототипу (сцена «Viking Village») в Unity.

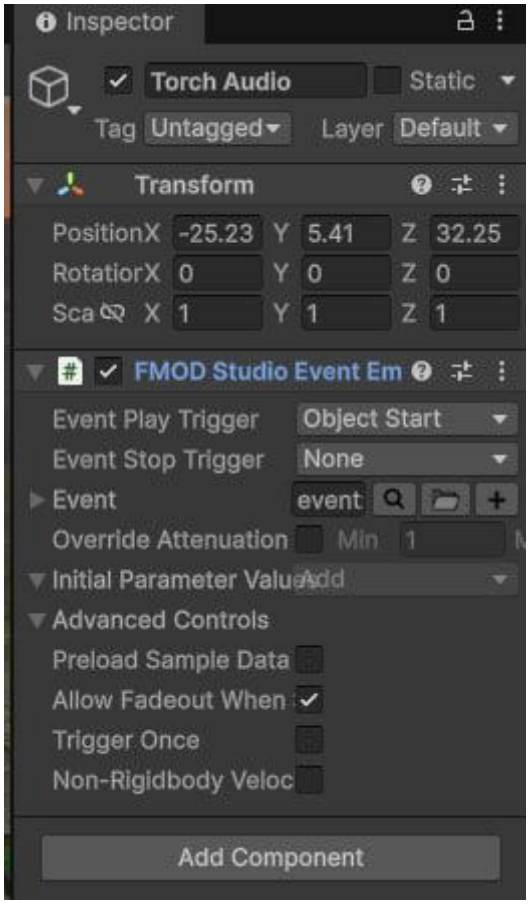
Додаток Г

Обладнання для польового запису (рекордер Zoom H1n).

Додаток Д

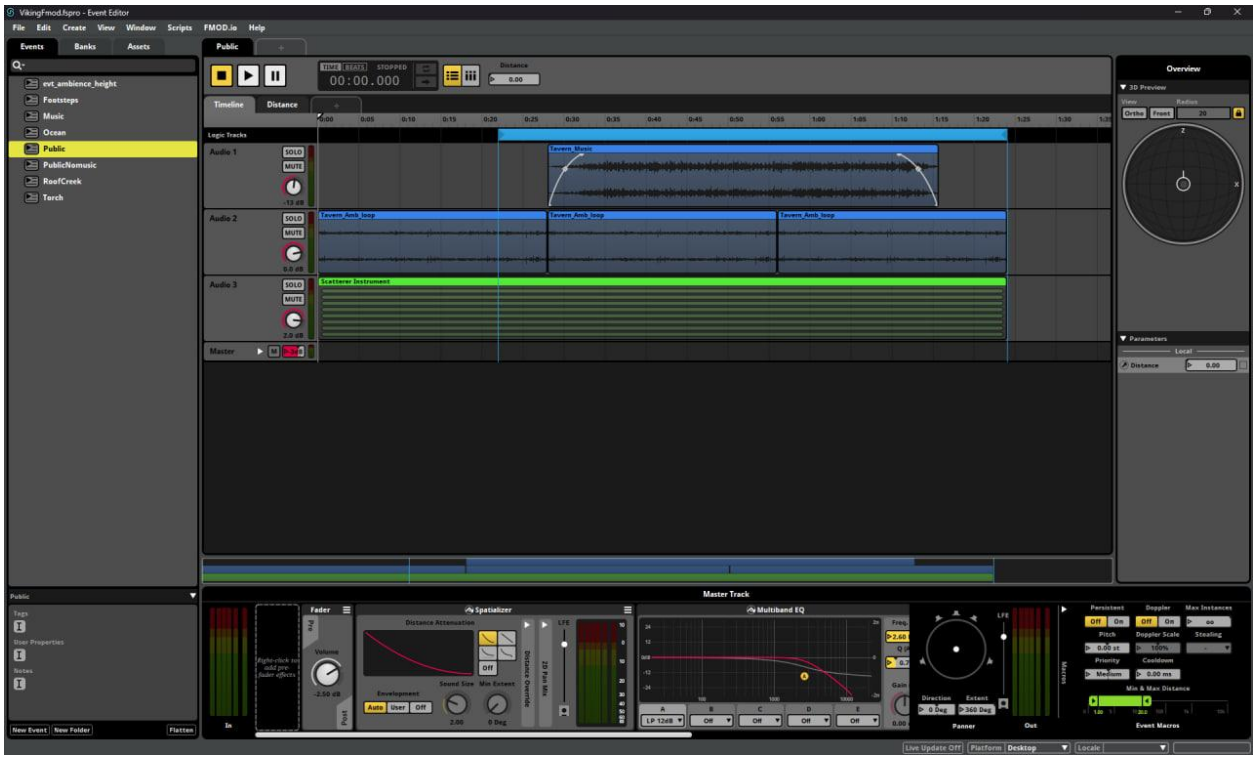
Спектральна обробка та еквалізація аудіосемплів у DAW.

