МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Харківський національний університет радіоелектроніки
Факультет Комп'ютерних наук
Кафедра Програмної інженерії

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

другий (магістерський)
(рівень вищої освіти)

Дослідження самонавчальних алгоритмів оптимізації кривої складності навчання сліпому друку

Виконав:
студент 2 курсу, групи ІПЗм-20-3
Станчик К.В.
(прізвище, ініціали)

Спеціальність	121 – Інженерія програмного забезпечення
Тип програми	освітньо-наукова
Керівник	доц. Назаров О.С.

Допускається до захисту

Зав.кафедри З.В.Дудар

2022 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

Кафедра Програмної інженерії

Рівень вищої освіти Другий (магістерський)

Спеціальність 121 – Інженерія програмного забезпечення

(код і повна назва)

Тип програми освітньо-наукова програма

Освітня програма Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

«__» _____ 202_ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студента Станчика Кирила В'ячеславовича

(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження самонавчальних алгоритмів оптимізації кривої складності навчання сліпому друку.

затверджена наказом університету від «__» _____ 2022 р. № ___

2. Термін здачі студентом закінченої роботи «__» _____ 2022 р.

3. Вихідні дані до проекту:

Технічне завдання, календарний план, методичні вказівки до кваліфікаційної роботи магістра

4. Зміст розрахунково-пояснювальної записки:

Вступ, аналітичний огляд, аналіз існуючих методів, постановка задачі, опис реалізації, формування вимог до дослідження, висновки, перелік посилань

5. Перелік графічного матеріалу:

9 рисунків, 4 таблиці.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз предметної галузі	31.01.2022	виконано
2	Постановка задачі	07.01.2022	виконано
3	Формування вимог	14.02.2022	виконано
4	Проектування алгоритмів у ПЗ	21.02.2022	виконано
5	Реалізація алгоритмів у ПЗ	07.03.2022	виконано
6	Планування дослідження	01.04.2022	виконано
7	Аналіз дослідження	15.05.2022	виконано
8	Підготовка пояснювальної записки	02.05.2022	виконано
9	Перевірка пояснювальної записки керівником, підготовка до перевірки на антиплагіат	15.05.2022	виконано
10	Оцінка роботи рецензентом, отримання відзиву від керівника атестаційної роботи, попередній захист роботи та проходження нормо контролю	20.05.2022	виконано
11	Захист атестаційної роботи	25.05.2222	виконано

Дата видачі завдання *17* *січня* 202*2* р.

Студент _____
(підпис)

Керівник роботи _____ доц. Назаров О.С.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ / ABSTRACT

Пояснювальна записка містить: 57 с., 39 рис., 4 табл., 13 джерел.

ВІДЕОГРА, ГЕЙМІФІКАЦІЯ, ГРА, ІГРОВИЙ ДИЗАЙН, ІГРОВИЙ БАЛАНС, МАТЧМЕЙКІНГ, СЛІПИЙ ДРУК, TRUE SKILL

Об’єктом дослідження є методи балансування складності навчаючої відеогри.

Метою роботи є дослідження методів проектування та задання ігрового балансу у відеоіграх та дослідження методів автоматичного коригування балансу відповідно до навиків гравця.

Результатом роботи є порівняльне дослідження методів задання ігрового балансу та дослідження методів аналізу успішності застосування різних підходів до автоматичного коригування балансу. Дослідження відбувається на базі задання балансу для навчаючої відеогри, що націлена на вивчення та тренування навику сліпого друку.

Explanatory note contains: 57 pages, 39 figures, 4 tables, 13 sources.

VIDEOGAME, GAMIFICATION, GAME, GAME DESIGN, GAME BALANCE, MATCHMAKING, TOUCH TYPING, TRUE SKILL

The research object is methods of balancing the complexity of educational video games.

The goal of the research is to study the methods of designing and methods of setting up the game balance in video games and the study of the methods of automatic balance adjustment according to the player's skills.

The results are a comparative analysis of methods for setting the game balance and research of methods for analyzing the success of different approaches to automatic balance adjustment. The research is based on setting a balance for an educational video game, which aims to teach and train the skill of touch typing.

