

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА ТЕХНОЛОГІЙ
Кафедра комп'ютерних інформаційних технологій

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач кафедри

_____Аліна САВЧЕНКО

«_____»_____2023 р.

КВАЛІФІКАЦІЙНА РОБОТА

(ДИПЛОМНА РОБОТА, ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ

“МАГІСТРА”

ЗА ОСВІТНЬО-ПРОФЕСІЙНОЮ ПРОГРАМОЮ “ІНФОРМАЦІЙНІ
УПРАВЛЯЮЧІ СИСТЕМИ ТА ТЕХНОЛОГІЇ”

Тема: «Метод підвищення ефективності ігрового процесу в Java-застосунку з вдосконаленою моделлю поведінки гравців»

Виконав: студент групи УС-211М Горічко Андрій Олегович

Керівник: професор кафедри КІТ Воронін Альберт Миколайович

Нормоконтролер _____Ігор РАЙЧЕВ

Київ – 2023

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
Факультет Комп'ютерних наук та технологій
Кафедра Комп'ютерних інформаційних технологій
Галузь знань, спеціальність, освітньо-професійна програма: 12
“Інформаційні технології”, 122 “Комп'ютерні науки”, “Інформаційні
управляючі системи та технології”

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Аліна САВЧЕНКО

" ___ " _____ 2023 р.

ЗАВДАННЯ

на виконання кваліфікаційної роботи студента

Горічка Андрія Олеговича

(прізвище, ім'я, по батькові)

1. Тема роботи: «Метод підвищення ефективності ігрового процесу в Java-застосунку з вдосконаленою моделлю поведінки гравців», затверджена наказом ректора від “29” вересня 2023р. за № 1976/ст

2. Термін виконання роботи: з 02 жовтня 2023р. по 31 грудня 2023р.

3. Вихідні дані до роботи: підвищення ефективності ігрового процесу в Java-застосунку шляхом розробки та впровадження вдосконаленої моделі поведінки гравців.

4. Зміст пояснювальної записки: вступ, вплив моделі поведінки гравців на ігровий процес, розробка Java-застосунку та моделі поведінки гравців, імплементація та налаштування моделі, проведення експериментів та аналіз результатів, висновки.

5. Перелік обов'язкового ілюстраційного матеріалу: актуальність теми, застосунок, технології які використовувались в додатку, функціональна схема, оптимізація моделі, результати.

6. Календарний план-графік

№ з/п	Завдання	Термін виконання	Підпис керівника
1.	Пошук і дослідження наукових джерел.	02.10.2023 – 08.10.2023	
2.	Розроблення та затвердження календарного плану виконання дипломної роботи.	09.10.2023 – 10.10.2023	
3.	Проведення консультацій з науковим керівником.	11.10.2023 – 13.10.2023	
4.	Написання Розділу 1. Вплив моделі поведінки гравців на ігровий процес.	14.10.2023 – 23.10.2023	
5.	Написання Розділу 2. Розробка Java-застосунку та Моделі поведінки гравців.	24.10.2023 – 08.11.2023	
6.	Написання Розділу 3. Імплементация та Налаштування моделі.	09.11.2023 – 22.11.2023	
7.	Написання Розділу 4. Проведення експериментів та аналіз результатів.	23.11.2023 – 30.11.2023	
8.	Оформлення пояснювальної записки дипломної роботи.	01.12.2023 – 04.12.2023	
9.	Написання, друк та підписання Рецензії у рецензента та Відгуку керівника у встановленому порядку.	05.12.2023 – 10.12.2023	
10.	Створення Презентації та доповіді.	11.12.2023 – 12.12.2023	
11.	Підготовка до захисту та попередній захист дипломної роботи на випусковій кафедрі.	13.11.2023 – 17.12.2023	

7. Дата видачі завдання: «02» жовтня 2023 р.

Керівник дипломної роботи _____
(підпис керівника)

Альберт ВОРОНІН
(П.І.Б.)

Завдання прийняв до виконання _____
(підпис випускника)

Андрій ГОРІЧКО
(П.І.Б.)

РЕФЕРАТ

Пояснювальна записка до дипломної роботи «Метод підвищення ефективності ігрового процесу в Java-застосунку з вдосконаленою моделлю поведінки гравців» викладена на 101 сторінках і містить 36 рисунків та 3 таблиці. Список бібліографічних посилань складається з 40 найменувань.

Ключові слова: JAVA, JAVA FX, БІБЛІОТЕКА LIBGDX, ТЕХНОЛОГІЯ SPINE, ГЕЙМДЕВ, КРОСПЛАТФОРМЕНІСТЬ, ОПТИМІЗАЦІЯ, МОДЕЛЬ ПОВЕДІНКИ ГРАВЦІВ.

Актуальність дипломної роботи полягає в розробці та впровадженні вдосконаленої моделі поведінки гравців у Java-застосунках, спрямованій на значуще підвищення якості геймплею. Ця нова модель враховує сучасні тенденції та виклики в геймдеві, пропонуючи ефективні алгоритми та стратегії, що сприятимуть реалістичній та захопливій взаємодії гравців у віртуальному середовищі.

Мета дипломної роботи: підвищення ефективності ігрового процесу в Java-застосунку шляхом розробки та впровадження вдосконаленої моделі поведінки гравців.

Об'єкт дослідження: ефективність і вдосконалення ігрового процесу в Java-застосунках за допомогою впровадженої моделі поведінки гравців.

Предмет дослідження: вплив та оптимізація ігрового процесу в Java-застосунках з використанням вдосконаленої моделі поведінки гравців.

У роботі висвітлено:

- дослідження впливу моделей поведінки гравців на динаміку та інтенсивність геймплею.
- розробку ефективної та реалістичної моделі поведінки гравців на основі отриманих досліджень.
- розробку архітектури програми та її компонентів, розробку ключових компонентів системи та їх взаємодії на основі обраної архітектури.
- впровадження оптимізованої моделі поведінки гравців у готовий Java-застосунок.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ.....	7
ВСТУП	8
РОЗДІЛ 1 ВПЛИВ МОДЕЛІ ПОВЕДІНКИ ГРАВЦІВ НА ІГРОВИЙ ПРОЦЕС	11
1.1. Огляд сучасного стану галузі розробки ігор	11
1.1.1. Сучасний стан галузі розробки ігор.....	11
1.1.2. Тенденції та виклики	14
1.2. Роль поведінкових моделей гравців у вдосконаленні ігрового процесу.....	19
1.2.1. Поведінкові моделі гравців як ключовий елемент гри.....	19
1.2.2. Вплив поведінкових моделей на геймплей.....	20
1.3. Ігрові двигуни та їхнє використання в Java-застосунках.....	21
1.3.1. Огляд ігрових движків та їх функції.....	21
1.3.2. Ігрові двигуни для Java-застосунків	27
1.4. Огляд інструментів та технологій для розробки Java-ігор	35
1.4.1. Вибір мови програмування для розробки	35
1.4.2. Використані бібліотеки та інструменти	36
Висновки до розділу 1	38
РОЗДІЛ 2 РОЗРОБКА JAVA-ЗАСТОСУНКУ ТА МОДЕЛІ ПОВЕДІНКИ ГРАВЦІВ.....	39
2.1. Архітектура програми та її компоненти	39
2.1.1. Огляд архітектури програми	39
2.1.2. Ключові компоненти системи	42
2.2. Опис моделі поведінки гравців.....	47
2.2.1. Детальний опис моделі поведінки гравців.....	47
2.2.2. Алгоритми та стратегії, що використовуються	49
Висновки до розділу 2	51
РОЗДІЛ 3 ІМПЛЕМЕНТАЦІЯ ТА НАЛАШТУВАННЯ МОДЕЛІ.....	53
3.1. Імплементация моделі поведінки гравців у Java-застосунку	53
3.1.1. Опис імплементации моделі у програмі	53

3.1.2. Використані технології та інструменти	65
3.2. Методи створення та налаштування моделі	70
3.2.1. Пояснення процесу створення моделі.....	70
3.2.2. Налаштування алгоритмів і стратегій	71
3.3. Роль моделі у вдосконаленні ігрового досвіду	73
3.3.1. Вплив моделі поведінки на ігровий процес.....	73
3.3.2. Вдосконалення геймплею завдяки моделі поведінки гравців	74
Висновки до розділу 3	75
РОЗДІЛ 4 ПРОВЕДЕННЯ ЕКСПЕРИМЕНТІВ ТА АНАЛІЗ РЕЗУЛЬТАТІВ.	77
4.1. Методика проведення експериментів	77
4.1.1. Організація та планування експериментальних досліджень	77
4.1.2. Підбір параметрів та умов експерименту	78
4.2. Зібрані дані та їхні характеристики	80
4.2.1. Збір та обробка експериментальних даних.....	80
4.2.2. Статистичний аналіз отриманих результатів	81
4.3. Аналіз результатів експерименту	86
4.3.1. Опис проведених експериментів та тестів.....	86
4.3.2. Оптимізація процесів у грі.....	88
4.4. Аналіз стратегій імплементації.....	89
Висновки до розділу 4	95
ВИСНОВКИ.....	96
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	99

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

ML	Machine Learning
DL	Deep Learning
NLP	Natural Language Processing
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
CDN	Content Delivery Network
API	Application Programming Interface
GUI	Graphical User Interface
VR	Virtual Reality
AR	Augmented Reality
HCI	Human-Computer Interaction
aaS	as a Service
SaaS	Software as a Service
PaaS	Platform as a Service
IaaS	Infrastructure as a Service
REST	Representational State Transfer
SDN	Software-Defined Networking
VCS	Version Control System
MLaaS	Machine Learning as a Service
DDS	Data Distribution Service
DevOps	Development and Operations
CI/CD	Continuous Integration/Continuous Deployment
UX	User Experience
MVP	Minimum Viable Product
ROI	Return on Investment

ВСТУП

Актуальність. Сучасна геймінг-індустрія стала важливою частиною культурного та розважального ландшафту, залучаючи мільйони гравців світового рівня. З ростом технологій і розвитком геймдеву, відкриваються нові можливості, однак із цим виникають нові завдання та виклики. Однією з ключових проблем, яку сьогодні ставлять перед розробниками ігор, є забезпечення ефективності ігрового процесу та створення реалістичної та цікавої моделі поведінки гравців.

Проблема ефективності геймплею стає актуальною в умовах стрімкого технологічного прогресу та росту очікувань гравців. Сучасні геймери вимагають не лише графічної краси, але й плавності гри на різних пристроях та конфігураціях. Оптимізація ігрового процесу є важливим аспектом для забезпечення комфортного користування та високої якості геймплею.

У контексті інтенсивного розвитку галузі виникає питання еволюції моделі поведінки гравців. Сучасні гравці очікують від персонажів у грі більшої інтелектуальної складності, адаптивності та реалістичності у вирішенні завдань. Це ставить перед розробниками виклик розробки вдосконаленої моделі поведінки, яка здатна адаптуватися до стилів гри кожного окремого гравця, забезпечуючи відчуття власної унікальності та участі в геймплейному процесі.

Java, яка визнана своєю кросплатформенністю та гнучкістю, виявляється важливою мовою програмування для розробників ігор. Проте, проблеми ефективності і необхідність оптимізації для роботи на різних пристроях залишаються важливими завданнями для її використання в ігровій сфері.

Дослідження в області підвищення ефективності ігрового процесу та вдосконалення моделі поведінки гравців в Java-застосунках визначається необхідністю вирішення цих викликів. Оптимізація геймплею та розробка інтелектуальних, реалістичних агентів може принести новий рівень задоволення для гравців та підняти планку геймдеву в цілому.

Дослідження розпочнеться оглядом сучасного стану галузі розробки ігор та визначенням основних тенденцій та викликів. Далі буде проведено аналіз впливу моделі поведінки гравців на ігровий процес, оглянуті ігрові двигуни та їхнє використання в Java, а також розглянуті інструменти та технології для розробки Java-ігор.

Розділ, присвячений розробці Java-застосунку та моделі поведінки гравців, включатиме огляд архітектури програми, детальний опис моделі поведінки, алгоритми та стратегії. Після цього буде проведена імплементація моделі у Java-застосунку та розглянуті методи створення та налаштування моделі, враховуючи роль моделі у вдосконаленні ігрового досвіду.

Експериментальний розділ передбачає методику проведення експериментів, збір та аналіз отриманих даних, а також аналіз результатів та їх вплив на ігровий процес.

Магістерська робота відкриває можливості для розробки інноваційних підходів у сфері геймдеву, вдосконалення геймплею та підвищення конкурентоспроможності Java-ігор на ринку. Високотехнологічне поєднання оптимізації та реалістичності в іграх може стати новим кроком у розвитку ігрової індустрії та задоволенні потреб сучасного гравця.

Метою дослідження – є підвищення ефективності ігрового процесу в Java-застосунку шляхом розробки та впровадження вдосконаленої моделі поведінки гравців.

Для досягнення поставленої мети вирішуються такі наукові завдання:

- Розробка архітектури програми та її компонентів.
- Дослідження впливу моделей поведінки гравців на динаміку та інтенсивність геймплею.
- Розробка ефективної та реалістичної моделі поведінки гравців на основі отриманих досліджень.
- Розробка ключових компонентів системи та їх взаємодії на основі обраної архітектури.

- Впровадження оптимізованої моделі поведінки гравців у готовий Java-застосунок.

Об'єктом дослідження – є ефективність і вдосконалення ігрового процесу в Java-застосунках за допомогою впровадженої моделі поведінки гравців.

Предметом дослідження – є вплив та оптимізація ігрового процесу в Java-застосунках з використанням вдосконаленої моделі поведінки гравців.

Наукова новизна роботи – полягатиме в розробці та впровадженні вдосконаленої моделі поведінки гравців у Java-застосунках, спрямованій на значуще підвищення якості геймплею. Ця нова модель враховує сучасні тенденції та виклики в геймдеві, пропонуючи ефективні алгоритми та стратегії, що сприятимуть реалістичній та захопливій взаємодії гравців у віртуальному середовищі.

РОЗДІЛ 1

ВПЛИВ МОДЕЛІ ПОВЕДІНКИ ГРАВЦІВ НА ІГРОВИЙ ПРОЦЕС

1.1. Огляд сучасного стану галузі розробки ігор

1.1.1. Сучасний стан галузі розробки ігор

Ігрова галузь переносить інтенсивний період розвитку та стає важливою частиною індустрії розваг. З кожним роком, кількість, якість та різновид випущених ігор зростає. В тому числі, геймдев стає ключовим сегментом технологічного ринку, що забезпечує великі прибутки та залучення талановитих розробників.

На даний момент ігри вражають своїм геймплеєм, сюжетом, графікою, звуковим оформленням, підтримкою виробника та підтримкою на різних платформах. Віртуальна реальність (VR) та розширена реальність (AR) розвивається та додає нові можливості для імерсивного геймплею. Новий рівень деталізації та реалізму забезпечують ігрові консолі нового покоління та потужні графічні карти для ПК.

Онлайн-ігри і геймінгові платформи вийшли на новий рівень, через що стали все більш популярними, такі як Xbox Live, PlayStation Network, Epic Games Store та Steam. Також розширюється спектр мобільних ігор для смартфонів та планшетів.

Ігри, у теперішній час, не тільки створюються з розважальною ціллю, а і з метою отримати не поганий прибуток в мільярди доларів виручки щороку. Розробка ігор – це складний та багатогранний процес, який вимагає використання передових технологій та моніторингу залежностей продуктів серед користувачів.

Кафедра КІТ (47)				НАУ 23.05.85.000 ПЗ			
Виконав	Горічко А.О.			ВПЛИВ МОДЕЛІ ПОВЕДІНКИ ГРАВЦІВ НА ІГРОВИЙ ПРОЦЕС	Літ.	Арк.	Аркушів
Керівник	Воронін А.М.					11	28
Консульт.					УС-211М 122		
Н. Контр.	Райчев І.Е.						

Сьогодні розробка ігрової індустрії, яка створюється на різних платформах та виробляє ігри, розвивається у дуже швидких темпах. Вона стає найприбутковішою галуззю у світі розваг, залучаючи не лише розробників, а й широку аудиторію.

Віртуальна реальність, реалістична графіка, захоплення рухів – цих всіх новітніх технологій вимагають сучасні ігри. Для розробки ігор використовуються такі основні технології як:

- Штучний інтелект – використовується розробниками для створення реалістичних персонажів, які можуть взаємодіяти з гравцем та приймати рішення. Розробники ігор мають унікальні можливості для розробки ігор за допомогою VR та AR пристроїв та інших різноманітних платформ. Усі платформи є унікальні по своєму, що надає можливість створювати нові ігрові продукти звертаючи увагу на особливості кожної з платформ.

Через певні тенденції в галузі ігор, напрямок індустрії їхнього розвитку змінюється та розвивається. Деякі приклади з основних трендів у розробці ігор наведено нижче:

- Мобільні ігри, з кожним днем мобільні ігри стають все популярнішими через глобальне збільшення використання смартфонів та планшетів.

- Мультиплеєрні ігри, також стають все більш популярними, тому що дають можливість гравцям взаємодіяти один з одним у реальному часі гри, та розраховані на велику кількість користувачів.

- Ширші технічні можливості платформ надають можливість надати високу якість графіки та потужність механіки для ігор на ПК або консолей.

- Розширення віртуальної реальності, Незважаючи на те, що VR-технології не є доступні для всіх, з кожним днем вони набирають все більшу популярність та все більше розвиваються, що дозволяє відкрити нові можливості для створення нових та захоплюючих ігрових світів.

Вплив стрімінгових платформ на розробку та маркетинг ігор:

Демонстрація ігрового процесу – на даний момент, одні з найпопулярніших стрімінгових платформ є Twitch та YouTube Gaming, які

дозволяють транслювати свою гру на мільйони глядачів, що зацікавлює стримерів та їх аудиторію та стимулює розробників створювати нові ігри.

Пряма взаємодія з гравцям – за допомогою сучасних технологій, у реальному часі, стрімери можуть допомагати зрозуміти розробникам, які ігри є актуальним та які аспекти гри потребують покращення, щоб залучити більшу кількість аудиторії та зробити гру комфортною для кожного.

Маркетинг та реклама – так як стрімери мають великий вплив на свою аудиторію, вони злегкістю можуть привернути увагу людей до нових ігор, демонструючи гру на стрімах.

До ролі спільноти та соціальних мереж у просуванні та розвитку ігор, можна віднести наступні пункти:

1. Створення спільноти навколо гри. Обговорення новин та оновлень гри відбувається у ігрових спільнотах, що зміцнює лояльність до продукту.

2. Зворотній зв'язок та покращення гри. Беручи до уваги відгуки гравців у соціальних мережах, розробники можуть звернути увагу на потреби аудиторії, що варто покращити та що змінити у грі.

3. Вірусний маркетинг. Користувачі соціальних мереж діляться враженнями від ігор, створюючи вірусний ефект, що допомагає розповсюджувати інформацію про нові ігри.

Важливу роль відіграють соціальні мережі та стрімінгові платформи, вони дозволяють користувачам ділитися своїми враженнями та розповсюджувати інформацію про нові ігри, тим самим просуваючи нові продукти ігрової сфери та встановлюючи зв'язок між розробниками та ігровою спільнотою.

Галузь розробки ігор є динамічною та інноваційною сферою, яка відслідковує актуальні тренди та використовує новітні технології. VR-технології, штучний інтелект та хмарні технології – з кожним днем все більше розвиваються та рухаються уперед, тому майбутнє розробки ігор обіцяє бути ще більш , багатостороннім та захоплюючим. Ми можемо спостерігати за все більш успішними іграми, які підтверджують своєю статистикою та значущістю, що індустрія ігор, має велику перспективу та майбутнє.

1.1.2. Тенденції та виклики

Межі можливостей у розробці ігор постійно розширюються, тому з'являються нові тенденції технологій у ігровому світі та змінюється погляд на створення відеоігор.

Використання штучного інтелекту (ШІ) в іграх – штучний інтелект використовується для створення реалістичних персонажів, оптимізації ігрового процесу та створення розумних супротивників

Розвиток віртуальної та розширеної реальності – ігри VR та AR набувають популярності, оскільки дозволяють гравцям занурюватися в інші світи та взаємодіяти з ними.

Гейміфікація – такі елементи використовуються в різних сферах, включаючи освіту, бізнес та здоров'я, з метою стимулювання участі та досягнення цілей.

Кросплатформенність – це коли гравці хочуть мати можливість грати, зберігати прогрес і взаємодіяти з іншими гравцями на різних платформах, включаючи консолі, ПК і мобільні пристрої.

Екологічні аспекти – при розробці ігор все більше уваги приділяється екологічності, зокрема зменшенню вуглецевого сліду та використанню відновлюваних ресурсів.

Етичні питання – в ігровій індустрії все більше дискусій про етичні аспекти, такі як представництво різних соціально-культурних груп і управління даними про гравців.

Створюючи нові виклики та відкриваючи нові можливості перед розробниками для створення захоплюючих та іммерсивних ігрових досвідів, індустрія ігор все більше та швидше розвивається.

Багате 3D-середовище

Створення тривимірних середовищ зі складним освітленням, текстурованням та фізикою стає все більш детальнішим за допомогою потужного обладнання. Через розсунення меж можливого, гравці занурюються в

атмосферу гри та насолоджуються великими відкритими світами та складними погодними системами.

Процедурна генерація

Для створення контенту в грі використовують алгоритми з процедурної генерації, що допомагає створювати персонажей, карти, квести або навіть цілі світи. Щоб створити унікальний досвід для кожного окремого користувача, розробники створюють унікальний динамічний контент який реагує на дії гравця.

Фотореалізм

Покращена графічна технологія допомагає створювати більш реалістичну графіку. Трасування променів – один з методів досягнення фотореалізму, який застосовується у сучасній ігровій індустрії, завдяки якому ігри можуть створювати межу між реальним та віртуальним світом.

Поліпшена технологія анімації

Покащені анімаційні технології, такі як захоплення руху та динамічна анімація, застосовуються у багатьох сучасних іграх, за допомогою яких розробники можуть створювати більш захоплюючих персонажів та оточення, які пожвавлюють гру.

Віртуальна та доповнена реальність

В розважальній індустрії, віртуальна і доповнена реальність мають величезний потенціал для революції ігрового світу. Для покращення ігрового процесу використовується доповнена реальність, яка з розвитком споживчих VR-гарнітур у сфері може спостерігатись великий попит на використання цих нових технологій у розробці ігор.