Додаток Е



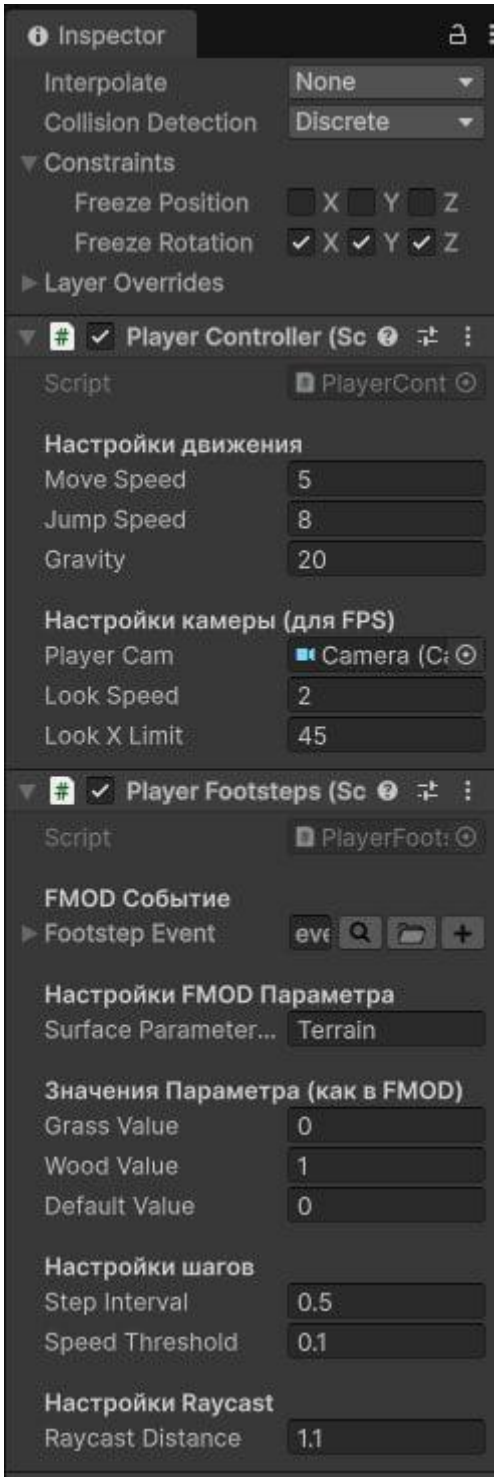
Налаштування компонента FMOD Studio Event Emitter в Unity.

Додаток Є



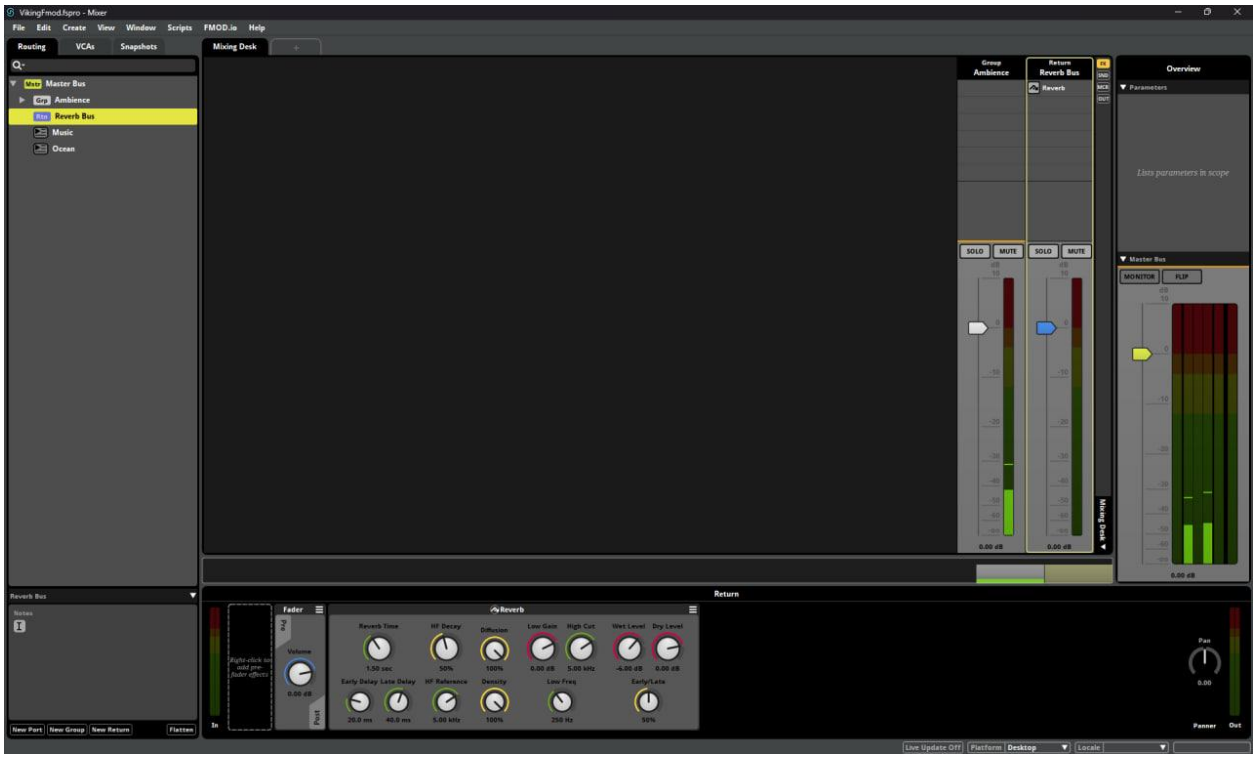
Графік просторового загасання звуку (Distance Attenuation) у FMOD.