Умови публікації пояснювальної записки

Я,

Станчик Кирило В’ячеславович
(прізвище, ім’я, по батькові)

студент групи ІПЗм-20-3, здобувач вищої освіти на другому (магістерському) рівні кафедра Програмної інженерії,
(повна назва кафедри)

заявляю: моя кваліфікаційна робота на тему Дослідження самонавчальних алгоритмів оптимізації кривої складності навчання сліпому друку,
(назва роботи)

що буде представлена до ЕК для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу ElArKhNURE. Всі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений з діючим положенням «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

ЗМІСТ

ВСТУП

Налаштування ігрового балансу є одним із ключових процесів проектування відеоігор. Баланс впливає на усі частини гри: від основного ігрового циклу до мета-ігрового циклу. У першому він створює гравцям виклик для покращення своїх реальних та віртуальних ігрових навиків, а у другому створює мотивацію загальній внутрішньо ігровій прогресії гравця.

Корегування балансу важливо для того, щоб у кожний момент зацікавлювати гравця не створюючи нездоланно важкий чи занадто нудний ігровий досвід, що у свій час впливає на утримання гравця у грі. Через свою важливість для досягнення бізнес цілей процес балансування зазвичай підлягає багатьом ітераціям розробки та постійному корегуванню.

Завдяки автоматичним системам корегування можна досягати балансу, що відповідає максимально широкій аудиторії гравців. При цьому скоротиться кількість ручної роботи по підготовці ігрового балансу та, потенційно, скоротиться кількість ітерації, що вимагають ручну роботу. Деякі частини ігрового процесу чи повноцінні жанри не можливо повністю збалансувати вручну і вони можуть існувати лише з використанням систем автоматичного корегування балансу.

Тим не менш їх присутність на ринку не найпоширеніша через сильну орієнтованість ігрових проектів на бізнес. Повністю самонавчальні алгоритми можна зустріти переважно лише на проектах, що вже давно на ринку і потребують додаткової оптимізації. Це викликано довгим циклом розробки таких рішень, що відтермінує досягнення необхідних метрик повернення інвестицій для нових проектів. Через цю ж причину їх не можна аналізувати у відриві від бізнес-підходів.

Лабораторія ігрової розробки на кафедрі програмної інженерії займається просвітою в контексті розробки відеоігор: від проектування концептів та вивчення методів балансування ігрових проектів, до розробки ігрових застосунків та вивчення підходів до аналізу проектів з точки зору бізнесу.

Об’єктом дослідження є навчальний ігровий застосунок, що обучає навику сліпого друку та дозволяє повторно тренувати та засвоювати цей навик. Ігровий застосунок та методи балансування реалізовані використовуючи вбудовані інструменти універсального ігрового рушія Unreal Engine 4.

Метою дослідження є вивчення методів проектування та задання ігрового балансу у відеоіграх, дослідження методів автоматичного коригування балансу відповідно до навиків гравця та дослідження підходів до аналізу успішності застосування різних підходів до задання та корегування балансу.

Тези до роботи опубліковані на дванадцятій міжнародній науково-технічної конференції «Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління».

1 АНАЛІТИЧНИЙ ОГЛЯД

Для початку проектування методів балансування треба розуміти предметну область для котрої виконується проектування. Також розуміння різних рішень допоможе заздалегідь почати планувати аналіз та потенційні майбутні ітерації орієнтуючись на актуальні проекти в сфері

Техніка сліпого друку є одним з варіантів покращення роботи з текстами за допомогою клавіатури. Суть техніки у використанні клавіатури без необхідності пошуку клавіш на ній. Бере свій початок у 1888 році і великий час використовувалася на друкарських машинках стенографістами. Залежно від тренованості, можна друкувати від 200 до 400 символів за хвилину, що значно знижує трудомісткість будь-якого завдання і скорочує час, що витрачається.

Більшість існуючих додатків для тренування цієї техніки використовують методику поступового навчання задаючи користувачу певні випадкові сполучення або існуючі слова у заданому порядку (рис.1.1).