До основні векторів розвитку на сьогоднішній день, можна віднести:

Графіка в іграх

- Реалістичні текстури – це ключовий елемент для того, щоб візуальний досвід гравців покращувався, розробники прагнуть створити більш деталізовані та реалістичні структури ігор.

- Трасування променів – ця технологія допомагає створювати більш реалістичне освітлення, відображення та тіні, дозволяє ігровим двигунам моделювати шлях світла у грі.
- HDR (High Dynamic Range) – дає покращення якості зображення, розширюючи динамічний діапазон, забезпечуючи яскравіші світлі ділянки та глибокі тіні.

VR та AR в ігровій індустрії

- Віртуальна реальність (VR) – це технологія, де гравці можуть взаємодіяти з навколишнім світом за допомогою спеціальних приладів, які занурюють у віртуальний світ.
- Доповнена реальність (AR) – відкриває нові можливості для взаємодії у реальному світі та озволяє інтегрувати віртуальні об'єкти в реальне оточення.
- Візуальний досвід гравців збагачується за допомогою зміни способу гри у VR та AR. Цікавий досвід для гравців та нові можливості для розробників надають ці технології, що робить їх все більш значущими в ігровій індустрії.

Ігри-сервіси

- Модель передплати – за фіксовану щомісячну плату, можна отримати платну підписку на велику кількість ігор, ця система стає все більш популярною серед користувачів.
- Онлайн-сервіси – такі як стрімінг, мультиплеєра та постійне оновлення контенту – всі ці можливості відкриває постійне з'єднання з інтернетом.
- Хмари для ігор – призначаються для того, щоб зберегти прогрес, дані у хмарі та грати ігри на різних пристроях дозволяє використання хмарних сервісів.
- Для користувачів та розробників ігри-сервіси змінюють модель споживання ігор. Вони роблять їх більш гнучкими, зручними та доступними, а також надають нові способи надання ігрового контенту та монетизації.

AI в іграх

- Розумні NPC (непротилежні персонажі) – створивши реалістичне та захоплююче ігрове оточення, штучний інтелект та просунуті алгоритми дозволяють NPC реагувати на дії гравців.
- Адаптивний геймплей – беручи до уваги унікальний стиль гри, дії гравця, рівень складності та подачу завдань, AI автоматично адаптується та сприяє створенню ігрового досвіду.

Blockchain в іграх

- Віртуальні активи – унікальні віртуальні активи, можуть бути обмінені чи використані гравцями в різних іграх, за допомогою технології блокчейну.
- Безпека – цифрові активи та транзакції всередині ігор та за їх межами захищені з допомогою блокчейну.
- Нові можливості для гравців – створення унікальних предметів та можливостей, а також більший контроль над власними даними надає блокчейн.

Вплив хмарних технологій на ігрову промисловість

- Хмарні ігри – вони покращують доступність та гнучкість ігрового досвіду, дозволяють транслювати вміст ігор на пристрої гравців та запускати ігри на віддалених серверах.
- Зберігання даних і розраховані на багато користувачів ігри – підтримувати стабільне з'єднання та синхронізацію даних гравців, забезпечити легкий доступ до ігор, які розраховані на велику кількість користувачів – всі ці задачі спрощують хмарні технології.
- Покращений гейм-дизайн – більша обчислювальна потужність, яку забезпечує хмарне обчислення, надає можливість розробникам створювати ігри з більш складним геймплеєм.

Покращити безпеку, доступність та якість ігрового досвіду, а також надати нові можливості для гравців та розробників зможе: AI, блокчейн та хмарні технології.

Основні значення інновацій та адаптації до нових технологій для успіху в індустрії ігор, можна віднести наступне:

- Гнучкість і адаптація. Потреби гравців та нові технології швидко змінюються, тому розробники повинні бути гнучкими та вчасно реагувати на зміни та потреби.
- Інновації та унікальний досвід. Ігрові компанії, що інвестують в інновації, можуть створювати унікальні ігри, здатні привернути увагу та утримувати аудиторію.
- Дослідження та розробка. Головним аспектом успішності та передування іншим компаніям є регулярне відслідковування нових тенденцій та технологій, що допомагає компаніям залишатися попереду конкуренції.

Адаптація ігрової індустрії, інновації та постійне дослідження ігрової індустрії є ключовими факторами для успіху в індустрії ігор, що дозволяє компаніям створювати захоплюючі передові ігри для широкої аудиторії.

На мою думку, в майбутньому, нові технології, такі як VR, AR та хмарні обчислення, змінять ігрову індустрію, надавши їй більш цікавий та персоналізований досвід для гравців та збагативши її новими можливостями. За новими тенденціями та інноваційними технологіями створюються сучасні ігри. Від фотореалізму до віртуальної та доповненої реальності – індустрія постійно розширює межі можливого. З розвитком та вдосконаленням апаратних забезпечень, можливості для створення ігор стають все більш безмежними. Майбутнє розробки ігор, з урахуванням усіх тенденцій та інновацій виглядає яскравішим, ніж будь-коли. Можна підкреслити основні висновки про сучасні технології та тренди у розробці ігор, а саме:

- Диверсифікація платформ – враховуючи всі пристрої від смартфонів до VR-систем, розробники повинні звернути увагу на специфіку кожного з цих пристроїв та адаптувати свої ігри до різних платформ.
- Технології VR/AR – впровадження віртуальної та доповненої реальності надає можливості для захоплюючого та унікального ігрового досвіду.

- Хмарні технології – хмарні обчислення дозволяють грати на різних пристроях без втрати якості.

1.2. Роль поведінкових моделей гравців у вдосконаленні ігрового процесу

1.2.1. Поведінкові моделі гравців як ключовий елемент гри

Моделювання поєдинки гравців у сучасних іграх залежить не лише від спроб відтворити реалізму, але й від забезпечення найвищого рівня задоволення та залучення гравців. Цей аспект стає ключовим елементом гри, оскільки дозволяє персоналізувати вашу взаємодію з гравцями, забезпечуючи унікальний ігровий процес та захоплюючий досвід.

Ключові особливості моделей поведінки гравців. Модель поведінки гравця включає багато аспектів, таких як інтелект штучного ігрового агента, механізм прийняття рішень, адаптивність до стратегій гравця та реакція на зміни в грі. Однією з ключових особливостей є можливість адаптуватися до стилю гри кожного гравця, роблячи взаємодію більш індивідуальною.

Моделюйте особистість і мотивацію гравця. Ефективні моделі поведінки включають аспекти особистості та мотивації гравця. Здатність гравця відчувати себе залученим у гру, долати труднощі та отримувати винагороду за досягнення залежить від якості вбудованої механіки, яка взаємодіє з моделлю поведінки.

Роль соціальної взаємодії в моделях поведінки. Важливою частиною моделі поведінки є їх взаємодія з іншими гравцями. Соціальні взаємодії, такі як спілкування та спільне вирішення проблем, можуть формувати розвиток гри та впливати на стратегії гравців.

Технічні проблеми та інновації в реалізації моделі поведінки. Розробка ефективних моделей поведінки стикається з технічними проблемами, такими як оптимізація алгоритмів і використання штучного інтелекту. Інновації в цьому

напрямку можуть включати використання машинного навчання для адаптації до унікальних стилів гри гравців і реалістичного моделювання поведінки.

Моделі поведінки гравців є ключовим аспектом сучасних ігор, визначаючи якість ігрового процесу та занурення гравця. Їхній вплив на створення гри та взаємодію з гравцями стає критично важливим компонентом успішної розробки.

1.2.2. Вплив поведінкових моделей на геймплей

Вплив моделей поведінки на процес гри стає важливою частиною розробки гри. Вони формують динаміку гри, роблять її веселою та складною. На цьому тлі ми розглянемо вплив моделей поведінки на ключові аспекти геймплею.

Диференціація гравців – це так звані моделі поведінки гравців пропонуються їх унікальністю, результатом чого є індивідуальні характеристики та стратегії. Це дозволяє створити різноманітність серед гравців і диференціювати ігровий процес, даючи кожному гравцеві унікальний досвід.

Ігровий баланс – ефективні поведінкові моделі взаємодіють із ігровою системою, щоб забезпечити баланс між різними рівнями складності. Це важливо, щоб гра залишалася складною та доступною для гравців усіх рівнів кваліфікації.

Сюжет і динамічний розвиток – моделі поведінки гравців можуть впливати на розвиток ігрового сюжету та динаміку подій. Взаємодія гравця з навколишнім світом може визначати хід подій і відкривати нові можливості для розвитку сюжету.

Взаємодія гравців у багатокористувацьких іграх – у багатокористувацьких іграх моделі поведінки гравців стають ще важливішими, оскільки вони вирішують стратегії та тактики в реальному часі. Емоційна взаємодія та спілкування гравців стають ключовими факторами, що визначають динаміку гри.

1.3. Ігрові двигуни та їхнє використання в Java-застосунках

1.3.1. Огляд ігрових движків та їх функції

Ігрові движки є ключовим елементом сучасної розробки ігор, що дозволяє розробникам створювати якісні та ефективні ігри. Ці механізми надають розробникам інструменти для реалізації графічних ефектів, фізики, аудіо, обробки вхідних даних та інших функцій, необхідних для створення ігор.

Ігровий движок, простіше кажучи, — це набір програмного забезпечення, необхідного для запуску комп'ютерних ігор і не пов'язаний строго з жодною грою. Тобто його двигун відрізняється від самої гри, і на його основі можна робити інші ігри.

Однак це не завжди так, і в деяких випадках движок, створений для гри, насправді невіддільний від неї. Так часто траплялося на початку ігрової індустрії, коли для кожної консолі випускалися окремі версії, які могли істотно відрізнятися від оригіналу. Так сталося, наприклад, з популярною в Україні та інших країнах грою Aladdin, версії якої для Super Nintendo і SEGA Mega Drive не були схожі один на одного.



Рис. 1.1. Скріншот з гри Aladdin, з лівого боку Super Nintendo, з правого боку SEGA Mega Drive

Однак зазвичай ігровий движок складається з набору менших ігрових движків, які реалізують більш специфічну функціональність.

Види ігрових движунів

Ігрові движки багато в чому схожі, хоча ви все одно можете спробувати класифікувати їх.

По-перше, існує два підходи, які визначають основний спосіб розробки ігор розробниками:

- З окремим редактором. Багато ігрових движків дозволяють користувачам створювати щось, не вдаючись до навичок програмування, а за допомогою зручного редактора, схожого на Photoshop або програми для редагування відео. Як правило, такі движуни максимально самодостатні і не вимагають підключення додаткових модулів, але їх функціональність може бути обмежена. Але сьогодні навіть найдосконаліші ігрові движки дозволяють працювати через редактор.
- У другому випадку движок передбачає, що розробники ігор пишуть програмний код безпосередньо. Такі рішення були запущені через середовище розробки, яке зазвичай використовують програмісти, і є не стільки готовим продуктом для створення ігор, скільки набором раніше написаних програм, які можна адаптувати до чогось нового.

Крім того, спосіб розповсюдження механізму також може відрізнятись, оскільки, як і будь-яке інше програмне забезпечення, розробники програмного забезпечення зазвичай хочуть отримати винагороду. Однак це не завжди так:

- Двигуни з відкритим кодом — це не безкоштовні версії певної технології, а зрілі інструменти розробки, створені ентузіастами. Такі рішення можна використовувати в необмеженій кількості, на них можна створювати комерційні проекти без оплати творця движка. Godot є прикладом такого движуна.
- Безкоштовні механізми можуть не надавати кінцевим користувачам весь вихідний код, але вони вільні для використання. У більшості випадків це означає, що розробник повинен заплатити частину прибутку від продажу

гри, якщо такий прибуток є. Зараз за такою схемою поширюються більшість популярних сторонніх ігрових движків – Unreal Engine, Unity, CryEngine та ін.

Повністю платні двигуни залишилися в минулому, і сьогодні ця категорія, швидше за все, відноситься до програмних платформ, створених безпосередньо великими групами розробників для своїх ігор. Через високу трудомісткість і вартість цього процесу можуть собі це дозволити лише найуспішніші команди – наприклад, двигун Frostbite в серії Battlefield і RAGE для GTA і RDR.

Однак не завжди можна розглядати ігрову платформу як універсальний движок: часто трапляється, що технічне оснащення гри складається з багатьох технологій, кожна з яких не є повноцінним движком, але виконує важливі функції.

Основні функції ігрових движків включають графіку та візуалізацію, що забезпечує відображення графічних об'єктів і реалізацію світла, тіней, текстур та інших графічних ефектів. Функція фізики імітує реальну фізику об'єктів у грі, дозволяючи їм взаємодіяти один з одним і реагувати на дії гравця. Звукова функція відповідає за управління звуковими ефектами, відтворенням музики та звукових елементів у грі. Функція введення обробляє вхідні дані від програвача, такі як клавіші, миша, джойстики та інші пристрої. Штучний інтелект використовується для реалізації алгоритмів штучного інтелекту, які контролюють поведінку неігрових персонажів і приймають рішення в грі.

Огляд основних двигунів (game engines)

Unity

Unity – це один з дуже популярних двигунів. Він не обмежує платформу та жанр користувача, але історично використання його функцій було зручним для створення ігор на мобільних платформах, а також невеликих проектів на ПК та консолях, але не дуже підходить для масштабних ігор.

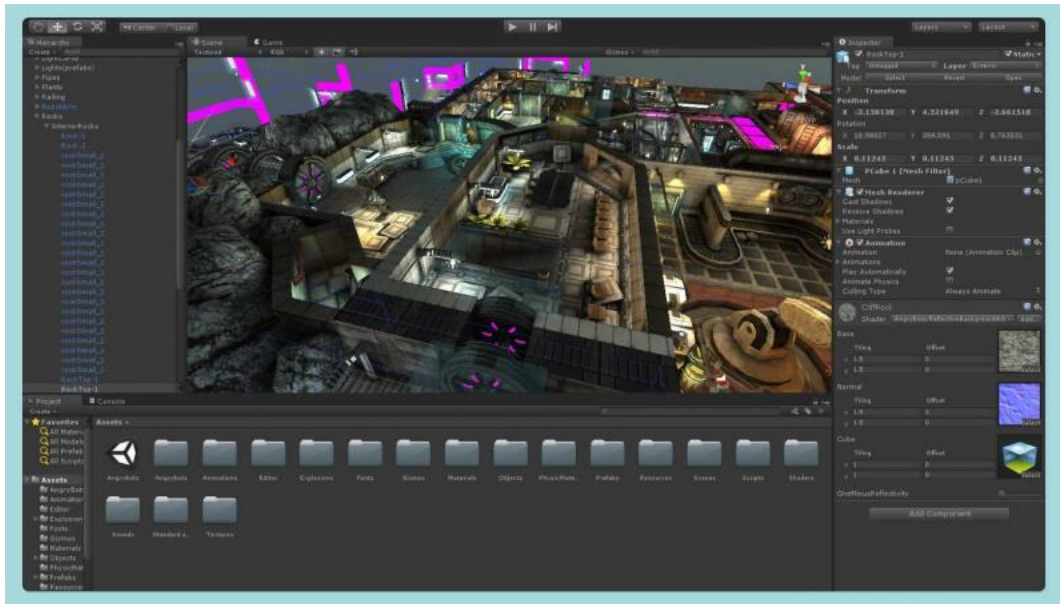


Рис. 1.2. Середовище розробки Unity

Unity – потужний інструмент, який має багато переваг. Гнучкий та розширюваний двигун із безліччю компонентів спрощує процес розробки. Велика кількість шаблонів, прикладів, уроків і курсів дозволяє швидко освоювати платформу. Обширна база знань і велика спільнота забезпечують підтримку та обмін ідеями. Простота у розробці завдяки мовам програмування C# та Bolt робить Unity досить доступним навіть для початківців. Велика кількість успішних кейсів свідчить про потенціал платформи. Крім того, Unity – крос-платформений інструмент, що робить його вибором для розробки для різних платформ.

Незважаючи на свої плюси, Unity також має свої недоліки. Для створення високоякісної гри потрібно глибоко зануритися в аспекти двигуна. Багато сирих версій і велика кількість багів можуть стати викликом для розробників. Відсутність відкритого коду для невеликих команд розробників обмежує можливості налаштувань. Величезний розмір Unity може становити проблему для обмежених ресурсів. Також важливо враховувати, що пріоритет розробки оновлень двигуна скерований на мобільні платформи, що може вплинути на інші аспекти. Крім того, орієнтація моделі підтримки на бізнес, а не на зручність для розробників, може впливати на сприйняття платформи.

Unreal Engine

Unreal Engine — це не просто серія відомих ігрових движків, а фактично лідер у створенні високобюджетних пригодницьких ігор. Двигун не накладає обмежень на платформи, на яких ви можете публікувати свою гру. Unreal Engine створювався в першу чергу для надзвичайно красивих ігор на консолі PlayStation і Xbox, а також потужних ПК, движки все ще використовуються для створення мобільних ігор для пристроїв Android і iOS.



Рис. 1.3. Середовище розробки Unreal Engine

Відомий своїми вражаючими графічними можливостями, підходить для створення високоякісних ігор із вдосконаленою графікою, має широкий набір інструментів розробки, а також інструменти графічної обробки та анімації, в основному використання у великих проектах, часто вибирається для створення великих проектів.

Unreal Engine – вражаючий інструмент з численними перевагами. Потужний редактор надає всі необхідні засоби для створення ігор. Гнучка архітектура ігрового движку дозволяє розробникам легко розширювати функціональність. Ігровий движок, розроблений для творчих ігор, націлений переважно на розробників, а не на бізнес, відзначається своєю спрямованістю на творчість. Unreal Engine готовий до створення AAA-проектів із самого початку,

що полегшує розробку великих та складних ігор. Крім того, він крос-платформений, що дозволяє створювати гри для різних платформ.

Незважаючи на свої переваги, Unreal Engine має свої недоліки. Вищий поріг входу може ускладнити для новачків швидке освоєння інструменту. Закрита і менш чисельна спільнота в порівнянні з іншими інструментами може обмежувати обмін ідеями та підтримку. Акцент на AAA-проектах робить Unreal Engine менш доступним для менших розробників чи невеликих студій. Розмір двигуна та його вимогливість можуть виявитися проблемними для обмежених за ресурсами проектів.

CryEngine

CryEngine – це ігровий двигун, розроблений компанією Crytek. Він призначений для створення високоякісних графічно ігор, і вперше був використаний у грі "Far Cry". CryEngine визначається своєю потужною графікою, реалістичними ефектами та розширеними можливостями для розробки великих та відкритих ігрових світів. Цей ігровий двигун широко використовується в індустрії відеоігор для створення вражаючих ігор з високим рівнем деталізації.

Відомий своєю потужною графікою та реалістичним рендерингом. Загалом використання в ігровій індустрії, зазвичай для створення ігор з унікальним геймплеєм та вражаючою графікою.

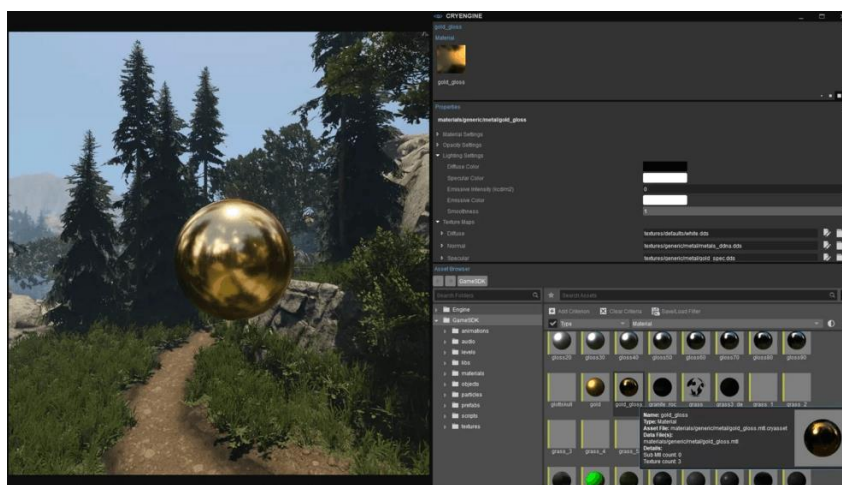


Рис. 1.4. Середовище розробки CryEngine

CryEngine – це вражаючий ігровий двигун, який відзначається своїм потужним редактором-пісочницею та величезним потенціалом для створення AAA проєктів. Зокрема, він славиться своїм високопродуктивним рендером та добре структурованою документацією. Проте, серед його недоліків можна виділити сильний акцент на шутерних елементах, не завжди зручний інструментарій для збірки та вимогу до серйозних знань в мові програмування C++.

Вибір движка та мови програмування залежить від потреб проєкту, рівня досвіду розробника та технічних можливостей, необхідних для створення конкретної гри.

1.3.2. Ігрові двигуни для Java-застосунків

У світі Java є кілька потужних ігрових движків, які дозволяють розробникам створювати ігри за допомогою цієї мови програмування. Деякі з найпопулярніших ігрових движків Java включають:



Рис. 1.5. Лого libGDX

LibGDX – це потужний та відкритий ігровий двигун, який підтримує розробку для різних платформ, включаючи Android, iOS, Desktop та HTML5. LibGDX надає широку функціональність для графіки, фізики, аудіо та інших аспектів гри.



Рис. 1.6. Демо розробки карткової 2D гри

Бібліотека є відкритою та крос-платформною, створені програми можуть бути запущені на Desktop, на Android та iOS – і навіть у браузері (з використанням HTML5). Розробка та підтримка рішення триває вже понад 10 років. В офіційному пакеті знаходиться різноманітність інструментів для всебічної розробки ігор, включаючи ресурси для роботи з 2D-графікою для створення привабливих візуальних елементів, можливості для роботи з 3D-графікою для надання тривимірного вигляду ігрового оточення, вбудований фізичний двигун, що враховує зіткнення об'єктів та створює реалістичну фізичну модель, функціонал для роботи з аудіо-потокком для впровадження звукових ефектів та музики, опції для обробки відео для розширення можливостей візуального контенту, функції взаємодії з різними контролерами, такими як клавіатура, миша, гейм-пади та інші, мережевий стек для написання ігор з високою ефективністю передачі даних, а також підтримка алгоритмів штучного інтелекту для створення ігор із комп'ютерним суперником, що володіють обдуманим та адаптивним геймплеєм.