Додаток Ж



Параметрична структура події «Footsteps» (адаптивні кроки).

Додаток З

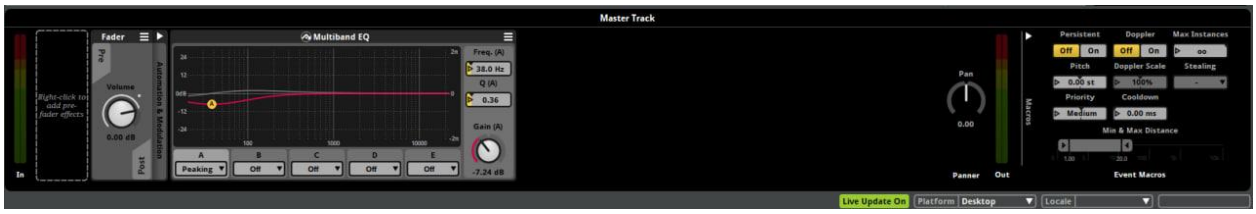


Маршрутизація сигналу та налаштування Send-ефектів у мікшері FMOD.

Додаток К

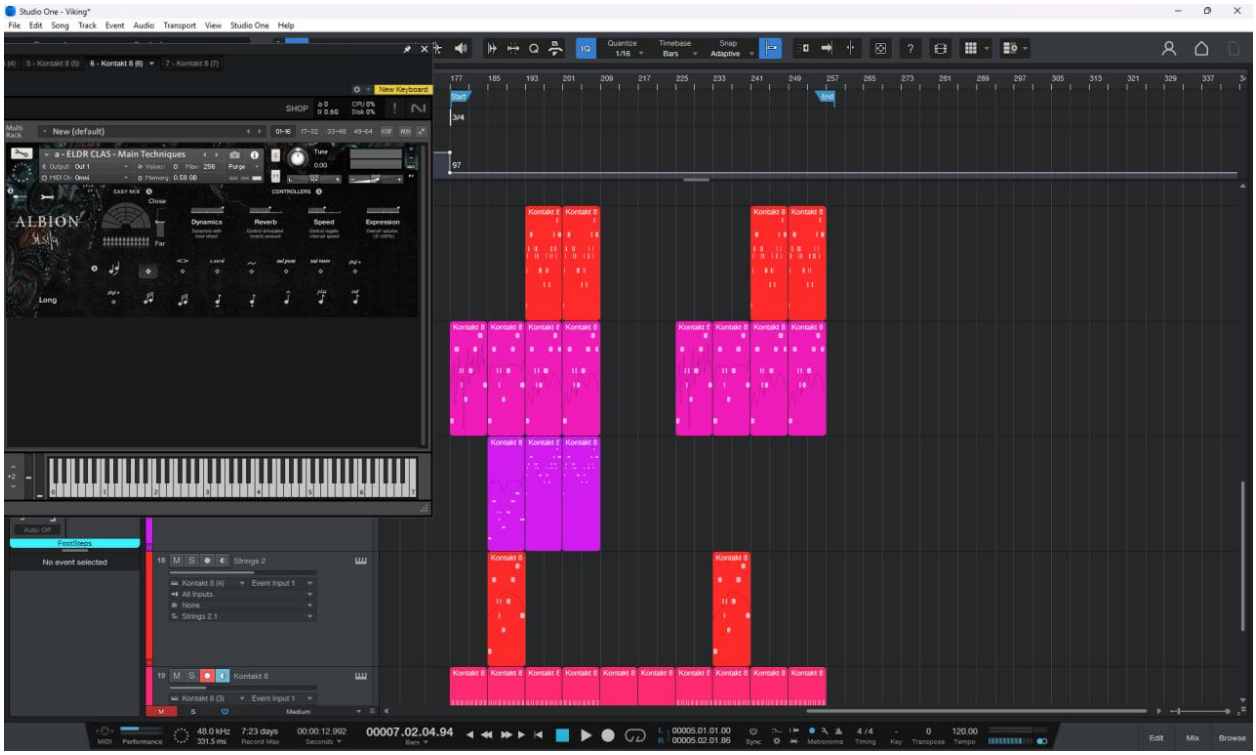
Організація банків та налаштування режиму Streaming для оптимізації.