Рисунок 1.1 – Приклад навчання техніці сліпого друку
у застосунку «Ratatype»

Фактично порядок слів, що надають ці додатки задає криву складності навчання, котра, в свою чергу, складними частинами змушує користувача підвищувати навик володіння технікою, а легкими частинами підтримує досягнення гравця і мотивує продовжувати.

Багато подібних додатків використовують елементи ігрофікації [1] і намагаються імітувати гру [2]. Зумовлено це тим, що гра – це процес вирішення проблеми, котрий виконується з ігровим настроєм [3]. Ігровий настрій допомагає утримувати користувача у додатку, а це в свою чергу необхідно, щоб користувач повністю завершив курс та опанував техніку – що і є ціллю самого додатку для котрої він створювався.

Якщо врахувати те, що додатки використовуються елементи ігрофікації (рис.1.2), то фактично вони є міні-іграми, а заданий порядок слів для тренування – є ігровим балансом.

Рисунок 1.2 – Відеогра-тренер друку «Keyboard Sports»

Ігровий баланс – це досягнення верхньої точки вузького коридору задоволення: якщо зробити гру занадто простою, вона буде нудною, а якщо зробите гру занадто складною – вона теж не буде приносити задоволення, а тільки біль. Завдання дизайнерів при створенні балансу – потрапити в цей дуже вузький діапазон. Математика балансу — це не завжди найбільша його частина. Це рутинна частина, яку доведеться робити.

Найбільша частина, яку доведеться робити – це тестування та ітерування балансу. При розробці ігор доведеться грати в гру нескінченно довго та робити багато ітерацій для корегування ігрового балансу.

Інколи баланс переходить на автоматизований рівень [4] – зазвичай це відбувається у онлайн іграх де потрібно збирати групи з декількох гравців разом. Часто цей процес відбувається на основі системи рейтингу, яка намагається зіставити гравців з приблизно однаковими здібностями.

Окремі гравці або команди шукають гру, і система порівнює їх з іншими подібними гравцями. Як тільки буде знайдено відповідну кількість гравців, матч проводиться, і гра може розпочатися.

По завершенню матчів на основі результату цих матчів відбувається корегування рейтингу гравців для майбутніх зборів матчів.

Основні переваги підбору матчів — це більш конкурентоспроможні матчі, можливість шукати як команда та грати з іншою командою, а також різноманітність ігрових конфігурацій. Компроміси включають тривалий час пошуку, щоб знайти гарний збіг, і неможливість вибрати точну карту або тип гри для гри.

Таким чином система автоматично підлаштовує ігровий баланс для того щоб він підходив навикам гравців.

2 АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ І АЛГОРИТМІВ

2.1 Огляд етапів роботи з балансом

Робота з ігровим балансом складається з декількох основних етапів:

– підготовка системи для задання балансу;

– підготовка початкової версії балансу;

– ітеративне корегування балансу.

Хоча етап підготовки початкового балансу і є етапом, що задає загальну форму кривої складності, котру бачить дизайнер – ця форма може бути помилковою і гарантовано у деталях буде не ідеальною. Цей факт робить останній етап – етап ітеративного корегування - найбільш важливим, водночас найбільш трудомістким та затяжним.

Для ітеративного корегування існує багато різних підходів: деякі відрізняються незначними елементами, а деякі діаметрально протилежні іншим.

2.2 Метод статичного балансу

Метод задання статичного балансу полягає у тому, що для кожної ігрової сутності баланс задається фіксованими значеннями і впродовж гри ці значення незмінні та однакові для всіх гравців.

При такому підході баланс може задаватися як безпосередньо у налаштуваннях ігрових сутностей у ігровому рушії, або, частіше, у вигляді таблиці даних, котру ігровий рушій зчитує на старті ігрової сесії.

Варіант задавання у вигляді таблиці даних дозволяє зберігати дані у зручному для читання та редагування форматі коли усі ігрові об’єкти одного типу є окремою вкладкою у електронній таблиці. Це ефективний спосіб для порівняння об’єктів один з одним та редагування за необхідності. Також ці таблиці часто містять додаткові поля необхідні для порівняння сутностей, що не

використовуються грою, але використовуються при математичних підрахунках за допомогою формул (рис. 2.1).