Окрім офіційних пакетів, за потреби до бібліотеки додаються розширення від сторонніх розробників. Наприклад, демо плагіна VisUI для проектування віконних інтерфейсів.

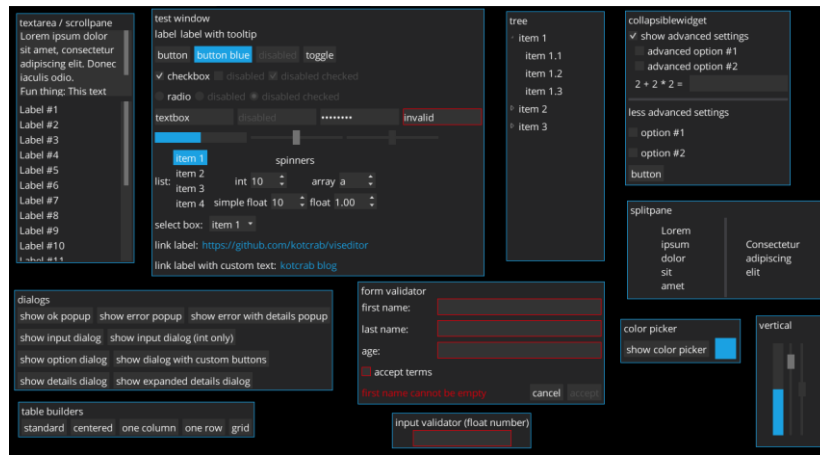


Рис. 1.7. Інтерфейс плагіна VisUI

Можна перерахувати ряд наступних переваг, таких як:

Сильна сумісність. Libgdx сумісний із більшістю платформ і реалізований за допомогою стандартного JavaSE. Він може працювати на Mac, Linux, Windows та інших системах, а також на платформах Android (можна використовувати з Android 1.5 і вище, а також з Android 2.1 і вище). У той же час надзвичайно висока сумісність забезпечує зручність налагодження та розробки. Ви можете використовувати той самий код на Android, щоб писати, тестувати та налагоджувати свою програму на настільному ПК, видаляючи загальні функції (тобто ви можете налагоджувати ігрову програму, написану за допомогою механізму Libgdx, не відкриваючи емулятор Android). Програми Linux і Android. Загальний процес розробки полягає в тому, щоб якомога довше залишатися на робочому столі та періодично перевіряти, чи може поточний код працювати на Android.

Висока ефективність. Libgdx — це фреймворк для розробки ігор, написаний головним чином мовою Java із додаванням коду C/C++, призначений для виконання деяких високопродуктивних операцій, таких як фізичні механізми чи обробка звуку. Як користувачу, вам потрібно лише звернути увагу на сторону Java, яка інкапсулювала весь власний код. Порівняно з іншими ігровими движками Android, переваги libgdx у продуктивності дуже очевидні.

Чітка структура. Libgdx — це користувач аудіо, файлів, графіки, математики, фізики, сцен, утиліт. Ці основні бібліотеки складаються зі звукових

операцій, читання файлів, 2D/3D візуалізації, операцій малювання Libgdx, упаковки Vox2D, 2D/3D ігрових компонентів і вбудованих Libgdx інструментів Libgdx.

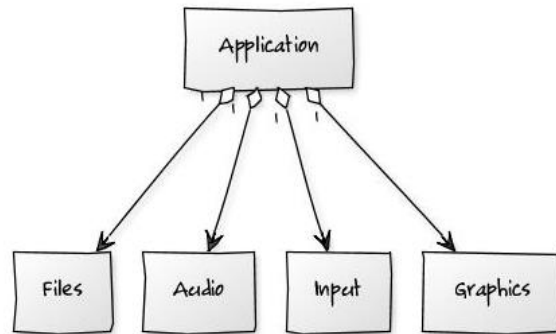


Рис. 1.8. Організаційний модуль libgdx

Упаковка Vox2D. Упаковка фізичного механізму Libgdx також вражає, використання jni для інкапсуляції сторони C++ vox2d є досить божевільним, що робить ефективність його роботи набагато швидшою, ніж інші фізичні механізми такого ж рівня, такі як jbox2d. Деякі популярні ігрові движки Android, які включають фізичні движки (такі як Andengine, Rokon тощо), активно використовують libgdx. Якщо ваша гра (особливо для Android) використовуватиме фізичний движок, надайте пріоритет Libgdx.

Багаті інструменти. Libgdx також має велику кількість інструментів розробки, візьміть як приклад редактор частинок, генератор шрифтів Hierobitmap, пакувальник текстур і редактор макета TWL.

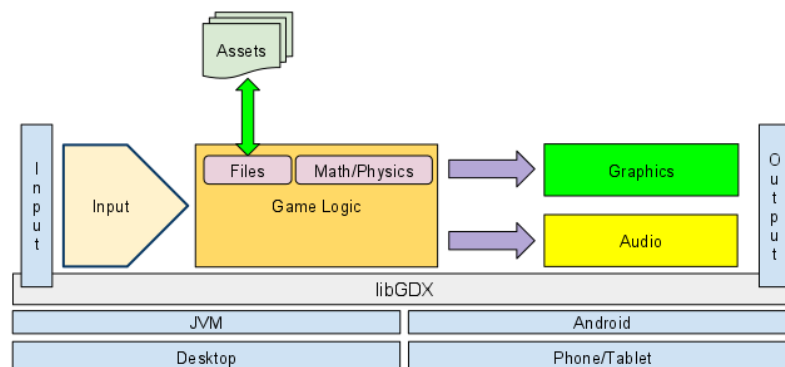


Рис. 1.9. Архітектура ігрового дизайну

Не дивлячись на всі переваги, на жаль також є неділіки:

Різниця в ефективності. Libgdx також не ідеальний, його мета — забезпечити єдину архітектуру для розробки ігор, яка забезпечує однакову поведінку на різних платформах. На жаль, цієї високої мети важко досягти. Найбільша різниця між настільними комп'ютерами та пристроями Android полягає в тому, що пристрої Android зазвичай набагато повільніші, ніж настільні комп'ютери. Це означає, що ваша програма працює дуже швидко на ПК, але працює з швидкістю равлика на Android. Необхідно розуміти безліч правил, яких можна і не можна дотримуватися. Розробники повинні дотримуватися цих правил, щоб програма могла нормально працювати навіть на слабшому Android.

Обмеження ресурсу зображення. Libgdx має власний декодер зображень, за допомогою Pixmap, який вони надають, операції візуалізації на рівні пікселів можуть виконуватися на вказаному зображенні, тому воно не покладається на Bitmap Android для завантаження та обробки зображень, але наразі підтримує лише png. формати зображень jpg і bmp. По-друге, Libgdx вимагає, щоб ширина та висота зображення, що використовується в грі, були цілим ступенем 2, інакше буде створено виняток Gdx і поведінка завантаження не буде дозволена (ширина та висота текстури мають бути ступенем 2).

JMonkeyEngine



Рис. 1.10. Лого JMonkeyEngine

JMonkeyEngine – це високопродуктивний механізм для створення 3D-ігор на Java. Він має велику спільноту розробників і підтримує різноманітні функції, включаючи шейдери, фізичне моделювання та обробку вхідних даних. Дозволяє повністю реалізувати свій потенціал і створювати неймовірні ігрові світи. Фреймворк побудовано на мові програмування Java, що робить його доступним для широкого кола розробників. Завдяки своїй гнучкості та простоті використання JMonkeyEngine дозволяє вам зосередитися на створенні ігрового вмісту, не відволікаючись на складність програмування.

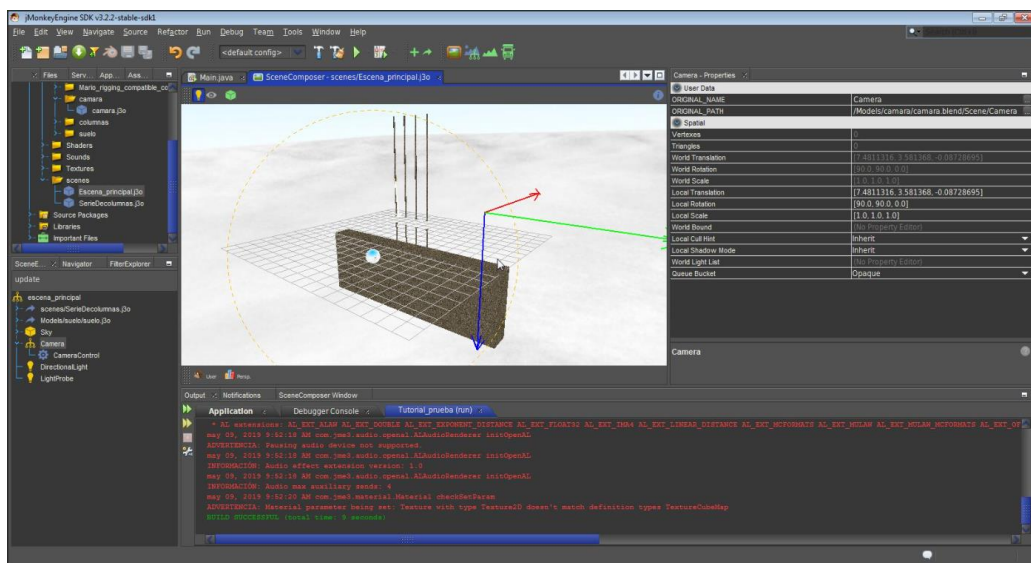


Рис. 1.11. Середовище розробки JMonkeyEngine

Однією з головних переваг JMonkeyEngine є його відкритий код. Це дозволяє розробникам вносити будь-які зміни в код двигуна відповідно до своїх потреб і створювати унікальний ігровий досвід. Як наслідок, JMonkeyEngine має активну спільноту розробників, які вносять нові функції та виправляють помилки.

JMonkeyEngine надає широкий набір інструментів для моделювання та анімації об'єктів, створення ефектів освітлення та тіні та маніпулювання звуками. Розробники можуть створювати та імпортувати 3D-моделі у форматах OBJ, FBX, Collada та інших.

Двигун має потужний редактор сцен, який дозволяє створювати складні ігрові світи з різноманітними об'єктами, сценами та ефектами. Редактор надає можливості для налаштування освітлення, матеріалів, трансформації об'єктів тощо.

Однією з головних особливостей JMonkeyEngine є його інтеграція з фізичним двигуном Bullet. Це дозволяє створювати реалістичні фізичні ефекти, такі як зіткнення об'єктів, гравітація, тертя та пружність. Фізичне моделювання в JMonkeyEngine дуже точне та продуктивне.

Двигун також має вбудовану систему управління звуком, яка дозволяє відтворювати музику та звукові ефекти в грі. Розробники можуть легко налаштувати джерело звуку, його розташування в просторі та інші параметри.

JMonkeyEngine ідеально підходить для розробки ігор на різних платформах, включаючи Windows, MacOS, Linux і Android. Він надає можливість експортувати гру у файл або запускати її безпосередньо в середовищі розробки.

Поєднання всіх цих функцій робить JMonkeyEngine чудовим вибором для розробників ігор, які хочуть створювати професійні ігри з реалістичною 3D-графікою та фізикою.

PlayN

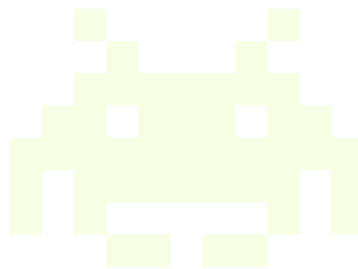


Рис. 1.12. Лого PlayN

PlayN – це абстракція для розробки ігор, яка дозволяє вам один раз написати код і запускати гру на різних платформах, таких як HTML, Android, iOS тощо. PlayN використовує Java та інші мови програмування. Потужна

міжплатформна бібліотека, яка дозволяє розробникам створювати ігри за допомогою єдиного коду та запускати їх на різних платформах, включаючи Android, HTML5, Flash тощо. PlayN пропонує високу продуктивність і простоту використання завдяки використанню Java як мови розробки.

Основні принципи, на яких ґрунтується PlayN, включають мультиплатформність, використання мови Java, графічний двигун та обробка подій та менеджмент ресурсів.

PlayN дозволяє створювати ігри, які можна запускати на різноманітних платформах, включаючи настільні комп'ютери, мобільні пристрої, браузері тощо. Це досягається за рахунок використання абстракцій, які дозволяють розробникам один раз написати код і запустити його на різних пристроях. PlayN використовує мову програмування Java, одну з найпопулярніших мов в індустрії розробки ігор. Це дозволяє розробникам використовувати безліч готових бібліотек і інструментів, спрощуючи процес створення ігор. PlayN містить потужний графічний механізм, який дозволяє відображати графіку та анімацію. Він також обробляє події користувача, такі як натискання кнопок і рухи миші, забезпечуючи інтерактивні ігрові сценарії. PlayN забезпечує зручний спосіб керування ігровими ресурсами, такими як зображення, звуки та шрифти. Розробники можуть легко імпортувати та використовувати ці ресурси у своїх іграх, спрощуючи розробку та обслуговування.

Структура ігрового движка PlayN включає наступні основні елементи. Класи роботи з графікою та анімацією включають функції, такі як малювання спрайтів, створення анімації, обробка текстур тощо. Обробники подій відповідають за обробку дій користувача, таких як натискання кнопок або рухи миші. Модуль управління ігровими ресурсами забезпечує завантаження та зберігання зображень, звуків та інших ресурсів. Фізична система відповідає за створення реалістичної взаємодії між об'єктами в грі. Інструменти для створення ігрових сцен, переходів між сценами та керування ходом гри сприяють ефективному розвитку ігор. Усі ці елементи взаємодіють, забезпечуючи

розробникам необхідні інструменти для створення високоякісних та захоплюючих Java-ігор за допомогою ігрового механізму PlayN.

Ці ігрові движки надають розробникам Java широкий набір інструментів для створення високоякісних та захоплюючих ігор, розширюючи можливості розробки ігор цією мовою.

1.4. Огляд інструментів та технологій для розробки Java-ігор

1.4.1. Вибір мови програмування для розробки

Обґрунтування вибору мови програмування Java

Одним із основних аспектів, який вплинув на мій вибір мови програмування для написання ігор, була універсальність і портативність Java. Java відома своїм слоганом «Write Once, Run Anywhere» (пиши один раз, запускай скрізь). Це означає, що якщо я розробляю гру на Java, вона працюватиме на будь-якому пристрої, який підтримує віртуальну машину Java (JVM). Це дозволяє розробникам не тільки економити час, але й створювати продукти, доступні широкому колу користувачів.

Ще однією перевагою Java є широкий вибір бібліотек і фреймворків, які полегшують процес розробки. Зокрема, багато ігрових фреймворків і бібліотек, таких як libGDX і jMonkeyEngine, розроблені спеціально для створення Java-ігор. Це дозволяє пришвидшити процес розробки та зосередитися на самому геймплеї та інших аспектах гри, а не на повсякденній роботі на рівні системного виклику.

Java також відома своєю великою та активною спільнотою розробників. Це означає, що завжди є багато ресурсів, документації та форумів, де ви можете знайти відповіді на свої запитання та допомогти вирішити проблеми. Спільнота також активно розвиває мову та надає нові функції, що робить Java сучасною та конкурентоспроможною мовою програмування.

Обґрунтування вибору JavaFX для розробки графічного інтерфейсу гри

JavaFX — це сучасна структура для розробки графічних інтерфейсів на Java. Є також вагомі причини для вибору JavaFX для створення графічної частини вашої гри.

По-перше, JavaFX надає потужні інструменти для створення красивих інтерактивних інтерфейсів. Він підтримує сучасні ефекти, анімацію та стилі, що дозволяє створювати привабливі та зручні ігрові інтерфейси. Зокрема, використання JavaFX дозволяє реалізувати візуальні ефекти, які покращують ваш ігровий досвід.

По-друге, інтеграція JavaFX з мовою програмування Java незаперечна. Це дозволяє вам легко взаємодіяти з логікою гри, написаною на Java, і відобразити зміни на екрані. Багато функцій JavaFX, таких як події та анімація, можуть бути легко інтегровані та дозволяють створювати інтуїтивно зрозумілі та ефективні інтерфейси.

Підсумовуючи, варто зазначити, що JavaFX постійно оновлюється та розвивається, що гарантує сучасні стандарти розробки та дозволяє вам скористатися новими можливостями для покращення графічної частини вашої гри.

1.4.2. Використані бібліотеки та інструменти

Використання фреймворку libGDX

Переносимість та кросплатформенність – однією з основних причин вибору фреймворку libGDX для розробки ігор є його висока портативність. Завдяки цій функції розробка виконується один раз, і гра може працювати на різних платформах, таких як Android, iOS, комп'ютер і веб-браузери. Це значно спрощує розгортання ігор на різних пристроях і робить libGDX привабливим варіантом для тих, хто націлений на широку аудиторію.

Широкий функціонал – LibGDX надає розробникам доступ до широкого спектру інструментів і функцій. Це включає обробку вхідних даних, аудіо, графіку, анімацію та фізику. Цей різноманітний набір функцій дозволяє

ефективно створювати високоякісні ігри, а розробники можуть зосередитися на ключових аспектах, таких як геймплей.

Активна спільнота – LibGDX має велику та активну спільноту розробників. Це важливо для отримання підтримки, вирішення труднощів і обміну досвідом. Активна спільнота може швидко вирішити проблеми та визначити найкращий спосіб впровадження певних функцій.

Використання технології Spine

Гнучкість анімацій – для створення анімації була обрана технологія Spine завдяки її гнучкості та використанню скелетної системи. На відміну від традиційних методів кадрювання, Spine дозволяє створювати плавні та реалістичні рухи персонажів. Ця гнучкість є важливою частиною розробки ігор, особливо коли йдеться про створення яскравих і яскравих анімацій.

Ефективне використання ресурсів – Spine дозволяє зберігати анімацію в оптимізованому форматі, що допомагає зменшити кількість файлів і ресурсів, необхідних для відтворення анімації. Це особливо важливо для мобільних платформ, де обмежені ресурси можуть вплинути на продуктивність гри та швидкість завантаження.

Інтеграція з libGDX – важливим аспектом вибору технології Spine була її проста інтеграція з libGDX. Це робить простим і ефективним використання анімацій, створених у Spine, у ваших проектах libGDX. Така синергія сприяє якісному візуальному втіленню ідей розробників і створенню ефективних і динамічних ігрових світів.

Інтеграція обраних технологій

Розробник, об'єднавши libGDX та Spine, може отримати потужний інструментарій для створення високоякісних та кросплатформених ігор. Поєднання libGDX та гнучких анімацій Spine дозволяють забезпечити якісний ігровий досвід на різних пристроях. Також, зручна інтеграція та оптимізоване використання ресурсів в розробці ігор, забезпечують ефективність і продуктивність. Це не тільки значно спрощує роботу розробника, а також сприяє

створенню ігор високої якості з ефективним використанням технічних можливостей.

Висновки до розділу 1

В галузі розробки ігор, яка є динамічною та інноваційною, відбуваються значущі зміни, орієнтовані на актуальні тренди та новітні технології. Розвиток VR-технологій, штучного інтелекту та хмарних технологій обіцяє зробити майбутнє розробки ігор ще більш багатостороннім та захоплюючим.

У сучасних умовах розробки ігор важливі такі аспекти, як диверсифікація платформ, технології VR/AR та хмарні технології. Розробники повинні уважно враховувати різноманітність пристроїв, використовуючи їхні особливості та адаптуючи гри до різних платформ. Впровадження віртуальної та доповненої реальності надає можливості для новаторського ігрового досвіду, а хмарні технології дозволяють грати на різних пристроях без втрати якості.

Моделі поведінки гравців є ключовим елементом успішної розробки ігор, визначаючи якість геймплею та взаємодію з гравцями. В багатокористувацьких іграх, особливо, емоційна взаємодія та спілкування гравців стають визначальними факторами, що впливають на динаміку гри.

Вибір ігрового движка та мови програмування залежить від потреб проекту, рівня досвіду розробника та технічних вимог. Java, зокрема, є популярним вибором для розробки ігор, забезпечуючи розробникам широкий набір інструментів та активну спільноту. JavaFX, з інноваційним підходом та постійним оновленням, визначає нові стандарти розробки ігор.

За використанням libGDX та Spine розробники отримують потужний інструментарій для створення високоякісних і кросплатформених ігор. Поєднання цих інструментів дозволяє ефективно розвивати ігри, забезпечуючи якість і зручність у використанні.

РОЗДІЛ 2

РОЗРОБКА JAVA-ЗАСТОСУНКУ ТА МОДЕЛІ ПОВЕДІНКИ ГРАВЦІВ

2.1. Архітектура програми та її компоненти

2.1.1. Огляд архітектури програми

Ключовим етапом при розробці кожного продукту є архітектура програми, тому що вона визначає структуру, взаємодію та функціональність системи. Цей процес визначає, як різні компоненти програми будуть взаємодіють один з одним, забезпечуючи ефективну та оптимальну роботу програми.

Ефективно розроблена архітектура дозволяє розробникам легко розширювати та змінювати програму в майбутньому, простіше кажучи робити будь-які маніпуляції не порушивши функціонал та працездатність, роблячи її більш гнучкою та готовою до майбутніх змін. У порівнянні з хаотичними підходами, архітектура системи забезпечує стабільність, надійність і простий в управлінні код під час розробки.

Наш продукт складається із трьох основних компонентів:

- сервер;
- математика;
- графічна частина;

<i>Кафедра КІТ (47)</i>				<i>НАУ 23.05.85.000 ПЗ</i>			
<i>Виконав</i>	<i>Горічко А.О.</i>			РОЗРОБКА JAVA- ЗАСТОСУНКУ ТА МОДЕЛІ ПОВЕДІНКИ ГРАВЦІВ	<i>Літ.</i>	<i>Арк.</i>	<i>Аркушів</i>
<i>Керівник</i>	<i>Воронін А.М.</i>					39	14
<i>Консульт.</i>					УС-211М 122		
<i>Н. Контр.</i>	<i>Райчев І.Е.</i>						

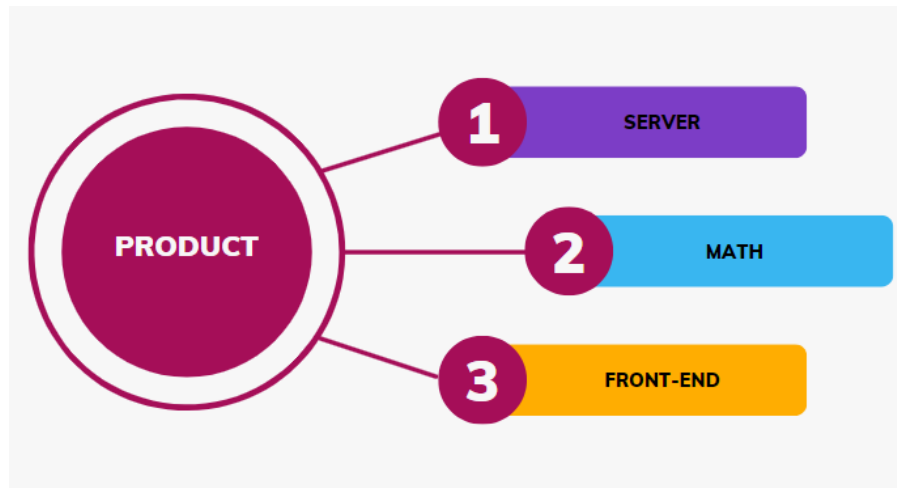


Рис. 2.1. Схема продукту

Дослідимо кожен компонент докладніше, розкривши їх функціонал, особливості та взаємодію.