Додаток Л



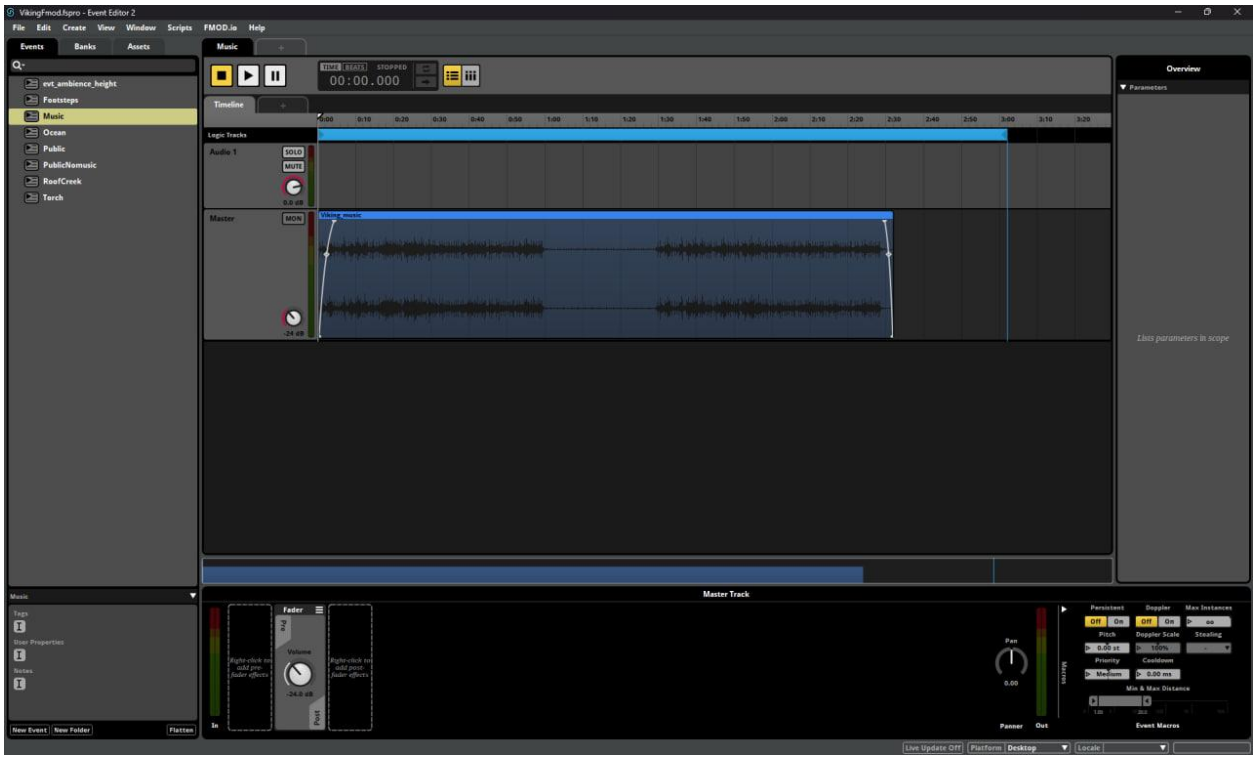
Індикація активного з'єднання Live Update для налагодження звуку в реальному часі.

Додаток М



Проєкт музичного аранжування та віртуальні інструменти в DAW.

Додаток Н



Налаштування циклічного відтворення (Loop Region) музики у FMOD.

Додаток П

Флеш-носій:

1.Текст кваліфікаційної роботи на тему «Особливості роботи звукорежисера під час створення музично-звукового образу відеогри»

2.Аудіо-відео додатки

3. Перевірка на плагіат. Звіт подібності: **Strikeplagiarism.com**