Цей підхід розповсюджений на невеликих і середніх ігрових проектах для підтримки загального балансу гри та у великих проектах для підтримки окремих частин, або для задавання початкового балансу об’єктів, котрий потім вже змінюється автоматизованими методами.

	A	B	C	D	E	F	G	H	I	J	K	L
1	Enemy	Level	Attack name	Attack visual	Attack type	Attack range	Attack speed	Damage	Preparation	Burst	Burst delay	Pure DPS
2	Sentinel	1	Magic-ball	Projectile	Standard	5	3	20	0,5	1	1,2	12
3		2	Magic-ball	Projectile	Standard	6,0	3,6	24,0	0,4	1	1,0	18
4		3	Magic-ball	Projectile	Standard	7,2	4,3	28,8	0,3	2	0,8	41
5	Shielded Sentinel	1	Shield kick	Melee	Melee	1	3	30	0,8	1	1,2	15
6		2	Shield kick	Melee	Melee	1,2	3,0	45,0	0,6	1	1,0	28
7		3	Shield kick	Melee	Melee	1,4	3,0	67,5	0,5	1	0,8	53
8	Archer	1	Arrow	Projectile	Special	10	6	30	1,5	1	2	9
9		2	Magic-arrow	Projectile	Special	15,0	7,2	45,0	1,4	1	1,6	15
10		3	Magic-arrow	Projectile	Special	22,5	8,6	67,5	1,2	1	1,3	27
11	Summoner	1	Summoners-ball	Projectile	Special	8	2	0	3	1	5	0
12		2	Summoners-ball	Projectile	Special	12,0	2,4	0,0	2,7	1	4,0	0
13		3	Summoners-ball	Projectile	Special	18,0	2,9	0,0	2,4	1	3,2	0
14	Tamer	1	Bite	Melee	Melee	1	3	25	0,3	1	1,5	14
15		2	Bite	Melee	Melee	1,2	3,0	37,5	0,2	1	1,2	26
16		3	Bite	Melee	Melee	1,4	3,0	56,3	0,2	1	1,0	49
17		2	Jump rush	Melee	Special	10	5	75	2	1	8	8
18		3	Jump rush	Melee	Special	15,0	6,0	112,5	1,8	2	6,4	23
19	Mini-boss	1	Magic-ball	Projectile	Standard	8	5	50	1	1	1,2	23
20		2	Magic-ball	Projectile	Standard	9,6	6,0	60,0	0,8	1	1,0	34
21		3	Magic-ball	Projectile	Standard	11,5	7,2	72,0	0,6	2	0,8	70
22		1	Fire-ball	Projectile	Standard	12	5	30	1,5	1	3	7
23		2	Fire-ball	Projectile	Standard	14,4	6,0	36,0	1,2	1	2,4	10
24		3	Fire-ball	Projectile	Standard	17,3	7,2	43,2	1,0	1	1,9	15
25		2	Teleportation	Cast	Special	8	0	0	3	1	5	0
26		3	Teleportation	Cast	Special	12,0	0,0	0,0	2,7	1	4,0	0

Рисунок 2.1 – Приклад статичної таблиці балансу

При подібному підході можуть існувати декілька рівнів кожного об’єкту, що підміняються по мірі проходження гравцем гри. При цьому кожен рівень або заданий у таблиці імпорту балансу, або заданий формулами з декількома початковими значеннями та одним або декількома змінними коефіцієнтами. Коефіцієнти так само задаються у цій таблиці і є статичними.

Ітеративне корегування балансу для відповідності рівню навиків гравця у статичному методі відбувається через ітеративні ігрові тести на фокус групах. Вибір респондентів фокус групи повинен залежати від того, що необхідно дізнатися. Швидше за все це люди людей, які належать до цільової аудиторії проекту. При цьому робляться записи під час спостережень за ігровим процесом та збирається зворотній зв’язок від фокус групи. На основі цього робляться висновки і

проводяться корегування у баланс: зміни порядку рівнів, або зміна балансу окремих рівнів чи їх частин.