Почнемо з серверної частини. Сервер у гемдейві є серверною частиною програмного забезпечення та відповідає за обробку та керування основною логікою гри. Цей компонент взаємодіє з клієнтами або гравцями, обробляє їхні запити та забезпечує синхронізацію гри в реальному часі. Сервер відповідає за керування станом гри, безпеку та координацію математичних розрахунків, щоб забезпечити серію інтерактивних подій для гравців.

Основні функції сервера. Сервер є центральною частиною системи, взаємодіючи з усіма компонентами. Його основні функції включають обробку запитів від графічної частини гри, координацію математичних розрахунків і збереження стану гри.

Зв'язок з іншими компонентами. Сервер взаємодіє з математичною частиною, отримуючи результати обчислень, і з графічною частиною, обробляючи її запити та надсилаючи відповіді. Крім того, сервер відповідає за обробку та зберігання ігрових даних, таких як статус гравця, рівень складності тощо.

Безпека та оптимізація. Беручи до уваги важливість забезпечення безпеки та продуктивності гри, сервер також включає механізми автентифікації, авторизації та оптимізовані алгоритми обробки запитів.

Розглянемо не від'ємну частину продукту та можна сказати основну, це математика. Математика є невід'ємною частиною розробки ігор, відповідає за обчислення, логіку гри та взаємодію об'єктів. Ігрові математики розробляють і впроваджують алгоритми для визначення фізичних властивостей об'єктів, логіку штучного інтелекту та визначають імовірнісні моделі, які генерують різні ігрові події.

Роль математики в іграх. Математика є важливою частиною гри, оскільки вона відповідає за обчислення та логіку, необхідні для правильного функціонування гри. Його можливості включають обчислення фізичних властивостей об'єктів, логіку для штучного інтелекту та алгоритми генерації випадкових подій.

Математичні алгоритми в розробці ігор. Ми розглянемо різні математичні аспекти розробки ігор, такі як розрахунок траєкторій руху, динаміка об'єктів, алгоритми штучного інтелекту та ймовірнісний аналіз.

Використовуйте математику в графічних ефектах. Математика також відіграє ключову роль у створенні візуальних елементів гри, таких як реалістичні анімації, маніпуляції світлом і тінями, анімовані переходи між рівнями тощо.

Заключна і найбільш відповідальна частина, яка відповідає за успішність та візуальне сприйняття, а також об'єднує всі інші частини в себе – це графічна частина. Графічна частина, яку часто називають «інтерфейсом», відповідає за візуальне представлення гри та взаємодію з гравцем. Він містить графічний механізм, який відтворює графіку, механізм введення та інтерфейс користувача. Графічна частина визначає зовнішній вигляд гри, візуальні ефекти і, що важливо, взаємодіє з сервером, обмінюючись інформацією та обробляючи відповіді в реальному часі.

Роль графічної частини. Розділ графіки відповідає за відображення гри та взаємодію з користувачем. Основні компоненти цієї частини включають графічний механізм, механізм введення, обробку подій та інтерфейс користувача.

Графічний движок і його можливості. Графічний механізм відповідає за рендеринг графіки гри. Він може включати різні технології, такі як трасування променів, оптимізація для використання різних відеокарт, підтримка 3D-графіки та інші функції.

Взаємодія з сервером. Графічна частина взаємодіє з сервером шляхом надсилання запитів та обробки отриманих відповідей. Це включає обмін даними про статус гри, події та інші аспекти.

Розробка інтерфейсу користувача. Окремим аспектом графічного розділу є розробка інтерфейсу користувача, який включає графічні елементи, введення та навігацію для легкої та ефективної взаємодії з грою.

Розглянувши архітектуру програми та виділивши ключові компоненти, такі як: серверна, математична та графічна частини, можна сказати, що детальне вивчення кожного компонента розкриває та показує важливість їх ролі в розробці гри та взаємодії між ними. Кожен компонент відіграє вирішальну роль у забезпеченні стабільності, безпеки, ефективності та візуальної привабливості вашої гри.

2.1.2. Ключові компоненти системи

Сервер геймдев побудовано на основі Spring Boot framework і є важливим компонентом, який забезпечує ефективну та надійну інфраструктуру для ігрового середовища. Spring Boot — це структура, яка спрощує та прискорює розробку серверних програм, забезпечуючи високий рівень абстракції та автоматизації щоденних завдань.



Рис. 2.2. Лого Spring Boot framework

Структурований підхід до розробки у Spring Boot дозволяє ефективно взаємодіяти з логікою мережі, ефективно обробляючи запити та відповіді. Це важливо для забезпечення оптимальної взаємодії між гравцями та серверами, що впливає на швидкість гри та загальний досвід користувача.

Однією з головних переваг використання Spring Boot є можливість ефективного впровадження систем безпеки, таких як автентифікація та авторизація. Це важливо для забезпечення конфіденційності та цілісності даних гри. За допомогою Spring Security, розширеного модуля безпеки у Spring, розробники можуть легко налаштувати політики безпеки, контролювати доступ до ресурсів і забезпечити надійну автентифікацію гравців.

Підсумовуючи, використання Spring Boot у розробці ігор стало стратегічним вибором для створення стабільного, ефективного та масштабованого ігрового сервера. Його гнучкість і простота використання роблять його ідеальним для розробників, які хочуть надати гравцям надійне та високоякісне ігрове середовище.

Математика у геймдеві та Return to Player (RTP), сама математика в розробці ігор є важливою частиною розробки ігор, особливо в контексті обчислень повернення гравцеві (RTP). RTP визначає відсоток грошей, який гравець може очікувати отримати від ставок протягом тривалого періоду гри. Визначаючи RTP, розробники можуть регулювати економіку гри, щоб забезпечити справедливий баланс між виграшем і програшем, тим самим впливаючи на гравців і їхній ігровий досвід.



Рис. 2.3. Return to Player (RTP)

Одним із ключових завдань математиків у розробці ігор є обчислення ймовірностей подій у грі. Наприклад, яка ймовірність того, що гравець виграє певну суму грошей або програє певну суму грошей за певних умов гри. Ці розрахунки базуються на статистичних і математичних моделях, що дозволяє гравцям визначати виграшні та програшні ситуації.

Оптимальний RTP дуже важливий для привабливості гри для гравців. Якщо RTP занадто низький, гравці можуть відчувати, що їхні шанси на перемогу незначні, що призведе до втрати інтересу. З іншого боку, якщо RTP занадто високий, гра може стати збитковою. Оптимальний RTP дуже важливий для привабливості гри для гравців. Якщо RTP занадто низький, гравці можуть відчувати, що їхні шанси на перемогу незначні, що призведе до втрати інтересу.

Однією з головних функцій RTP є створення відчуття справедливості серед гравців. Коли гравець знає, що гра пропонує адекватні рівні бонусів, він відчувається захищеним і впевненим у цілісності ігрового процесу. Це допомагає побудувати довгострокові відносини між гравцями та розробниками, що є важливим аспектом успішної ігрової компанії.

У віртуальних іграх, де відбувається велика кількість ігрових операцій, обчислення RTP вимагають не лише глибоких знань математичних концепцій, але також потребують урахування різноманітних ігрових сценаріїв та взаємодії з ігровим середовищем. Тільки такий комплексний підхід може створити фінансово вигідну і в той же час захоплюючу гру для гравців.



Рис. 2.4. Лого Java та JavaFX

Графічна частина гри реалізована на Java та JavaFX і відіграє ключову роль у створенні візуально привабливого та захоплюючого ігрового середовища. Java і JavaFX — це потужна комбінація для реалізації графічної частини вашої гри. JavaFX, як графічна структура, забезпечує зручний інтерфейс для створення багаторівневих інтерфейсів і обробки подій від користувача. Використання графіки в JavaFX дозволяє створювати красиві та інтерактивні ігрові сцени. Використання JavaFX дозволяє розробникам створювати багаторівневі графічні інтерфейси, анімацію та спеціальні ефекти, які ще більше занурюють гравців у гру.



Рис. 2.5. Лого JSON

Підтримувати та керувати ігровими ресурсами, такими як зображення, анімація, звуки, шрифти та текст, за допомогою файлів .JMM. Зберігання ігрових активів у цьому форматі дозволяє зберігати структури даних у тому самому форматі, що й JSON, забезпечуючи розробникам легку читабельність і зручний доступ до ресурсів. Використання JSON для аналізу активів, таких як зображення, анімація, звуки та текст, є ефективним способом керування ігровими активами. JSON (Java Metadata Model) дозволяє структурувати дані в зручному для читання форматі, що спрощує роботу розробників і дозволяє легко змінювати і доповнювати ресурси. Розбір JSON дозволяє гнучко читати дані та ресурси, указані у форматі JSON, і динамічно завантажувати їх у гру. Це особливо корисно для змінних або динамічних ігрових об'єктів (таких як анімація або різні рівні), оскільки їх можна змінювати без перекомпіляції коду.



Рис. 2.6. Лого Spine

Технологія Spine для анімації забезпечує гнучкий і реалістичний рух персонажа. Цей метод анімації допомагає створити яскравий і емоційно насичений ігровий процес, поглиблюючи ігровий досвід. Насправді Spine є потужним інструментом для створення анімації в графічних програмах. Використовуючи технологію Spine, розробники можуть створювати гнучку та динамічну анімацію, яку можна легко інтегрувати в ігри. Spine використовує скелетну анімацію, що дозволяє анімувати різні частини вашого персонажа, забезпечуючи плавний, природний рух. Зміною положення і поворотом кісток можна керувати програмно, що відкриває безліч можливостей для створення різноманітних анімаційних ефектів. Використання Spine також дозволяє зберігати анімацію в компактному форматі, що допомагає оптимізувати розмір даних і швидкість завантаження. Це важливо для ігор, оскільки забезпечує плавне та ефективно відтворення анімації на різних пристроях.

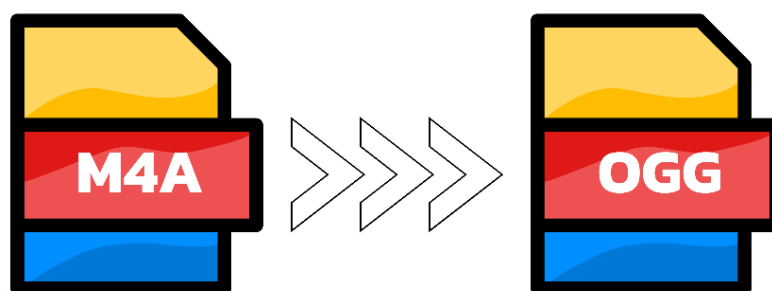


Рис. 2.7. Аудіоформати .m4a та .ogg

Аудіоформати .m4a та .ogg, які використовуються для представлення звуків у грі, забезпечують високу якість аудіо та сумісність між платформами. Це важливо для забезпечення однаково високоякісного звукового супроводу

незалежно від того, на якому пристрої ви граєте. Звук в іграх відіграє важливу роль у створенні атмосфери та поглибленні ігрового досвіду. Для нашого випадку використання форматів .m4a та .ogg є розумним та дуже гарним вибором.

- .m4a — це аудіоформат, який забезпечує високоякісний звук у файлі меншого розміру. Цей формат часто використовується для забезпечення чіткого високоякісного звуку в іграх.
- .ogg — ще один популярний аудіоформат, відомий своєю компактністю та високою якістю, і його використання забезпечує сумісність між платформами, щоб гравці могли насолоджуватися високоякісним аудіо.

Використання цих форматів є важливим аспектом оптимізації, оскільки вони забезпечують високу якість звуку без значного збільшення розміру гри. Такий підхід покращує продуктивність гри та надає гравцям приємні звукові ефекти.

Загалом, графічна частина, побудована на Java та JavaFX з використанням технології Spine, у поєднанні з використанням найкращих аудіоформатів створює вражаюче та захоплююче ігрове середовище, яке сприяє успіху гри серед широкого кола гравців.

2.2. Опис моделі поведінки гравців

2.2.1. Детальний опис моделі поведінки гравців

Базується на принципах ігрових автоматів і включає багато ключових аспектів, які визначають поведінку гравця. Графічні та звукові елементи, різноманітні функції та ігрові можливості в ігрових автоматах визначають непередбачуваний та захоплюючий характер гри. Детальний опис моделі поведінки допоможе краще зрозуміти, як різні аспекти гри впливають на взаємодію з нею гравця.

Типові дії гравців

Натискання на кнопку "Спін" – гравці можуть змінювати частоту натискання кнопки обертання залежно від своєї гри. Деякі можуть спробувати більше в пошуках перемоги, а інші можуть стратегічно намагатися менше.

Використання Turbo mode – «Турбо» режим визначається швидшими інтервалами між обертаннями та призначений для тих, хто шукає швидкий ігровий процес та адреналін. Гравці можуть періодично вибирати цей режим для більшої життєвості.

Використання AutoPlay – це функція автовідтворення стане в нагоді гравцям, які не хочуть витратити час на щоразове натискання кнопки. Він надає можливість налаштувати кількість автоматичних обертань і умови зупинки.

Ставки та RTP

Діапазон ставок – гравці можуть обирати ставки від 0,1 до 100. Розмір ставки визначає ризик і потенційні шанси на виграш, а різноманітність ставок робить гру доступною для гравців з різним бюджетом.

Вплив RTP – високий рівень повернення гравцеві (RTP) дає гравцям впевненість у перемозі. Перед розробниками стоїть завдання налаштувати RTP, щоб забезпечити економічну привабливість гри.

Вплив аудіо та візуалізації

Вплив музики – удіоелементи, включаючи музику, можуть визначати настрій гравця. Вибір гармонійної музики може підняти настрій і зробити гру більш захоплюючою.

Вплив візуалізації – якісна графіка і ефектна анімація створюють враження справжньої гри. Візуалізація дуже важлива, щоб гра виглядала захоплюючою та привертала увагу гравця.

Таймінги перед анімаціями та фічами

Витримка таймінгів – встановлення часу перед анімацією або функцією відіграє важливу роль у створенні зручного ігрового процесу. Гравці можуть мати різні ритми гри, і оптимізований час враховуватиме цю варіацію.

Детальний аналіз моделей поведінки гравців може дозволити розробникам краще зрозуміти потреби різних категорій гравців і оптимізувати ігри для різних сценаріїв взаємодії. Такий підхід допомагає створювати ефективні, захоплюючі ігри, які відповідають різним стилям і стратегічним уподобанням гравців.

2.2.2. Алгоритми та стратегії, що використовуються

Одним з ключових елементів ігрових автоматів є алгоритм і стратегія, які визначають гравців. Є багато методів, за допомогою яких гравці можуть оптимізувати свій ігровий процес і збільшити свої виграші. Давайте розглянемо кілька популярних алгоритмів і стратегій, які використовують гравці в іграх на ігрових автоматах, даліше розглянемо кожну стратегію більш детальноше.

Стратегія фіксованих ставок – гравці визначають фіксовану суму ставки на обертання. Ця стратегія дозволяє вам управляти своїм бюджетом і ігровим часом, знижуючи ризик великих втрат. Гравці можуть використовувати цю стратегію, щоб забезпечити стабільну гру незалежно від результату кожного обертання. Можна виділити основні аспекти, такі як стабільний бюджет, скільки кожна ставка є фіксованою, гравці можуть ефективно керувати своїм бюджетом. Збереження гри – гравці можуть продовжувати грати тривалий час, навіть якщо виграші та програші невеликі. Обмежені можливості максимізації виграшів – гравці можуть втратити можливість збільшити свої виграші, оскільки їхні ставки залишаються незмінними.

Прогресивна стратегія ставок – гравці коригують свої ставки на основі результатів попередніх обертань. Наприклад, збільшуйте ставку після кожної перемоги або зменшуйте ставку після програшу. Цей метод дозволяє максимізувати ваші виграші під час щасливих обертань і мінімізувати втрати під час менш успішних обертань. До основних аспектів можна віднести максимізація виграшів – гравці можуть збільшувати свої ставки, коли виграють, максимізуючи ваш прибуток, адаптивність – за різними результатами стратегія змінюється, збільшення ризику – високі ставки можуть призвести до швидших програшів,

якщо оберти пройдуть погано, великі коливання – взяти на себе ризик великих змін обмінних курсів.

Стратегія "Пресування на виграш" – гравці залишають свої ставки незмінними або лише збільшують їх після кожного виграшу. Мета цього — збільшити відсоток виграшів і зменшити втрати, зберігаючи стабільність гри. Ця стратегія передбачає, що середні бонуси мають тенденцію до зростання. Включає наступні аспекти, а саме стабільні виграші – гравці забезпечують стабільний прибуток, зменшуючи свої ставки після програшу, мінімізація втрат – збільшуючи свої ставки після кожного виграшу, щоб зменшити можливі втрати. З негативних аспектів можливий знижений виграш – а збільшення своїх ставок після виграшу може призвести до втрати більшої кількості виграшних можливостей у короткостроковій перспективі, залежність від патернів – ефективність цієї стратегії може залежати від наявності певних закономірностей у бонусах.

Стратегія "Ліміт втрат" – гравець встановлює ліміт програшу, і після досягнення цього ліміту він припиняє гру. Така стратегія допомагає уникнути серйозних фінансових втрат і не витримувати бюджет. З позитивних аспектів це захист бюджету – гравці захищають свій бюджет, встановлюючи ліміти втрат, відповідальна гра – заохочується відповідальна гра, оскільки гравці не ризикують. Також присутні аспекти негативні це можливість втратити виграш – гравець ризикує припинити гру, навіть якщо він уже виграв, що позбавляє його можливості максимізувати свій прибуток, стоп-лосс із малими втратами – стоп-лосс дозволяє гравцям вийти з гри з невеликими відшкодованими втратами.

Стратегія "Максимальний RTP" – гравці обирають ігри з високим показником повернення гравцеві (RTP), оскільки це збільшує ймовірність виграшу в довгостроковій перспективі. Вибір гри з найкращим RTP стає ключовим елементом цієї стратегії. Такі аспекти як збільшена ймовірність виграшу – вибір гри з високим RTP збільшить ймовірність виграшу гравця, економічна привабливість – у довгостроковій перспективі гравці можуть отримати більше бонусів, є позитивною частиною. Але також присутні і

протиположні аспекти обмежений вибір ігор – гравці можуть бути обмежені у виборі ігор і орієнтуватися на RTP, відсутність гарантій – навіть ігри з високим RTP не гарантують перемогу в кожній грі.

Алгоритм зупинки – гравець визначає конкретні умови, за яких гра закінчується. Це може бути досягнення певної кількості вигравів, програвів або обертань. Алгоритм зупинки дозволяє гравцям зберігати контроль над грою, а саме наступними аспектами, контроль – встановивши певні умови для завершення гри, гравці можуть зберігати контроль над грою, збереження виграшу – допомагає уникнути втрат та закріпити виграш гравця. З негативних, пропуск можливих вигравів – не дозволивши грі розвинутися гравці можуть втратити можливість отримати більше вигравів, варіабельність гри – від зупинки гри залежить подальший результат.

Кожна стратегія має свої унікальні аспекти, переваги та недоліки, і вибір конкретної стратегії залежить від особистих уподобань і стратегії гравця. Ця різноманітна стратегія додає глибини та різноманітності грі, роблячи її більш цікавою для широкого кола гравців, а поєднання різних стратегій може допомогти гравцям забезпечити різноманітність і задоволення від гри.

Висновки до розділу 2

Розглядаючи архітектуру програми та основні компоненти, такі як серверна, математична та графічна частини, можна визначити ключові аспекти, які впливають на стабільність, безпеку, ефективність та візуальний аспект гри. Кожен компонент відіграє важливу роль у взаємодії, сприяючи створенню гри високої якості.

Зокрема, графічна частина, заснована на Java та JavaFX з використанням технології Spine, разом із використанням передових аудіоформатів, формує захоплююче ігрове середовище, сприяючи широкому застосуванню гри серед гравців.

Детальний аналіз моделей поведінки гравців дозволяє розробникам краще розуміти потреби гравців та оптимізувати ігри для різних сценаріїв взаємодії. Цей підхід сприяє створенню ефективних та захоплюючих ігор, які враховують різні стилі та стратегії гравців.

Враховуючи, що кожна стратегія має свої унікальні аспекти, вибір конкретної стратегії залежить від особистих уподобань та стратегії гравця. Різноманітність стратегій додає глибину та різноманітність грі, зроблюючи її привабливою для широкого кола гравців, а комбінація різних стратегій підсилює задоволення від гри.

РОЗДІЛ 3

ІМПЛЕМЕНТАЦІЯ ТА НАЛАШТУВАННЯ МОДЕЛІ

3.1. Імплементация моделі поведінки гравців у Java-застосунку

3.1.1. Опис імплементации моделі у програмі

Наука яка використовує математичні принципи та моделі в проектуванні, розробці та оптимізації ігрових систем – математика (в розробці ігор). Щоб забезпечити баланс та стабільність гри вона аналізує всі аспекти та властивості гри. Return to Player (RTP) – функція, яка допомагає визначити, яка частина грошей, витрачених гравцями, повертається у виграшах. Також математика впливає на математичні моделі, баланс гри та на інші сторони, які забезпечують успішність ігрових продуктів.

Основи та цілі включають визначення ймовірностей, розробку математичних алгоритмів, генерування випадкових чисел і обчислення повернення гравцеві (RTP).

Що стосується визначення ймовірностей, це передбачає призначення математичних ймовірностей різним ігровим подіям, таким як виграш, програш або активація бонусу. Ймовірність розраховується на основі положення символу на барабанах, його типу та взаємодії з іншими елементами гри.

Створення математичних алгоритмів включає створення алгоритмів для розрахунку ставок, визначення бонусів і бюджетів гравців. Це допомагає збалансувати виграші та поразки, щоб зацікавити гравців. Він також враховує такі фактори, як тип символу, комбінації, бонусні раунди та безкоштовні обертання.

Кафедра КІТ (47)				НАУ 23.05.85.000 ПЗ			
Виконав	Горічко А.О.			ІМПЛЕМЕНТАЦІЯ ТА НАЛАШТУВАННЯ МОДЕЛІ	Літ.	Арк.	Аркушів
Керівник	Воронін А.М.					53	24
Консульт.					УС-211М 122		
Н. Контр.	Райчев І.Е.						

Генерація випадкових чисел використовує генератор для моделювання випадкових результатів гри, забезпечуючи непередбачуваність і різноманітність гри. Він також використовується для моделювання різних ігрових сценаріїв, щоб забезпечити випадковість результатів.



Рис. 3.1. Допоміжна сцена на якій описуються математичний вплив

Розрахунок повернення гравцеві (RTP) передбачає визначення та обчислення відсотка грошей, які гравець може очікувати взамін від гри. Він враховує ймовірність виграшу на основі комбінації символів на барабанах, обчислює загальний RTP гри та використовує його як ключовий аспект для визначення фінансової привабливості гри.

За допомогою математики отримуємо великий спектр плюсів, до яких можна віднести:

- Баланс гри – математика забезпечує справедливість гри, вигідний досвід для гравців та збалансованість гри.

- Прогнозування – математичні моделі допомагають точно прогнозувати ймовірність виграшу.
- Ігрові механіки – математика може допомогти розробникам збалансувати гру.
- Ставки та виграші – математика забезпечує гравцям справедливий і прибутковий досвід.
- Віртуальні спостереження – застосування математичних моделей може точно передбачити ігровий процес і ймовірність перемоги.

Попри велику кількість плюсів, існують декілька негативних факторів, можна віднести:

- Складність розробки – потрібно багато часу та роботи для реалізації складних математичних алгоритмів та формул.
- Відсутність гарантій – результати гри залишаються випадковими навіть з точним розрахунком ймовірностей.

Без належно розробленої математичної моделі гра може втратити баланс і спричинити несправедливість або нудьгу гравців. З іншого боку, правильне використання математики забезпечує стабільний і захоплюючий ігровий процес, що допомагає забезпечити якісний ігровий досвід.

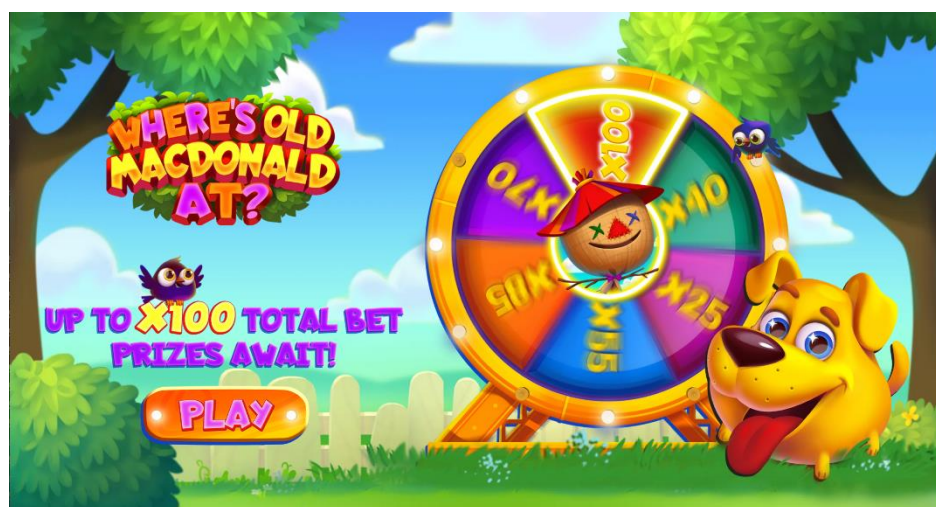


Рис. 3.2. Сцена завантаження гри

Фронтенд частина, так звана, більш простими словами інтерфейсна частина розробки гри — це частина розробки гри, яка відповідає за візуальні та звукові аспекти ігрового процесу, які взаємодіють з гравцем. Він зосереджений на створенні інтерфейсів користувача (UI), анімації, графічних ефектів та інших елементів, які впливають на зовнішній вигляд і взаємодію гравців з грою.

Інтерфейсна частина розробки гри в основному включає графічну частину, інтерфейс і керування, звукову частину, анімацію та ефекти, оптимізацію та сумісність. Розділ графіки охоплює створення та інтеграцію графіки, анімації та спеціальних ефектів. Інтерфейс і керування включають розробку та впровадження інтерфейсів користувача для полегшення взаємодії гравця з грою. Звукова частина передбачає інтеграцію музики та звукових ефектів для створення атмосфери та посилення емоцій гравців. Анімація та ефекти включають реалізацію рухомих об'єктів і візуальних ефектів. Оптимізація та сумісність включають оптимізацію для різних платформ, щоб забезпечити сумісність інтерфейсної частини з ігровим движком та іншими компонентами гри. Інтерфейсна частина визначає візуальний стиль і естетику гри, впливає на перше враження гравця та сприяє захоплюючому ігровому досвіду.

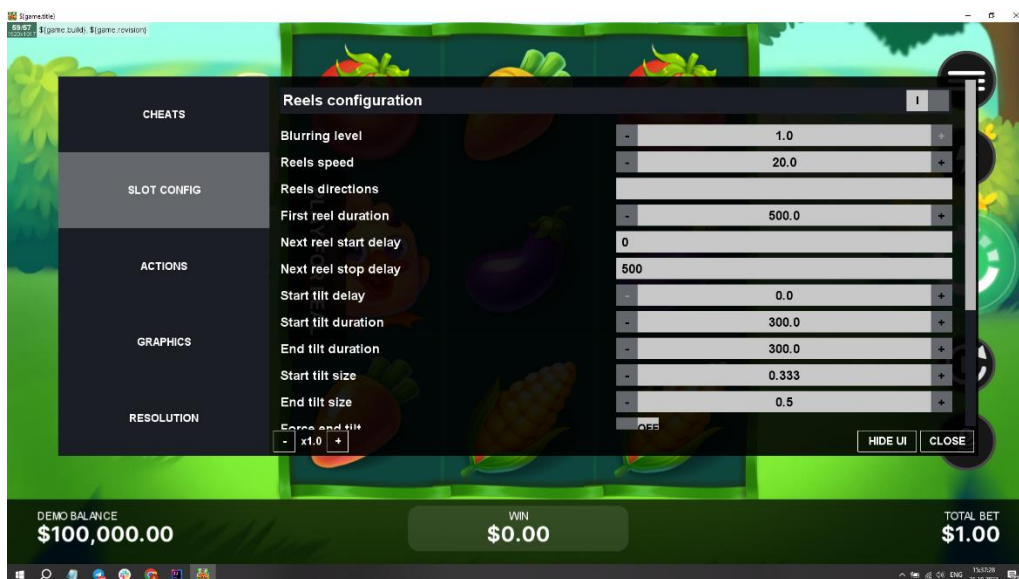


Рис. 3.3. Меню конфігурації барабанів

Конфігурація барабанів у розробці ігор включає низку параметрів, які визначають рух і вигляд барабанів у слот-грі. Розглянемо докладніше кожен параметр.

Рівень розмиття (Blurring level) визначає, наскільки розмитими є символи на барабанах під час обертання, що впливає на візуальний ефект руху.

Швидкість барабанів (Reels speed) визначає, наскільки швидко вони обертаються, впливаючи на динаміку та емоційні нагороди гри.

Напрямок катушки (Reel direction) встановлює напрямок обертання барабанів, дозволяючи створювати різноманітні ефекти для покращення візуального досвіду.

Тривалість першого барабана (First reel duration) вказує, як довго він залишається в русі до зупинки, утворюючи ефект анімації.

Затримка наступного барабана (Next reel start delay) визначає затримку до початку обертання наступного барабана, впливаючи на темп і ритм гри.

Затримка наступної зупинки (Next reel stop delay) вказує затримку перед зупинкою, що впливає на візуальний ефект і точність зупинки.

Затримка початкового нахилу (Start tilt delay) визначає затримку перед нахилом барабана, тоді як тривалість початкового нахилу (Start tilt duration) та тривалість кінцевого (End tilt duration) нахилу вказують на тривалість ефекту анімації, коли він припиняє починатися та закінчуватися.

Розмір початкового (Start tilt size) перекоосу та розмір кінцевого (End Tilt Size) перекоосу впливають на розмір і зовнішній вигляд перекоосу. Силовий кінцевий перекоос (Force end tilt) означає ефект примусового завершення нахилу після зупинки барабана.



Рис. 3.4. Меню конфігурації барабанів продовження

Очікувана початкова швидкість (Anticipation start speed) визначає швидкість початкового обертання, що впливає на візуальний ефект початку гри.

Очікувана кінцева швидкість (Anticipation end speed) вказує на очікувану кінцеву швидкість обертання барабанів.

Очікувана тривалість посилення (Anticipation acc duration) визначає, скільки часу триває посилення перед обертанням, що впливає на динаміку та інтригу гри.

Час очікування (Anticipation time) — це загальний час очікування, перш ніж барабани почнуть обертатися, створюючи загальний візуальний ефект і впливаючи на темп гри.

Детальний аналіз параметрів конфігурації барабанів підкреслює їх ключову роль у створенні візуально привабливих і емоційно насичених ігрових вражень, дозволяючи розробникам точно налаштовувати їх для досягнення бажаних ефектів і взаємодії з гравцями.



Рис. 3.5. Бонусна гра. Spine елемент колесо

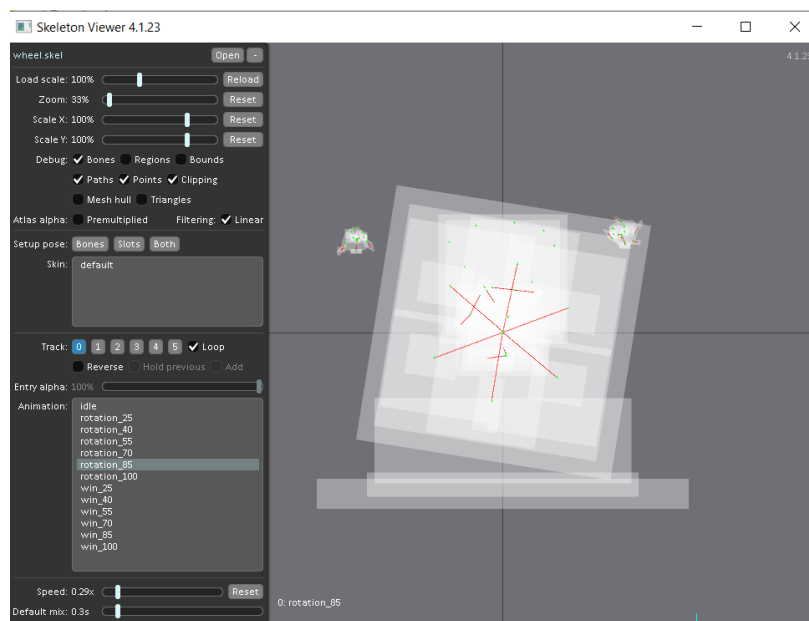


Рис. 3.6. Skeleton Viewer. Прототип колеса з кістками

Spine анімація та анімація, створена всередині за допомогою коду, надає розробникам важливий набір інструментів для створення різноманітних і вражаючих анімаційних ефектів у розробці ігор.

Тривалість анімації (Animation Duration) визначає тривалість відтворення анімації та впливає на час, коли анімовані об'єкти рухаються або змінюються.

Затримка перед відображенням анімації (Animation Delay) визначає час затримки перед початком відтворення анімації. Якщо активовано, це створить ефект очікування та розширення.

Початок анімації визначає момент часу або подію, яка викликає початок анімації, керуючи точним моментом активації відносно інших подій у грі.

Кінець анімації вказує момент завершення анімації та поведінку після її завершення.

Цикл анімації визначає, чи буде анімація автоматично повторюватися після її завершення, і забезпечує повторюваний цикл анімації.

Параметри трансформації (TransformationParameters) включають масштабування, обертання та зміщення, які змінюють положення та форму об'єкта під час процесу анімації.

Елемент керування інтерполяцією визначає тип інтерполяції для плавних переходів між ключовими кадрами, що впливає на вигляд і реалістичність переходу.

Уповільнення та прискорення (Ease In, Ease Out) контролюють розгортання анімації, забезпечуючи плавний початок і кінець руху анімації, тим самим покращуючи візуальну якість і додаючи руху природності.

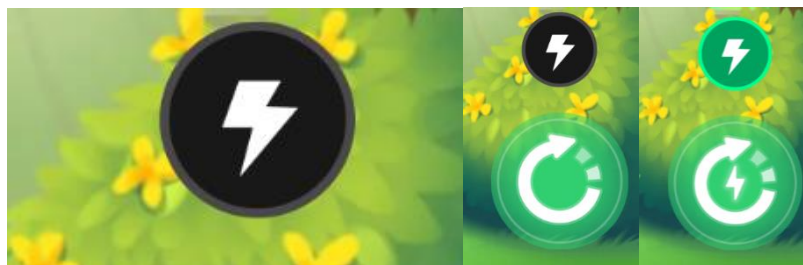


Рис. 3.7. Кнопка режим Turbo та її стани

Режим Turbo — це функція в ігрових автоматах, яка пришвидшує темп гри, дозволяючи гравцям обертати барабани та швидше отримувати результати. Це рішення для тих, хто шукає більш динамічний і яскравий ігровий процес. Вплив на геймплей даного режиму, із позитивної сторони швидша гра – режим Turbo дозволяє гравцям пришвидшити гру та швидше отримувати результати обертання, а також добавляє ефективніший геймплей – гравці можуть зробити більше обертань за менший час, що може призвести до більшого ефекту обертання. З негативного втрата атмосферності – щоб грати в ігри швидше,

гравці можуть втратити можливість насолоджуватися деталями гри, а також присутні можливі великі втрати – швидка гра також призводить до швидшого виснаження балансу, що може вплинути на гравців із меншою фінансовою стабільністю.

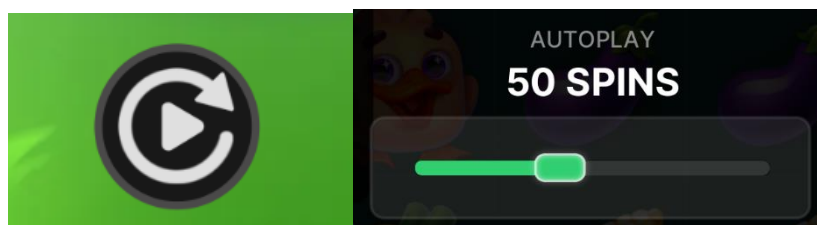


Рис. 3.8. Кнопка Autoplay та спливаючі вікно з вибором кількості спінів

Автозапуск (Autoplay) — це функція в ігрових автоматах, яка дозволяє гравцям автоматично обертати барабани без необхідності постійно натискати кнопку обертання. З плюсів можна виділити зручність – автоматичне відтворення забезпечує зручність для гравців, оскільки немає необхідності постійно натискати кнопку обертання та економія часу – гравці можуть заощадити час, встановивши кількість автоматичних обертань і перерв під час гри. З мінусів втрата контролю – гравці можуть втратити контроль над грою, оскільки автоматична гра вирішує, коли обертатися та потенційні втрати – автоматичне обертання може призвести до втрат, якщо гравець не контролює його.

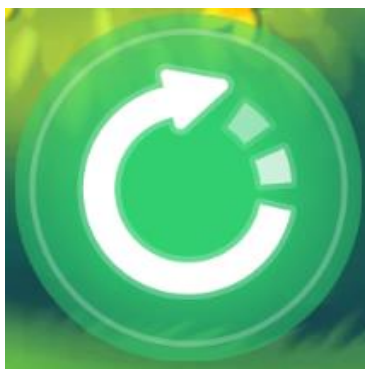


Рис. 3.9. Кнопка “СПІН”

Швидкість Реагування на Кнопку "Спін", чуйність кнопки обертання — це термін, який описує, скільки часу потрібно, щоб гра реагувала на натискання гравцем кнопки обертання. Можна виділити основні позитивні моменти як більший контроль – швидка реакція кнопки «Обертання» дає гравцям більше контролю над кожним обертанням та миттєві результати – гравці одразу отримують результати обертання, що забезпечує більш динамічний ігровий процес. З негативних моментів стрес – занадто швидка гра може призвести до стресу, оскільки гравці відчуватимуть тиск, змушений швидко приймати рішення та відсутність концентрації – занадто швидка гра може призвести до відсутності концентрації та необдуманих рішень.

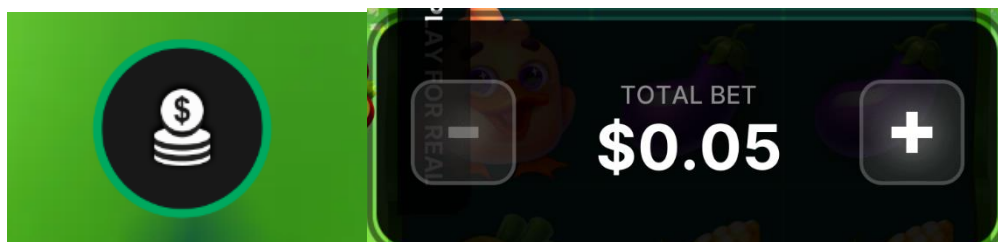


Рис. 3.10. Кнопка вибору ставки та спливаючі вікно з вибором ставки

Вибір доступних ставок — це можливість гравця вибрати суму ставки, яку він готовий витратити на одне обертання барабанів у грі. Основні аспекти контроль фінансів – гравці можуть обирати ставки, які відповідають їхнім фінансовим можливостям та ігровим стратегіям, експериментація – можливість маневрувати між різними ставками дозволяє гравцям спробувати різні ігрові стратегії, потенційні втрати – великі ставки можуть призвести до значних втрат, тоді як малі ставки можуть зменшити виграш, залежність від удачі – вибір ставок впливає на ступінь довіри до удачі та виграшу в грі.

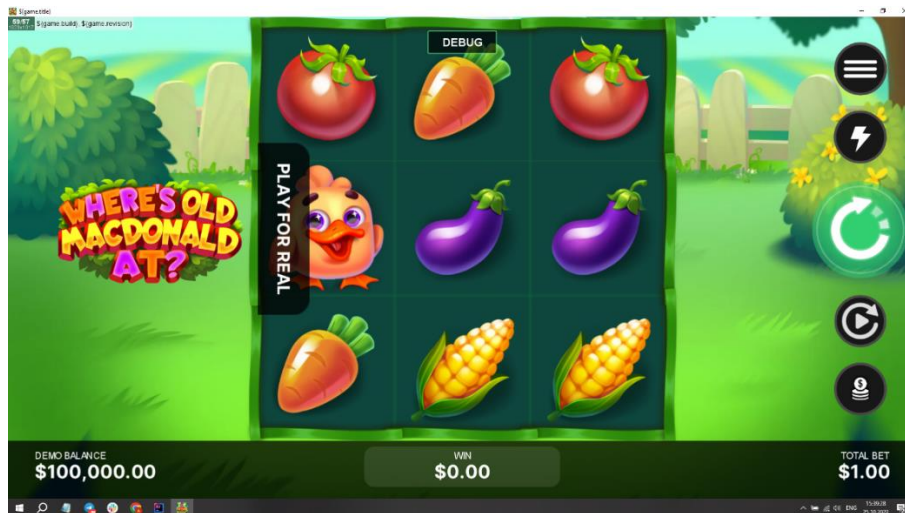


Рис. 3.11. Головна сцена в грі

Інтерфейс — це взаємодія гравців із грою за допомогою таких графічних елементів, як кнопки, меню, віджети та інші елементи. У розробці гри інтерфейс відіграє важливу роль у забезпеченні зручності та задоволення гравця під час гри.

Розташування кнопок має бути стратегічним, щоб полегшити доступ гравця та мінімізувати втрату часу на пошук. Ігрові меню мають бути простими у використанні, зрозумілими та забезпечувати швидкий доступ до основних функцій.

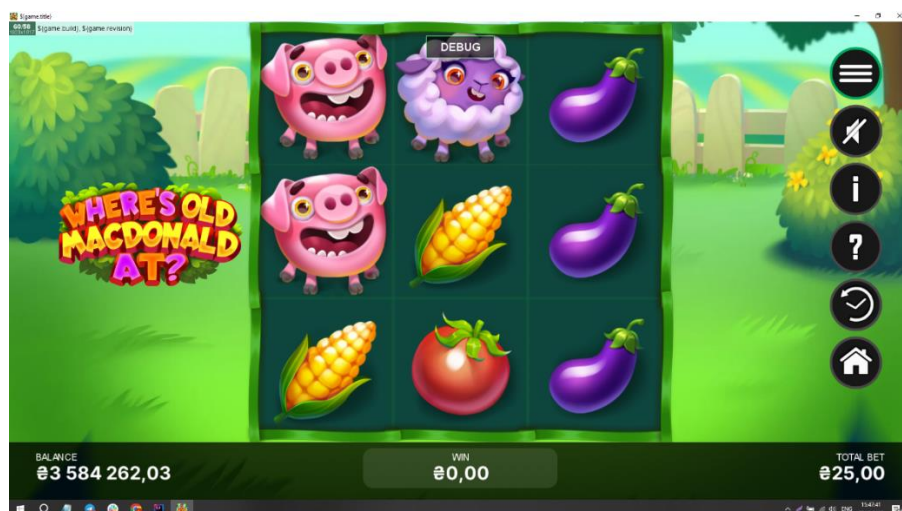


Рис. 3.12. Головна сцена гри із відкритим меню

Зручність включає адаптивність інтерфейсу до різних пристроїв, що дозволяє комфортно грати на комп'ютерах і мобільних пристроях. Гравці

повинні мати можливість налаштувати інтерфейс на свій смак. Інтерфейс має мінімізувати відволікання від гри та забезпечувати плавну взаємодію. Інформаційна прозорість важлива для передачі ключової інформації гравцям про ставки, бонуси та інші параметри гри.

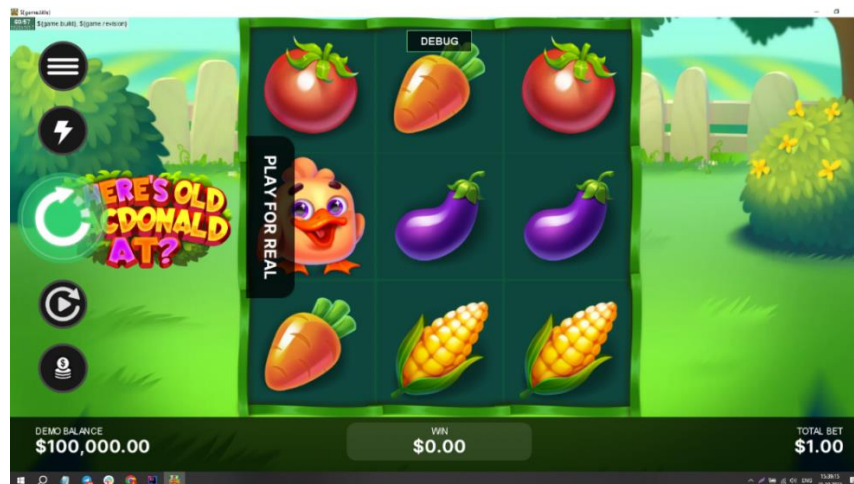


Рис. 3.13. Головна сцена гри із кнопками зліва

Графічне оформлення має бути привабливим і відповідати стилю гри, залишаючи позитивне враження. Чіткі зображення та символи допомагають швидше зрозуміти елементи інтерфейсу.

Взаємодія з кнопками підтримується звуковим і візуальним зворотнім зв'язком. Інтерфейс повинен бути інтуїтивно зрозумілим навіть новачкам. Загальна логіка передбачає спрощення інтерфейсу та уникнення складності та перенасиченості елементів. Взаємодія з інтерфейсом має відбуватися природним шляхом, не вимагаючи від гравця особливих зусиль.

Правильно розроблений інтерфейс у розробці гри відіграє ключову роль у створенні позитивного ігрового досвіду та забезпеченні комфорту гравця.

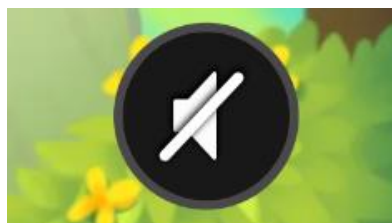


Рис. 3.14. Кнопка включення/вимкнення звуку

Аудіо частина розробки гри включає всі аспекти аудіоелементів у грі, такі як музика, звукові ефекти, голосовий супровід тощо. Інші аудіокомпоненти. Ця частина гри дуже важлива, оскільки впливає на емоційний стан гравця та додає атмосфери гри.

Музика повинна відповідати жанру гри та покращувати ігровий процес. Динаміка музики важлива для реагування на різні сцени гри, включаючи битви, спеціальні події та переходи між рівнями.

Звукові ефекти повинні точно відтворювати звуки, які відповідають подіям у грі. Використання технології просторового звуку дозволяє створити ефект занурення та покращити ігровий досвід.

Технічна оптимізація звукової частини дуже важлива для забезпечення високоякісного звучання на різних пристроях та аудіосистемах без перевищення розміру файлу. Синхронізація звуків з ігровими подіями забезпечує максимальне занурення.

Звук може викликати у гравців різні емоції, від нервозності до радості, тим самим покращуючи враження від гри. Правильно підібрані звуки можуть створити певну ігрову атмосферу і створити неповторний світ і настрої.

Звук є невід'ємною частиною розробки ігор і має величезний вплив на досвід гравця та загальний успіх гри.

3.1.2. Використані технології та інструменти

Java 15 та JavaFX

Java 15 — це випущена версія мови програмування Java, яка надає розробникам оновлення та інновації. Він надає сучасні можливості для створення різноманітних програм, у тому числі ігор. До основних переваг можна віднести оновлення та вдосконалення, використання Java 15 надає розробникам доступ до останніх оновлень і вдосконалень мови програмування Java, що забезпечує нові функції та підвищує продуктивність розробки сумісність з сучасними технологіями, забезпечує сумісність із сучасними технологіями, забезпечуючи розробникам доступ до найновіших технологічних бібліотек та

інструментів, а також висока швидкість та продуктивність, нові версії Java часто включають оптимізацію, яка покращує швидкість і продуктивність програми.

JavaFX — це бібліотека та платформа для створення графічних інтерфейсів користувача (GUI) у програмах, які використовують мову програмування Java. Він широко використовується в розробці ігор для створення візуально привабливих ігрових інтерфейсів. Має ряд переваг, таких як графічний інтерфейс високого рівня, JavaFX допомагає покращити взаємодію з користувачем, спрощуючи створення стильних інтерактивних графічних інтерфейсів для ігор, легка інтеграція як бібліотеку для розробки графічних інтерфейсів JavaFX можна легко інтегрувати з іншими частинами програми, забезпечуючи гнучкість розробки, можливості анімації та візуалізації надає широкий спектр інструментів для реалізації анімації, що дозволяє створювати захоплюючі та динамічні ігрові сцени.

Поєднання Java 15 і JavaFX є взаємозалежними технологіями для розробки графічних інтерфейсів за допомогою мови програмування Java. Використання Java 15 дозволяє використовувати новітні мовні функції та технологічні інструменти для оптимального розвитку.

Поєднання Java 15 і JavaFX забезпечує зручний і ефективний процес розробки ігор. JavaFX полегшує створення інтерфейсів, а Java 15 надає доступ до сучасних засобів розробки.

JavaFX дозволяє легко інтегрувати графічні та логічні рівні вашої гри. Використання Java 15 забезпечує сумісність і доступ до останніх функцій мови.

Використання останньої версії Java допомагає підтримувати сучасні графічні стандарти та функції, покращуючи вигляд вашої гри.

LibGDX

LibGDX — це потужний інструмент розробки ігор із вражаючим унікальним поєднанням функцій і переваг. Його кросплатформна природа дозволяє перетворити ваші ідеї в ігри, які подобаються користувачам на різних пристроях. Гнучкість і розширюваність LibGDX надають розробникам безмежні можливості, дозволяючи легко впроваджувати нові функції та розширювати

функціональні можливості за допомогою різноманітних плагінів і розширень. LibGDX оптимізовано для високої продуктивності та є ідеальним вибором для тих, хто хоче створювати ігри з великими графічними елементами та анімацією без шкоди для швидкості та плавності гри. Завдяки активній спільноті розробників і доступній документації робота з LibGDX стає захоплюючою подорожжю, де кожен може знайти підтримку та поділитися цінними знаннями. Його унікальна здатність підтримувати 2D і 3D графіку робить LibGDX чудовим вибором для розробки візуально привабливих і вражаючих ігор. Завдяки інтегрованій мультимедійній підтримці звуку, музики та відео LibGDX прокладає шлях для створення багатогранних ігор, які занурюють гравців у захоплюючі світи розваг. Низькі вимоги до ресурсів дозволяють створювати високоякісні ігри, зберігаючи оптимальну продуктивність пристрою.

JMM (JSON)

.JMM — це формат файлу, який використовується для зберігання ігрових активів у форматі JSON. Переваги при використанні JSON, сюди можна віднести читабельність, зручний для читання та редагування формат, який полегшує роботу з ігровими ресурсами та загальна сумісність, поширений формат обміну даними, який полегшує інтеграцію з іншими інструментами та бібліотеками.

Spine

Spine — це технологія ігрової анімації, яка дозволяє створювати гнучку та реалістичну анімацію для персонажів і об'єктів у вашій грі. Головна перевага Spine полягає в тому, що він забезпечує широкий контроль над рухом і зовнішнім виглядом об'єктів у реальному часі, забезпечуючи ефективне використання ресурсів і високий рівень деталізації в анімації. Основні характеристики та переваги Spine заключаються в скелетній анімації, використовували методи скелетної анімації, коли об'єкти складаються з частин (скелетів), які рухаються та обертаються навколо точок з'єднання, це дозволяє створювати гнучкі та реалістичні рухи. Параметризовані анімації є динамічними, такі параметри як розмір, колір та інші, тому що їх можна змінювати в реальному часі гри. Реалізація у графічних двигунах, можна легко інтегрувати в різні графічні

движки, такі як LibGDX, Unity, Cocos2d, що дозволяє використовувати його в різноманітних ігрових проектах. Ефективність ресурсів – анімації, створені за допомогою Spine, можуть бути досить легкими для системи, оскільки вони базуються на скелетних моделях, що забезпечує ефективне використання ресурсів та підтримка різноманітних платформ, найбільш значим є те, що Spine підтримує різні платформи, що дозволяє використовувати створені анімації на різних пристроях і операційних системах.

Використання Spine в геймдеві відбувається наступним чином, інтеграція ігрового механізму – розробники можуть інтегрувати бібліотеку Spine у свої ігрові механізми для створення анімації. Створення ключових кадрів – аніматори створюють анімацію, визначаючи ключові кадри та переходи між ними. Динамічні зміни – розробники можуть надати можливість динамічно змінювати параметри об'єкта під час відтворення анімації, а також оптимізація ресурсів які Spine дозволяє оптимізувати використання ресурсів і підтримувати анімацію легкою та ефективною.

Google Analytics

Google Analytics – це безкоштовна служба веб-аналітики від Google, яка дає змогу вимірювати й аналізувати трафік на вашому веб-сайті чи в програмі. Цей інструмент став невід'ємною частиною власників веб-сайтів і розробників, оскільки він надає важливі дані про трафік і поведінку користувачів.

Основні функції та можливості Google Analytics, заключаються в аналізі трафіку, Google Analytics збирає дані про кількість відвідувань вашого веб-сайту чи програми, розподіл трафіку на основі різних критеріїв (географічне розташування, тип пристрою, браузер тощо). Вимірювання поведінки користувачів відстежуючи різні дії користувачів, такі як перегляди сторінок, взаємодії з елементами, час, проведений на сайті, Google Analytics дозволяє зрозуміти, як користувачі взаємодіють із вашим вмістом. Конверсії та цілі, можна встановлювати цілі та вимірювати конверсії, які є діями, які ви хочете виконати від користувачів (наприклад, заповнити форму, купити продукт). Пошуковий трафік який Google Analytics відображає дані про трафік із

пошукових систем, інших веб-сайтів і соціальних мереж. Аналіз ефективності реклами якщо використовувати рекламні кампанії, Google Analytics може надати вам дані про їх ефективність і вартість. Аналіз та призначення – сервіс надає важливі дані про використання мобільних пристроїв і показники ефективності мобільного вмісту, і ви можете розділити свою аудиторію на різні сегменти для більш детального аналізу та порівняння. Підключення додаткових модулів – існують різні доповнення та інтеграції, які дозволяють розширити можливості Google Analytics на основі вашого бізнесу.

Використання Google Analytics в геймдеві переважно для моніторингу взаємодії гравців, щоб аналізувати такі показники, як взаємодія гравця з грою, час гри тощо. Вимірювання конверсій у грі, ставити цілі та вимірювати конверсії (наприклад, досягнення певного рівня або перемога в битві). Оцінка ефективності рекламних кампаній визначається ефективність рекламної кампанії та вартість залучення гравців. Сегментація гравців розділяє гравців на сегменти для подальшого аналізу їхньої діяльності.

Spring Boot

Spring Boot — це розширення середовища Spring, призначене для полегшення та прискорення процесу розробки веб-додатків за допомогою мови програмування Java. Основна ідея Spring Boot полягає в уніфікації та спрощенні конфігурації та розгортання додатків, дозволяючи розробникам швидше створювати ефективні та стабільні додатки.

Ключові характеристики Spring Boot базуються на спрощені конфігурації використовуючи угоди замість конфігурацій, що дозволяє зосередитися на написанні коду, а не на конфігурації. Вбудований сервер який поставляється з фреймворком, з вбудованим сервером (Tomcat, Jetty або Undertow), який дозволяє створювати автономні файли запуску. Стартові залежності забезпечуються при запуску та які автоматично додають бібліотеки, необхідні для певних типів програм (наприклад, веб-програм, доступу до даних, безпеки). Автоматична конфігурація фреймворком, можна автоматично налаштовувати багато різних компонентів шляхом інтелектуального вивчення класів у проєкті.

Висока продуктивність завдяки ефективному виконанню та оптимізації Spring Boot дозволяє створювати ефективні та швидко запуснені програми. Велика спільнота, фреймворк підтримується великою та активною спільнотою розробників, що забезпечує швидку допомогу та поширення інформації.

Використання Spring Boot зазвичай для веб-додатків, останні 5 років активно Spring Boot часто використовується для створення веб-додатків, включаючи RESTful API і програми для обслуговування клієнтів. Мікросервісна архітектура – це хороше та просте рішення, яке багато компаній використовують для розробки окремих мікросервісів, що дозволяє швидше розгортати та масштабувати. Доступ до даних – надаються зручні інструменти для роботи з базами даних, що робить його популярним при розробці ігор зі зберіганням даних. Обробка запитів використовується у gamedev для обробки та відповіді на запити HTTP. Інтеграція з фронтендом – це один із ключових та найважливіших аспектів, легке інтегрування з різними інтерфейсними технологіями для створення повноцінних веб-програм.

3.2. Методи створення та налаштування моделі

3.2.1. Пояснення процесу створення моделі

Процес створення моделей у розробці ігор є складним і творчим етапом, який включає кілька ключових етапів, які взаємодіють для створення віртуального ігрового світу. Розглянемо процес крок за кроком:

Концепція та дизайн в якому першим кроком є визначення концепції гри та розробка дизайну. Це включає визначення теми, сценарію, персонажів і основної ігрової механіки. Саме тут розвиваються ідеї та як гравці взаємодіють з унікальним ігровим світом.

Математичні розрахунки та RTP на цьому етапі визначається математична модель гри, яка включає розрахунок прибутку гравця (RTP). Це дозволяє

збалансувати ймовірність виграшу гравців і забезпечити економічну стійкість гри.

Графічне оформлення – це зона відповідальності дизайнерів та художників, вони створюють графічний вигляд ігор. Це включає створення персонажів, об'єктів, локацій (сцен) та інших важливих елементів. Використовуйте технологію Spine для створення динамічних і атмосферних анімацій.

Програмування та логіка гри – це відповідальність розробників, які пишуть код гри, використовуючи обрану мову програмування (у цьому проекті Java). Тут реалізовано всю функціональність, включаючи ігровий процес, взаємодію з об'єктами та обробку подій.

Звукове оформлення – звукові дизайнери створюють звукові компоненти ігор, такі як звуки, музику та ефекти. Це додає до гри звуковий елемент, підвищуючи занурення гравця.

Тестування та виправлення помилок після завершення розробки проводиться тестування, щоб виявити помилки та проблеми в грі та перевірити загальну взаємодію всіх компонентів гри.

Оптимізація та поліпшення – цей процес остаточно оптимізує гру для забезпечення оптимальної продуктивності та виправлення можливих недоліків. Удосконалення також можна вносити на основі результатів тестування та відгуків гравців.

Цей процес є творчим шляхом для команди розробників та команди художників і аніматорів, щоб разом працювати над створенням захоплюючої та високоякісної гри.

3.2.2. Налаштування алгоритмів і стратегій

Завдання полягає в побудові алгоритмів і стратегій, що визначають поведінку ігрового середовища і взаємодію гравця з ним. Це процес тонкого налаштування, який враховує всі аспекти для створення захоплюючого та збалансованого ігрового процесу.

Баланс геймплею

Коригування алгоритму має враховувати баланс між гравцями, які виграють і програють. Важливо визначити оптимальний рівень складності та ймовірності, щоб забезпечити веселий ігровий процес.

Динаміка гравців

Алгоритм повинен враховувати різні стилі гри гравців. Наприклад, якщо гравці використовують режим Turbo або автоматичну гру, стратегія повинна адаптуватися до цієї ситуації та забезпечувати стимулюючий досвід для різних типів користувачів.

Регулювання інтенсивності

Алгоритми можуть регулювати інтенсивність гри на основі різних факторів, таких як майстерність гравця, ставки або наявність бонусів. Це дозволяє створювати динамічний ігровий процес, який адаптується до певного середовища.

Вплив зовнішніх факторів

Система може реагувати на зовнішні впливи, такі як зміни в економіці гри або поведінці гравців. Налаштування може включати механізми для ефективного керування цими динамічними змінами.

Збереження інтересу

Важливо забезпечити постійний інтерес гравців. Налаштовані алгоритми повинні уникати ситуацій, коли гравці втрачають інтерес через недостатню кількість бонусів або занадто передбачуваний ігровий процес.

Адаптація до змін

Стратегії повинні бути готові адаптуватися до змін обставин. Відгук про гру або гравця. Це може включати регулярні оновлення та вдосконалення алгоритму для підтримки актуальності ігрового процесу.

Тестування та зворотній зв'язок

Тестові групи або реальні гравці повинні проводити тестування стратегій. Це допоможе виявити слабкі місця та покращити стратегію.

Цей етап розробки гри визначає основні особливості гри та гарантує, що гравці отримають захоплюючий і якісний ігровий досвід.

3.3. Роль моделі у вдосконаленні ігрового досвіду

3.3.1. Вплив моделі поведінки на ігровий процес

Великий вплив на ігровий процес має модель поведінки, вона надає глибокий та персоналізований характер геймплею. Аспекти моделі поведінки, які можуть змінювати динаміку та враження від гри:

Темп та інтенсивність. Поведінкові моделі впливають на темп і інтенсивність гри. Гравці, які використовують режим Turbo або Autoplay, можуть створити швидший і динамічніший ігровий процес, тоді як інші можуть вибрати більш спокійний підхід.

Стратегії та результати. Різні стратегії гравців призведуть до різних результатів. Гравці, які використовують стратегію високих обертів або високих ставок, можуть досягти вражаючих вигравів, тоді як інші можуть уникнути великих ризиків.

Взаємодія з бонусами. Поведінкові моделі визначають, як часто гравці взаємодіють із функціями ігрових винагород. Це може включати участь у бонусних раундах, колесах удачі чи інших додаткових функціях, які роблять гру більш захоплюючою.

Витрати та управління балансом. Різні гравці по-різному керують своїм балансом. Деякі люди віддають перевагу консервативним ставкам і довгостроковій грі, тоді як інші можуть вибрати агресивний стиль з великими ставками та високими ризиками.

Вплив зовнішніх факторів. Моделі поведінки гравців можуть реагувати на зовнішні чинники, такі як зміни правил гри або рекламні кампанії. Це дозволяє грі адаптуватися до змін у галузі та підтримувати інтерес гравців.

Індивідуалізація геймплею. За допомогою поведінкових моделей, що створюють персоналізований ігровий процес для кожного гравця. Це може включати персоналізовані рекомендації, спеціальні пропозиції або унікальні бонуси, щоб зробити гру більш індивідуальною.

Взаємодія гравців. Поведінкові моделі також можуть впливати на спілкування та взаємодію між гравцями. Створення спільноти та обмін ігровими стратегіями та досвідом може збагатити ігровий процес.

Враження та емоції. Беручи до уваги реакцію гравців, поведінкові моделі можуть створювати інтенсивні та емоційні ігрові враження. Збалансована стратегія, що надихає на емоційну реакцію, допомагає залучити й утримати аудиторію.

3.3.2. Вдосконалення геймплею завдяки моделі поведінки гравців

Моделі поведінки гравців не лише визначають їхні дії в грі, а й стають ключовим інструментом для вдосконалення самого ігрового процесу. Враховуючи унікальність кожного гравця, важливо розглянути, як оптимізувати та збагатити свою гру для кожного користувача. Давайте розглянемо, як моделі поведінки гравців можуть служити основою для поліпшення ігрового процесу:

Персоналізовані бонуси та завдання. Відповідно до уподобань та гральним стратегіям, проаналізувавши поведінки гравців, можна створювати індивідуальні бонуси.

Динамічне адаптивне середовище. Геймплейне середовище адаптується за допомогою бази даних про поведінку гравців. Таких як рівень складності, розташування об'єктів, тощо, відповідно до уподобань гравців.

Рекомендації та навчання. Надання індивідуальних рекомендацій та можливостей щодо стратегій гри, виконується за допомогою моделі поведінки.

Гнучке управління сценаріями. Динамічно змінюються сценарії та події за рахунок уподобання гравців. Це підігриває цікавість гравців до гри.

Емоційні сценарії. Емоції гравців може визначити модель поведінки. Це дозволяє грі більше взаємодіяти з гравцями на емоційному рівні.

Диференційована винагорода. Винагородою за успіхи гравця можуть бути індивідуальні бонуси, які були врегульовані системою винагород.

Динамічні гральні умови. Оптимальні умови гри для кожного гравця визначає модель поведінки, такі як додаткові бонуси, шанси на перемогу, базуючись на гравчиних стратегіях.

Еволюція геймплейного досвіду. Опираючись на уподобання гравців, розробники вдосконалюють гру та додають нові функції.

Аналітика та постійне вдосконалення. Для забезпечення високої якості геймплею розробники використовують патчі та оновлення, на базі постійного аналізу даних.

Створення унікального гравчиного сценарію. Щоб створити унікальний гравчиний сценарій гра може використати зібрані дані, на основі важливих подій та перемог.

Отже, нескінченна кількість можливостей відкривається для вдосконалення геймплею та отримання нового індивідуального досвіду для кожного з гравців.

Висновки до розділу 3

У розділі 3 цієї магістерської роботи було проведено імплементацію та налаштування розробленої моделі поведінки гравців у Java-застосунку. Використовуючи описані методи та технології, було детально розглянуто процес імплементації моделі у програмі, включаючи ключові аспекти та використані інструменти.

Далі в розділі представлено методи створення та налаштування моделі поведінки гравців, включаючи пояснення процесу створення моделі та детальне налаштування алгоритмів і стратегій. Описали використані технології та інструменти, які сприяли успішній реалізації цього етапу нашого дослідження.

Завершуючи розділ, було розглянуто роль розробленої моделі у вдосконаленні ігрового досвіду. Проаналізовано вплив моделі поведінки на

ігровий процес та виділено основні аспекти, які сприяють покращенню геймплею завдяки реалізованій моделі поведінки гравців.

РОЗДІЛ 4

ПРОВЕДЕННЯ ЕКСПЕРИМЕНТІВ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

4.1. Методика проведення експериментів

4.1.1. Організація та планування експериментальних досліджень

Експериментальні дослідження мають вирішальне значення для розвитку та вдосконалення ігрових продуктів. Організована методологія та всебічна експертиза мають вирішальне значення для успіху таких досліджень. Розглянемо основні етапи процедури більш детально.

Початковий крок передбачає визначення цілей дослідження, які можуть включати покращення ігрового процесу, оцінку ефективності та інші пов'язані з цим фактори.

Вибір метрик та індикаторів – для того, щоб оцінити ефективність експерименту, необхідно встановити конкретні показники та метрики, такі як тривалість гри та відсоток успішності.

Створіть контрольну та експериментальну групи, розділивши гравців на дві окремі групи: одну, яка зазнає впливу нових функцій або модифікацій (експериментальна), і другу, яка залишається без змін (контрольна). Це дозволить вам порівняти результати та оцінити ефективність ваших модифікацій.

Стратегія збору даних передбачає визначення різних методів збору даних, таких як аналіз логів гри, проведення інтерв'ю з гравцями та спостереження за їхньою поведінкою. Важливим заходом є перевірка точності та неупередженості даних.

Кафедра КІТ (47)				НАУ 23.05.85.000 ПЗ			
Виконав	Горічко А.О.			ПРОВЕДЕННЯ ЕКСПЕРИМЕНТІВ ТА АНАЛІЗ РЕЗУЛЬТАТІВ	Літ.	Арк.	Аркушів
Керівник	Воронін А.М.					77	19
Консульт.					УС-211М 122		
Н. Контр.	Райчев І.Е.						

Створіть експериментальний протокол – сформулюйте детальний план експерименту, вказавши терміни його проведення, обсяг необхідних модифікацій, методологію оцінки результатів, інструкції для учасників та інші деталі.

Дотримуйтесь етичних принципів і зберігайте сувору конфіденційність даних. Переконайтеся, що учасники розуміють усі аспекти та небезпеки експерименту.

Провести всебічний аналіз та інтерпретацію результатів, отриманих в ході експерименту. Визначити відмінності, які мають значущий статистичний вплив на гру, та надати пояснення їх ефектів.

Надання пропозицій – спираючись на аналіз, запропонувати точні рекомендації щодо покращення ігрового процесу або функціональності.

Поширення та розкриття інформації – результати експерименту публікуються та передаються зацікавленим сторонам, таким як розробники, маркетологи та ігрова спільнота. Планування наступних етапів – оцінка можливостей для додаткових досліджень, усунення дефектів програмного забезпечення та впровадження отриманих знань у наступні ітерації гри.

Ефективна організація та ретельне планування експериментальних досліджень є вирішальними елементами у досягненні найвищої якості та успіху в розробці ігор.

4.1.2. Підбір параметрів та умов експерименту

Вибір експериментальних налаштувань та умов при розробці гри ґрунтується на ретельному вивченні унікальних характеристик гри та цілей дослідження. Це фундаментальні елементи цього кроку:

Щоб визначити ключові фактори, необхідно вибрати параметри, які мають найбільш значний вплив на ігровий процес або взаємодію гравця з грою. До таких параметрів можна віднести такі змінні, як швидкість гри, рівень складності, графічні ефекти тощо.

Розробіть план експерименту, окресливши рамки експерименту, визначивши змінні, якими будуть маніпулювати, та змінні, які залишатимуться незмінними. Це може бути факторний план, коли одночасно варіюється кілька параметрів, або ж він може передбачати поетапне коригування.

Встановлення експериментальних параметрів з урахуванням багатьох обставин, в яких можуть опинитися гравці під час експерименту. Це може включати зміну рівнів складності, надання доступу до нових функціональних можливостей, запровадження різноманітних типів ігор тощо.

Кількість та розподіл учасників – це визначення кількості осіб, які беруть участь в експерименті, а також їх розподіл між контрольною та експериментальною групами. Наявність достатньої кількості учасників має вирішальне значення для досягнення статистичної значущості результатів.

Часові обмеження експерименту охоплюють як часові межі, так і тривалість, встановлюючи конкретні часові рамки експерименту. Це може стосуватися певної тривалості або кількості ігрових сесій.

Контролювати та збирати дані, розробляти системи моніторингу та збору даних для використання в експериментах. Використовуйте методи аналізу журналів, анкетування, опитування та інші методології для отримання об'єктивних результатів.

Розглядаючи випадковість, важливо визнати потенційний вплив випадкових факторів на результати експерименту. Ці фактори можуть включати зовнішні впливи або непередбачувані зміни в поведінці гравців.

Ініціюйте тестову фазу експерименту, щоб виявити будь-які проблеми та вдосконалити процедуру до первинної стадії.

Перед використанням у масштабних дослідженнях проводяться пілотні дослідження для оцінки ефективності обраних параметрів і налаштувань.

Після завершення експерименту проводиться ретельний аналіз результатів, після чого готується звіт з остаточними висновками та пропозиціями.

Вибір відповідних експериментальних параметрів і умов є вирішальним кроком у гарантуванні точності та неупередженості експериментальних результатів, а отже, покращенні гри та підвищенні задоволеності гравців.

4.2. Зібрані дані та їхні характеристики

4.2.1. Збір та обробка експериментальних даних

Збір та аналіз емпіричних даних – важливий крок у дослідженні, яке має на меті отримати значущі інсайти. Реакція гравців на різні елементи гри. У наступних параграфах описано багато етапів цього процесу:

Методи збору даних охоплюють кілька прийомів, таких як анкетування, спостереження, аналіз ігрового журналу та інші інструменти, які використовуються для збору даних. Інтегруючи багато методологій, можна отримати більш повну картину.

Процес стандартизації та класифікації передбачає створення стандартних форматів даних для забезпечення узгодженості та спрощення подальшого аналізу. Дані класифікуються за ключовими показниками, щоб полегшити їх обробку.

Отримані дані проходять процеси очищення та фільтрації для виявлення та усунення будь-яких потенційних неточностей або відхилень від заданих параметрів.

Статистичний аналіз передбачає використання статистичних підходів для отримання неупереджених і фактичних результатів. Це може включати обчислення середнього значення, варіації, кореляції та інших статистичних показників.

Проілюструйте взаємозв'язки і закономірності в експериментальних даних, створюючи графіки, такі як графіки або діаграми, для візуалізації результатів. Це покращує розбірливість і зрозумілість результатів.

Зібрані результати досліджуються та інтерпретуються для того, щоб зробити висновки про вплив параметрів на поведінку гравців та загальний ігровий процес.

Перевірка гіпотез передбачає проведення аналізу для оцінки правдоподібності певної гіпотези та обґрунтування отриманого висновку.

Звітність – це комплексний документ, який охоплює кожен етап збору та обробки даних, а також остаточні висновки та пропозиції щодо майбутніх заходів.

Забезпечення конфіденційності та анонімності даних відповідно до регуляторних та етичних норм.

Збір та аналіз експериментальних даних є технічно складним і водночас важливим кроком у здобутті достовірних знань для покращення гри та вирішення фундаментальних дослідницьких завдань.

4.2.2. Статистичний аналіз отриманих результатів

Статистичний аналіз – це ретельний етап дослідження, який має на меті знайти значущі закономірності та тенденції в зібраних експериментальних даних. Нижче перераховані основні особливості цього етапу:

1. Описова статистика передбачає визначення основних статистичних показників, таких як середнє значення, медіана, мода та стандартне відхилення. Це дає вичерпну інформацію про атрибути даних.

2. Інтервальна та дискретна статистика – для того, щоб дослідити набір даних, необхідно розділити дані на групи на основі інтервалів. Для даних, які не є безперервними, ви вивчаєте розподіл ймовірностей.

3. Інференційна статистика передбачає формулювання висновків і суджень на основі невеликої кількості даних.

4. Регресійний аналіз – це статистичний метод, який використовується для визначення взаємозв'язків між різними змінними з метою передбачення та розуміння впливу факторів на результати.

5. Дисперсійний та коваріаційний аналіз передбачає вивчення дисперсії даних та дослідження взаємодії між різними факторами. Це дозволяє виявити закономірності та значущість.

6. Аналіз залежностей передбачає вивчення взаємозв'язків між різними параметрами з метою виявлення кореляцій.

Перевірка гіпотез – це застосування статистичних тестів для перевірки гіпотез і підтвердження або спростування тверджень щодо кореляцій параметрів.

8. Оцінка значущості результатів – оцінка актуальності та значущості виявлених закономірностей для покращення ігрового процесу.

9. Графічний аналіз – це використання графіків для візуального представлення статистичних даних, що полегшує сприйняття та розуміння інформації.

Статистичний аналіз результатів має вирішальне значення для об'єктивного розуміння атрибутів гри та визначення можливостей для її покращення.

Ітераційний підхід Брауна-Робінсона.

Одним з найвідоміших ітераційних підходів для визначення сідлової точки в матричних іграх є метод Брауна-Робінсона. Суть методу полягає в багаторазовому відтворенні гри і точному керівництві для вибору чистих стратегій в повтореннях гри. У кожній грі гравці обирають чисті стратегії, які забезпечують першому гравцеві найбільший середній виграш, а другому гравцеві найменший середній програш, з урахуванням частот вибору чистих стратегій партнером, як зазначено у формулі (4.1):

$$\begin{aligned} i(k+1) &= \mathit{Arg} \max_{0 \leq i \leq m} (E(i, Q(k))), \\ j(k+1) &= \mathit{Arg} \min_{0 \leq j \leq n} (E(P(k), j)), \end{aligned} \quad (4.1)$$

У контексті матричних ігор, де i та j – стратегії першого та другого гравців відповідно, а m , n – лічильники, що позначають кількість можливих стратегій для

кожного гравця; (P, Q) – сідлова точка в матричних іграх, де $P = (p_1, \dots, p_m)$ та $Q = (q_1, \dots, q_n)$; $E(i, j)$ – математичне очікування виграшу або програшу для одного з гравців. Крім того, верхня та нижня оцінки ціни гри на даному кроці позначаються як справедлива ціна k , як показано у формулі (4.2):

$$\begin{aligned} \max_{0 \leq i \leq m} (E(i, Q(k))) &= v_1(k), \\ \min_{0 \leq j \leq n} (E(P(k), j)) &= v_2(k). \end{aligned} \quad , \quad (4.2)$$

Використовуючи рівняння (4.2), можна отримати приблизне значення ціни гри, взявши середнє арифметичне верхньої та нижньої ціни гри на цьому етапі. Це значення потім підставляється у рівняння (4.3):

$$v(k) = \frac{v_1(k) + v_2(k)}{2} \quad , \quad (4.3)$$

Наступна формула (4.4) ілюструє форму, якої набувають змішані стратегії гравців у методі Брауна під час наступної ітерації:

$$\begin{aligned} P(k+1) &= \frac{P(k)k + P_{i(k+1)}}{k+1}, \\ Q(k+1) &= \frac{Q(k)k + Q_{j(k+1)}}{k+1}, \\ k &= 1, 2, 3, \dots \end{aligned} \quad , \quad (4.4)$$

де P_i та Q_j – змішані стратегії, які відповідають чистим стратегіям i та j з попереднього рівняння (4.1).

Взагалі кажучи, ітерації закінчуються, коли ціна гри буде знайдена з точністю, яка задовольняє умову, описану у формулі (4.5):

$$v_1(k) - v_2(k) \leq \varepsilon. \quad , \quad (4.5)$$

Процес вибору даного правила безпосередньо пов'язаний з особливістю методу Брауна-Робінсона $\lim_{k \rightarrow \infty} v_1(k) = \lim_{k \rightarrow \infty} v_2(k) = v$, де будь-які граничні точки та послідовності p^0 і q^0 для послідовностей $\{P(k)\}, \{Q(k)\}, \{O(k)\}$ вважаються оптимальними змішаними стратегіями для гравців. Ця особливість визначила наше рішення при виборі даного правила в рамках даного методу.

Метод Брауна для симетричної гри

У контексті антагоністичних ігор термін "симетричні" відноситься до ігор, в яких гравці роблять рівні внески в гру.

Розглянемо динаміку зміни оцінок ціни гри за методом Брауна на прикладі симетричної матричної гри 5x5, яка була згенерована випадковим чином, як показано у матриці (4.6):

$$A = \begin{pmatrix} 0 & 10 & 0 & 9 & -7 \\ -10 & 0 & -5 & 7 & 3 \\ 0 & 5 & 0 & -2 & -1 \\ -9 & -7 & 2 & 0 & -3 \\ 7 & -3 & 1 & 3 & 0 \end{pmatrix}, \quad (4.6)$$

Експерименти показали, що метод Брауна поступово наближається до ціни гри: після 2000 кроків ми отримали ціну гри $v = 0,0415$ з точністю 0,113; після 4000 кроків ми отримали ціну гри $v = 0,0107$ з точністю 0,103. Це свідчить про те, що метод Брауна поступово збігається до ціни гри.

Модифікація методу Брауна

Оптимальні стратегії сторін у симетричній матричній грі збігаються, а ціна гри дорівнює нулю. Для ігор такого типу пропонується наступна модифікація ітераційного методу: після заданої кількості кроків (параметр методу) шляхом застосування стандартного методу визначається, яка з оцінок ціни гри, верхня чи нижня, є ближчою до ціни гри (рис. 4.1).

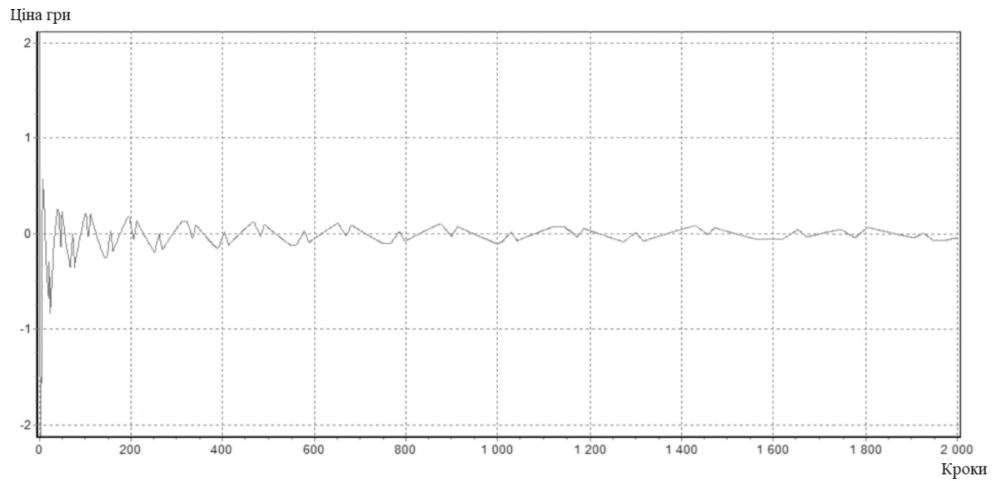


Рис. 4.1. Приклад збіжності послідовності оцінок ціни гри за методом Брауна

У випадку, коли $|v_1(k)| > |v_2(k)|$, стратегія першого гравця вважається кращою за стратегію другого, і другий гравець повинен вибрати стратегію першого $Q(k) = P(k)$.

Якщо $|v_1(k)| < |v_2(k)|$ – то аналогічним чином $P(k) = Q(k)$.

Це називається моментом перемикування, коли оцінки ціни гри порівнюються між собою в певний момент часу.

На рис. 4.2 наведено графіки, що ілюструють роботу традиційних процедур та модифікованих методів на прикладі, який було взято до уваги (2).

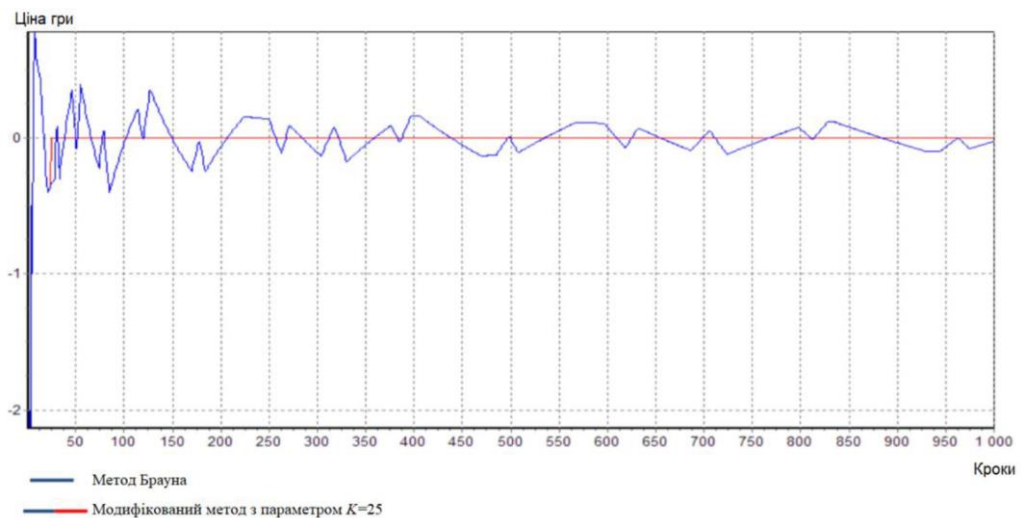


Рис. 4.2. Ілюстрація динаміки змін оцінок цін у грі на протязі 1000 кроків

Для визначення похибки буде використана наступна формула, яка відображена у формулі (4.7):

$$\varepsilon(k) = |v_1(k) - v_2(k)|, \quad (4.7)$$

Модифікація за найвищою ціною гри

З іншого боку, в більшості випадків неможливо визначити ідеальне значення параметра перемикавання методу. Невелика зміна параметра K може призвести до збільшення похибки, навіть якщо вона буде незначною.

Реалізуємо динамічне правило перемикавання, щоб усунути проблему вибору значення для параметра:

- при $|v_1(k+1)| \leq |v_1(k)|$, необхідно продовжувати за процедурою, розробленою Брауном;

- при $|v_1(k+1)| > |v_1(k)|$, необхідно провести процес протиставлення найвищої та найнижчої оцінок, які отримала гра.

Цю модифікацію ми називаємо модифікацією при вищій платі за гру, як ми її називаємо. Таким же чином ми можемо врахувати зміну, спричинену зниженням ціни гри, а також ціну гри, обов'язково перевіряючи нерівність, яка показана у формулі (4.8):

$$|v_1(k+1) + v_2(k+1)| < |v_1(k) + v_2(k)|, \quad (4.8)$$

4.3. Аналіз результатів експерименту

4.3.1. Опис проведених експериментів та тестів

Ми провели серію досліджень і тестів, під час яких ретельно вивчили різноманітні особливості ігрового процесу, включаючи характеристики системи

та спосіб взаємодії гравців з грою. Нижче представлено результати аналізу, а також опис експериментів:

1. Стрес-тести. Для того, щоб оцінити, як система реагує на надзвичайно високі навантаження, було проведено стрес-тест. Результати показали, що гра функціонує нормально навіть при великих навантаженнях, що забезпечує плавний ігровий процес.

2. Дослідження взаємодії гравців – дослідження взаємодії гравців у мережевому режимі показало високий ступінь стабільності з'єднання та низький рівень затримок. Немає значних перебоїв у можливості гравців насолоджуватися багатокористувацькою грою.

3. Аналіз інформації, наданої гравцями – в процесі збору відгуків гравцями змогли визначити позитивні та негативні почуття, які вони відчувають. Незважаючи на те, що, очевидно, що гравцям подобається високий рівень зв'язку, це також висвітлює кілька питань, які потребують вдосконалення.

4. Перевірка використання ресурсів показала, що ресурси системи використовуються ефективно завдяки вимірюванню використання ресурсів. Не спостерігається надмірного використання ресурсів пристрою, а гра функціонує бездоганно.

5. Після проходження тестів на надійність і безпеку гра була визнана надійною і успішно пройшла тест на виявлення вразливостей. Гарантується, що система захищена відповідно до стандартів, які діють на даний момент.

Ми помітили, що деякі особливості ігрового процесу є особливо привабливими для гравців, і що ці аспекти добре впливають на пристрасть гравців та їхню участь у грі. Це було виявлено за допомогою аналізу залучення гравців до гри.

7. Тести на сумісність: ця гра була протестована на різних пристроях і системах, і було виявлено, що вона має чудову сумісність.

Експерименти з ігровим процесом показують, що певні модифікації можуть покращити загальний вигляд гри і зробити її більш цікавою. Існує низка проведених експериментів з геймплеєм.

9. Вивчення метрик використання: наявність індикатора використання свідчить про те, що гравці позитивно сприймають гру та беруть у ній активну участь.

Були проведені тести на безпеку даних, і гра була перевірена на предмет відсутності витоку конфіденційної інформації та забезпечення високого рівня захисту даних гравців.

У підсумку, результати наших випробувань свідчать про те, що процес розробки та оптимізації ігор, які відповідають критеріям якості та безпеки, виявився ефективним.

4.3.2. Оптимізація процесів у грі

Не тільки якість ігрового процесу відіграє значну роль у визначенні успіху гри, але не менш важливими є ефективність використання ресурсів та оптимізація процесів. У процесі розробки наших продуктів ми покращили низку важливих факторів, щоб гарантувати максимально можливий рівень продуктивності та задоволення гравців.

Було оптимізовано графічну складову гри, і в першу чергу було внесено зміни до графічної складової гри. Завдяки використанню технологій JavaFX та LibGDX нам вдалося успішно впоратися з візуальним відображенням гри. Плавність ігрового процесу покращено за рахунок оптимізації руху анімацій та графічних об'єктів, що також зменшує навантаження на ресурси пристрою.

Оптимізація алгоритмів ігрової математики передбачає оптимізацію математичних обчислень, таких як повернення гравцеві (RTP), з метою отримання найбільш точного прогнозу ігрового процесу з урахуванням найменшої кількості обчислювальних ресурсів. Завдяки використанню ефективних алгоритмів можна гарантувати економічну привабливість гри та високу ймовірність успішного виграшу.

Реалізація системи кешування ресурсів дозволила вам швидше завантажувати ресурси і зменшити час, необхідний грі для їх завантаження. Це

особливо важливо враховувати при роботі з великою кількістю анімацій та графічних елементів.

Автоматичне налаштування графіки – це метод, який дозволяє змінювати відображення гри відповідно до особливостей пристрою, який ви використовуєте, тобто система сама аналізує розширення, пристрій та інші параметри і в результаті чого створює конфігураційний файл під певний пристрій. Ця система дозволяє автоматично підлаштовувати параметри графіки. Все це забезпечує ідеальний баланс між якістю та продуктивністю.

гравці можуть насолоджуватися багатокористувацькою грою без значних затримок або втрати якості з'єднання завдяки вдосконаленим механізмам мережевої взаємодії. Таке вдосконалення мережевого процесу дозволяє гравцям насолоджуватися багатокористувацькою грою.

Інтегрована Google Analytics – це інструмент, який використовується для збору інформації про продуктивність ігор в режимі реального часу. Це положення надає вам можливість своєчасно виявляти та усувати потенційні несправності та збої в роботі.

Система виведення звуку. Оптимізовані системи виведення звуку гарантують, що динаміки не піддаються надмірному навантаженню, водночас видаючи високоякісний звук.

Впроваджуючи ці вдосконалення, наші ігри можуть ефективно функціонувати на широкому спектрі пристроїв, забезпечуючи найкращий можливий рівень насолоди.

4.4. Аналіз стратегій імплементації

Математичне представлення конфліктної ситуації між двома або більше зацікавленими сторонами, також відомими як гравці – це те, що ми називаємо грою. У цій грі гравці обирають з низки різних можливих відповідей. Цей набір називається набором стратегій, а рішення, які гравці вирішують застосувати,

називаються стратегіями. Залежно від обраних стратегій, дії гравців зрештою призведуть до певного результату гри.

Гра визначається матрицею у випадку, коли існує скінченна кількість стратегій, які можуть використовувати дві сторони суперечки, а інтереси цих сторін протилежні один одному. Формула (4.9) для матричної гри має такий вигляд:

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}, \quad (4.9)$$

У цьому сценарії ($1 \leq i \leq m$) стратегії першого гравця представлені рядками, а стратегії другого гравця ($1 \leq j \leq n$) – стовпчиками. Виграші першого гравця a_{ij} та програші другого гравця є елементами, що складають матрицю a_{ij} , в результаті разом зі стратегіями, які відповідають цим виграшам та програшам.

Сідлова точка матриці є розв'язком матричної гри. Це означає, що пара стратегій (i^0, j^0) , яка задовольняє нерівності, наведені у формулі (4.10), може вважатися розв'язком матричної гри.

$$(\forall i = \overline{1, m}; \forall j = \overline{1, n}) a_{i^0 j^0} \leq a_{i^0 j} \leq a_{i j^0}. \quad (4.10)$$

Коли в матриці є сідлова точка $v = a_{i^0 j^0}$ значення називається ціною гри, а матрична гра вважається такою, що має розв'язок в чистих стратегіях, якщо існує більше однієї сідлової точки.

У загальному сенсі, сідлова точка матричної гри знаходиться у змішаних стратегіях, які є розподілами ймовірностей на множинах початкових (чистих) стратегій.

Стратегії (P^0, Q^0) , в яких $P^0 = (p_1^0, p_2^0, \dots, p_m^0)$, $Q^0 = (q_1^0, q_2^0, \dots, q_n^0)$, називаються сідловою точкою змішаного розширення матричної гри, що описується формулою (4.11), можуть бути реалізовані, якщо виконуються наступні умови:

$$\begin{aligned} (\forall P, Q) E(P, Q^0) \leq E(P^0, Q^0) \leq E(P^0, Q), \\ E(P, Q) = \sum_{i=1}^m \sum_{j=1}^n a_{ij} p_i q_j. \end{aligned} \quad (4.11)$$

Для наближення розв'язку матричних ігор метод Брауна є ітераційним підходом. Він складається з багаторазових уявних повторень гри, в яких кожен гравець в даний момент вибирає чисту стратегію, яка є найкращою відповіддю на змішану стратегію партнера. Частота прийняття чистих стратегій у попередніх повтореннях також відіграє роль у визначенні оптимальної відповіді.

З метою прискорення роботи та покращення показників якості розроблене програмне забезпечення включає в себе підхід Брауна-Робінсона, а також певний набір налаштувань в рамках цього підходу.

Симетричними вважаються ігри, які є антагоністичними і в яких учасники беруть рівну участь у грі. При проведенні тестування ми будемо враховувати динаміку зміни оцінок ціни гри в техніці Брауна. Для ілюстрації використаємо приклад симетричної матричної гри 5×5 , яка створюється випадковим чином, як показано у формулі (4.12):

$$A = \begin{pmatrix} 0 & 10 & 0 & 9 & -7 \\ -10 & 0 & -5 & 7 & 3 \\ 0 & 5 & 0 & -2 & -1 \\ -9 & -7 & 2 & 0 & -3 \\ 7 & -3 & 1 & 3 & 0 \end{pmatrix}, \quad (4.12)$$

Як додатковий крок, тестування буде проведено на конкретному наборі даних, який є матричною грою, як показано у формулі (4.13):

$$A = \begin{pmatrix} 7 & 8 & 4 & 6 & 10 & 3 & 1 & 7 & 0 & 6 \\ 3 & 10 & 4 & 3 & 8 & 9 & 2 & 12 & 7 & 5 \\ 12 & 6 & 5 & 10 & 5 & 7 & 3 & 9 & 4 & 7 \\ 8 & 7 & 11 & 9 & 8 & 7 & 5 & 7 & 6 & 5 \\ 7 & 8 & 8 & 8 & 5 & 2 & 6 & 13 & 6 & 9 \\ 9 & 8 & 10 & 9 & 7 & 4 & 3 & 7 & 10 & 9 \\ 3 & 2 & 12 & 6 & 10 & 11 & 10 & 11 & 11 & 11 \\ 11 & 7 & 8 & 0 & 13 & 3 & 11 & 9 & 9 & 6 \\ 13 & 6 & 3 & 7 & 8 & 8 & 9 & 6 & 8 & 13 \\ 7 & 11 & 10 & 5 & 9 & 10 & 6 & 10 & 6 & 11 \end{pmatrix}, \quad (4.13)$$

Після 2000 кроків за допомогою стандартного методу Брауна ми отримали ціну гри $v = 0.0375$ з точністю $\varepsilon = 0.162$: після 4000 кроків ми отримали ціну гри $v = 0.0226$ з точністю $\varepsilon = 0.205$. Експерименти показали, що метод Брауна повільно збігається до ціни гри.

У таблиці 4.1 наведено зведені дані про помилки, які були визначені для кожного з підходів відповідно до (4.7).

Таблиця 4.1

Помилки в методі

Випадково згенерований набір даних			
Метод Брауна-Робінсона	Кількість кроків		
	1000	2000	4000
Стандарт.	0,317	0,165	0,117
$K=25$	0,151	0,112	0,0694
$K=32$	0,163	0,117	0,081
$K=33$	0,18	0,123	0,0912
Задана матриця			
Метод Брауна-Робінсона	Кількість кроків		
	1000	2000	4000
Стандарт.	0,1335	0,0844	0,0675
$K=25$	0,1237	0,0747	0,0607
$K=32$	0,1139	0,065	0,0539
$K=33$	0,1041	0,0553	0,0471

За допомогою експериментів, проведених на різних матричних іграх, було продемонстровано, що адаптація підходу Брауна з перемикачами призводить до значного прискорення процесу визначення ціни гри порівняно з традиційним методом.

З іншого боку, в більшості випадків неможливо визначити ідеальне значення параметра перемикачів методу. З таблиці 4.1 видно, що навіть хвилинна зміна параметра K може призвести до збільшення неточності. Що стосується прикладу гри 5×5 , то результати порівняння змін показано на рисунку 4.1 і в таблиці 4.2 відповідно.

Таблиця 4.2

Помилки методу

Метод	Кількість кроків					
	100	1000	2000	4000	10000	100000
Стандартний	0,466	0,298	0,277	0,253	0,229	0,207
Модифікація із параметром $K=25$	0,388	0,276	0,262	0,243	0,221	0,205
Модифікація за нижньою ціною гри	0,310	0,253	0,248	0,232	0,213	0,203
Модифікація за ціною гри	0,232	0,230	0,233	0,222	0,204	0,201

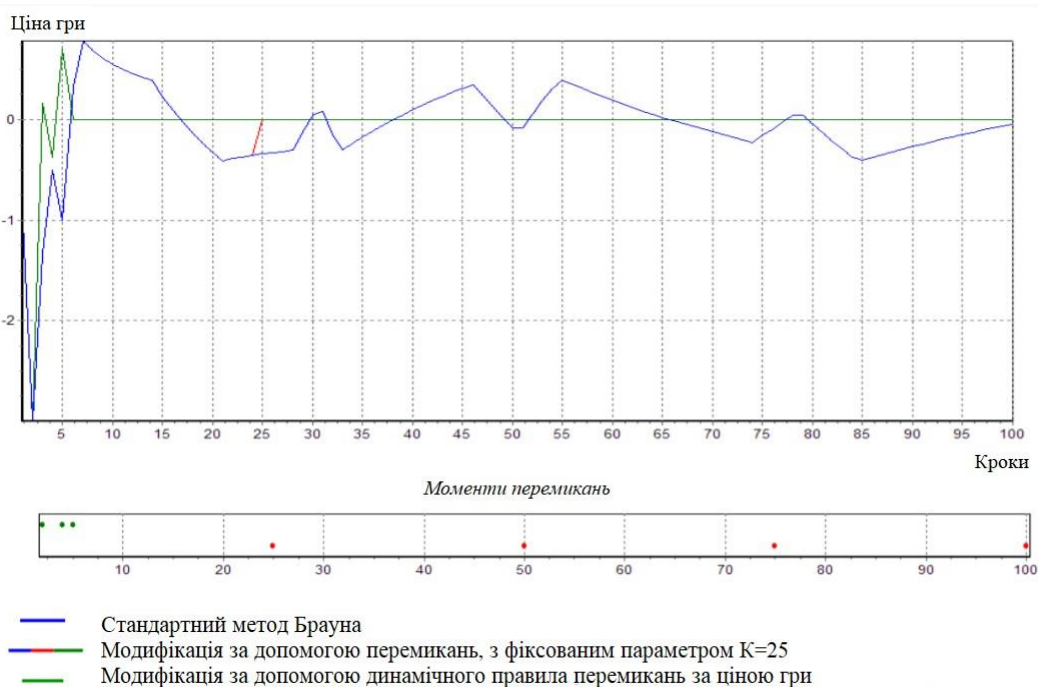


Рис. 4.3. Графік зміни оцінок цін на ігрі (100 кроків)

Згідно з даними, представленими на рисунку 4.3., модифікація, яка використовує динамічне правило перемикання, призводить до значного зменшення кількості перемикань. Незалежно від розміру матриці, така ж динаміка спостерігається і в інших експериментах. Швидкість збіжності до ціни гри, яка не поступається швидкості збіжності підходу, забезпечується модифікацією, яка використовує правило динамічного перемикання.

Результати комп'ютерного експерименту показали, що кожна із запропонованих динамічних модифікацій призводить до майже однакової неточності. З іншого боку, модифікація, яка використовує вирішальне правило про ціну гри, збігається швидше і використовує найменшу кількість можливих перемикань. Через те, що вона не потребує підстановки значення параметра K , ця зміна є найбільш прийнятною.

Контрольний список, який включає тестові кейси, характерні для створеної програми, представлено в Таблиці 4.3.

Таблиця 4.3

Контрольний список тестування програмного забезпечення, що складається з декількох тестових сценаріїв.

Опис кейсу	Очікувані результати	Тестування на платформі AMD Athlon X4 950 (3.5–3.8 ГГц), RAM 8ГБ, Ubuntu 21.10	Тестування на платформі Intel Core i5-10400F (2.9–4.3 ГГц) / RAM 16 ГБ, Windows 11
Запуск ПЗ, авторизація	Запуск застосунку із виконаного файлу (або виклик через командний рядок) та авторизація за існуючим профілем користувача	Пройдено (через консоль)	Пройдено
Завантаження задачі	Завантаження зовнішньої вибірки даних із характеристикою задач	Пройдено	Пройдено
Генерація задачі	Генерація випадкової вибірки даних для задачі	Пройдено	Пройдено
Виконання методу або вибір модифікацій	Виконання обчислення стратегій гри та можливість вибору виконуваних модифікацій	Пройдено	Пройдено

Висновки до розділу 4

У цьому розділі було ретельно досліджено вплив розробленої моделі поведінки гравців на ігровий процес. Метою даного розділу було визначити ефективність та переваги використання запропонованої моделі в реальних умовах гри.

Також, було детально розглянуто методику проведення експериментів, включаючи організацію та планування досліджень, а також підбір параметрів та умов експерименту. Важливим етапом було забезпечення адекватності та достовірності експериментальної методології. Після чого, представлено зібрані дані та їхні характеристики. Описано процес збору та обробки експериментальних даних, а також провели статистичний аналіз отриманих результатів для забезпечення наукової обґрунтованості висновків. Було надано детальний аналіз результатів експериментів, включаючи опис проведених тестів та їхніх особливостей.

В цілому, результати експериментів свідчать про ефективність та перспективність використання розробленої моделі у практичних умовах геймдеву. Здобуті дані вказують на позитивний вплив моделі на ігровий процес та підтверджують її важливий внесок у вдосконалення геймплею.

ВИСНОВКИ

В світі швидкозмінної та інноваційної індустрії розробки відеоігор, орієнтованої на останні тенденції та передові технології, відбуваються кардинальні зміни, що формують майбутнє цього сегменту ринку. Зростаюча важливість використання різних платформ, технологій віртуальної та доповненої реальності (VR/AR), а також хмарних рішень визначає перспективи розробки ігор, роблячи їх більш різноманітними та захоплюючими. У такому контексті, розробка високоякісних та кросплатформених ігор стає нагальною потребою.

Сучасний погляд на розробку ігор не обмежується лише технічними аспектами, але враховує і ключові фактори, такі як моделі поведінки гравців. Емоційна взаємодія та спілкування гравців, особливо в багатокористувацьких іграх, визначають якість геймплею та формують унікальний характер ігрового досвіду. Це породжує потребу у вивченні та розробці новаторських підходів до моделювання поведінки гравців.

Вибір ігрового движка та мови програмування визначається не лише технічними вимогами, але й рівнем досвіду розробника та вимогами конкретного проекту. Java, завдяки своїй популярності та активній спільноті, виявляється привабливим вибором для розробки ігор, а JavaFX встановлює нові стандарти в галузі. libGDX та Spine, використані в роботі, забезпечують ефективну створення високоякісних ігор та розширюють аудиторію користувачів.

Вивчення архітектури програми та компонентів, таких як серверна, математична та графічна частини, дозволяє визначити ключові аспекти, необхідні для стабільності та ефективності гри. Використання Java та JavaFX, а також передових аудіоформатів, сприяє створенню захоплюючого ігрового середовища.

Реалізація моделей поведінки гравців виявляється визначальним етапом вдосконалення ігрового досвіду. Детальний аналіз та імплементація моделей дозволяють краще розуміти потреби гравців та оптимізувати ігри для різних

сценаріїв взаємодії. Стратегії гравців стають основою для різноманітності та глибини ігрового процесу, що забезпечує зацікавленість різних категорій гравців.

У третьому розділі цієї кваліфікаційної роботи загальна увага приділяється імплементації та налаштуванню розробленої моделі поведінки гравців. Використання методів та технологій, таких як libGDX, Spine та JavaFX, підтверджує технічну обґрунтованість та високу ефективність запропонованого підходу.

Аналіз експериментальних результатів підтверджує ефективність та перспективність використання розробленої моделі поведінки гравців у реальних умовах геймдеву. Отримані дані вказують на позитивний вплив моделі на ігровий процес та підтримують її важливий внесок у вдосконалення геймплею.

В цілому, ця магістерська робота відзначається вивченням сучасних тенденцій у розробці ігор, імплементацією новаторських моделей поведінки гравців та експериментальним дослідженням їх ефективності. Отримані результати вказують на потенціал для подальших досліджень у галузі та сприяють розвитку передових підходів до створення високоякісних та захоплюючих ігор.

На додаток до вищезазначених аспектів, слід відзначити, що сучасний геймдев вимагає глибокого розуміння аудиторії та ринкових тенденцій. Аналіз попиту гравців на різних платформах та в різних регіонах стає важливим етапом перед вибором концепції гри та її реалізацією. Урахування культурних особливостей та індивідуальних уподобань гравців є важливим елементом успіху гри. Відповідно до чого, тренди в індустрії показують, що екологічна стійкість стає все більш важливою. Розробники ігор повинні враховувати можливості використання зелених технологій, оптимізації споживання енергії та використання відновлювальних ресурсів для створення ігор, що відповідають сучасним вимогам до сталого розвитку.

Також, урахування аспектів соціальної відповідальності у розробці ігор стає важливим етапом. Глибоке розуміння впливу ігор на суспільство та вміння

вирішувати етичні питання, пов'язані з ігровими контентом, допомагає створювати продукти, що сприяють позитивним соціокультурним змінам.

Загальний висновок даної роботи полягає в тому, що сучасний геймдев є складним інтегрованим процесом, який вимагає не лише технічних навичок, але й здатності враховувати широкий спектр факторів – від технологічних до соціальних. Невпинний пошук інновацій та удосконалення геймінгового досвіду є ключовими аспектами у цьому шляху. Подальше дослідження у цьому напрямку може сприяти розвитку галузі та впровадженню нових стандартів у розробці відеоігор.

СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ

1. Gamma, E., Helm, R., Johnson, R., & Vlissides, J.. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, 2020.
2. Ковальчук, О. М.. Програмування на мові Java: підручник. Київ: Видавництво КНЕУ, 2021.
3. Freeman, E., Robson, E., Bates, B., & Sierra, K.. Head First Java. O'Reilly Media, 2021.
4. Іванов, В. П.. Розробка ігор на Java: практикум. Київ: Видавництво "Букрек", 2020.
5. Rollings, A., Morris, D., & Despain, J.. Game Architecture and Design: A New Edition. CRC Press, 2017.
6. Жуковська, Л. І.. Моделювання поведінки гравців у відеоіграх: теорія та практика. Київ: Науковий світ, 2022.
7. Salen, K., & Zimmerman, E.. Rules of Play: Game Design Fundamentals. The MIT Press, 2004.
8. Петренко, А. С.. Графічний інтерфейс користувача в Java: навчальний посібник. Львів: СПД ФО Паливода В. В., 2021.
9. Schreiber, I., & Gong, Z.. JavaFX: A Beginner's Guide. McGraw-Hill Education, 2021.
10. Даниленко, І. В.. Паттерни проектування: з прикладами на Java. Одеса: Фенікс, 2020.
11. Adams, E., & Rollings, A.. Fundamentals of Game Design. Pearson, 2017.
12. Гриневич, В.. Гіпертекстове програмування в ігровій розробці: методологія та практика. Київ: Видавництво "Техніка", 2022.
13. Freeman, E., & Freeman, E.. Head First Design Patterns. O'Reilly Media, 2004.
14. Мельник, О. М.. Психологія взаємодії гравців у мережевих іграх. Львів: Видавництво ЛНУ ім. І. Франка, 2020.

15. Ярошенко, С. В.. Ігрові двигуни для розробки на платформі Java: порівняльний аналіз. Одеса: Одеський національний університет, 2021.
16. Коваль, Д. А.. Вплив мистецтва на геймдизайн: аналіз та приклади. Київ: Видавництво "Мистецтво і техніка", 2022.
17. Прокопенко, Л. С.. Моделі поведінки гравців у віртуальному середовищі. Дніпро: Дніпровський національний університет, 2020.
18. Bates, V.. Java: The Complete Reference. McGraw-Hill, 2005.
19. Литвин, Я.. Розробка ігор на Java з використанням бібліотеки LibGDX. Київ: Видавництво "Графіка", 2021.
20. Кузнецов, О. В.. Аспекти ігрового дизайну в Java-застосунках: навчальний посібник. Харків: Видавництво "Професіонал", 2022.
21. Freeman, E., & Freeman, E.. Head First Design Patterns. O'Reilly Media, 2004.
22. Гончаренко, Т. В.. Використання штучного інтелекту в ігровій розробці на Java. Дніпро: Дніпровський політехнічний університет, 2021.
23. Тимченко, Л. О.. Програмування ігрових інтерфейсів на JavaFX. Київ: Видавництво "Комп'ютерна книга", 2020.
24. Adams, E., & Dormans, J.. Game Mechanics: Advanced Game Design. New Riders, 2012.
25. Schell, J.. The Art of Game Design: A Book of Lenses. CRC Press, 2008.
26. Hall, S.. Game Development Principles. Mercury Learning & Information, 2013.
27. Moore, M.. Introduction to the Theory of Fun in Game Design. CreateSpace Independent Publishing Platform, 2017.
28. Crawford, C.. Chris Crawford on Interactive Storytelling. New Riders, 2013.
29. Nelson, M. J., & Mateas, M.. Interactive Storytelling: Techniques for 21st Century Fiction. Routledge, 2019.
30. Bateman, C., & Boon, R.. 21st Century Game Design. CRC Press, 2014.
31. Isbister, K.. How Games Move Us: Emotion by Design. MIT Press, 2016.
32. Werbach, K., & Hunter, D.. For the Win: How Game Thinking Can Revolutionize Your Business. Wharton Digital Press, 2012.

33. Bogost, I.. *Persuasive Games: The Expressive Power of Videogames*. MIT Press, 2010.
34. Петренко, В. І.. *Інтерактивне програмування графічних елементів у Java*. Львів: Видавництво "Професіонал", 2021.
35. Кулаков, Л. М.. *Аспекти безпеки у відеоіграх: підручник*. Київ: Видавництво "Комп'ютерна академія", 2020.
36. Григоренко, О. В.. *Основи мультимедійних технологій у геймдизайні*. Харків: Видавництво "Інженер", 2021.
37. Марченко, С. П.. *Ігрова анімація в Java: практичний посібник*. Київ: Видавництво "Технічна література", 2022.
38. Бондаренко, О. С.. *Тестування ігор на Java: методи та інструменти*. Дніпро: Видавництво "Графіка", 2020.
39. Шевченко, І. В.. *Розробка мобільних ігор на платформі Android з використанням Java*. Київ: Видавництво "Техніка", 2021.
40. Лисенко, А. П.. *Вирішення проблем ефективності у Java-іграх: практичний підхід*. Львів: Видавництво "Мистецтво програмування", 2022.