Для все запущеного проекту, що знаходиться на етапі підтримки, можливе корегування статичного балансу на основі отриманих даних фактичної аудиторії, а не фокус груп. Це дає більш точні і чисті дані про відповідність заданого балансу і фактичних можливостей гравців.

Слід зауважити, що загальний підхід до ігрових тестів залишається актуальним для будь якого методу.

Основні переваги методу статичного балансу:

– легкий спосіб інтеграції;

– можливість бачити фінальні значення у зручному вигляді.

Основні недоліки методу статичного балансу:

– складність ігрового процесу може не відповідати навикам гравця;

– ітеративне корегування вимагає велику кількість ітерацій;

– корегування відбується на основі суб'єктивного досвіду;

– під час зміни необхідно змінювати велику кількість значень.

2.3 Метод заснований на тригерних подіях

Тригерний метод є еволюцією методу статичного балансу і базується на ньому у заданні початкових значень.

Він також включає в себе роботу зі статичними таблицями ігрового балансу, але корегування балансу відбувається для кожного гравця окремо в залежності від його навиків.

Так складність гри підлаштовується під гравця в залежності від його попередніх дій у ігровому процесі. Гравцям з кращими навиками, що пройшли декілька складних рівнів на найкращий результат гра швидше підвищує складність роблячи ворогів більш складними. Або гравцю, що не може пройти певний рівень гра встановить на одного ворога менше таким чином понижуючи складність.

Зміна балансу може робитися як зміною певних коефіцієнтів, якщо баланс побудовано на формулах, або простим вибором іншої строки балансу.

При цьому треба у баланс додавати додаткові коефіцієнти та проробляти архітектуру тригерів кожної події на основі якої необхідно змінювати баланс.

Тригери можуть як задаватися для специфічної події так і бути уніфікованою системою. Другий варіант є більш пріоритетним, бо уніфіковану архітектуру з тригерами, що повторно використовуються у різних частинах гри легше документувати та підтримувати.

Тригерний метод так само вимагає велику кількість ігрових тестів, але вони полегшуються бо здебільшого необхідно збирати окремі групи з однаковим великим, середнім чи низьким рівнем навиків і на них тестувати в яких межах мають відбуватися автоматичні зміни.

Також ігрові тести при тригерному методі допомагають визначити місця для специфічних тригерів, бо виявляють проблемні місця у ігровому процесі де гравцям важко або навпаки нудно.

Основні переваги методу тригерної зміни балансу:

– ігровий процес підстроюється під рівень навиків гравця;

– велика кількість під-методів налаштування, котрі можна обрати в залежності від необхідної точності та розміру проекту;

Основні недоліки тригерної зміни балансу:

– в залежності від реалізації корегування балансу може виявитися досить грубим;

– ітеративне корегування вимагає велику кількість ітерацій налаштування подій і коефіцієнтів;

– додатково до балансу необхідно проектувати архітектуру тригерів.

2.4 Система рейтингової оцінки навиків

Прикладом такого підходу є система TrueSkill – це система рейтингу на основі навичок гравців розроблена в Microsoft Research для Xbox Live [5]. Метою систем рейтингу у іграх є виявлення і відстеження навичок гравців у грі або специфічному ігровому режимі, щоб мати можливість поєднати їх у збалансовані змагальні матчі.

Класична система рейтингу TrueSkill використовує лише підсумкову таблицю всіх команд у матчі, щоб оновити оцінки навичок (ранги) усіх гравців у матчі. Система рейтингу TrueSkill 2 також використовує індивідуальні бали гравців, щоб зважити внесок кожного гравця в кожну команду. В результаті TrueSkill 2 набагато швидше з’ясовує навички нового гравця.

У рейтинговій системі TrueSkill навички гравця характеризується двома числами:

– μ – середня майстерність гравця;

– σ – ступінь невизначеності в майстерності гравця.

Система рейтингу підтримує віру в майстерність кожного гравця, використовуючи ці два числа. Якщо невизначеність все ще висока, система рейтингу ще не знає точних навичок гравця. Навпаки, якщо невизначеність невелика, система рейтингу твердо переконана, що майстерність гравця близька до середньої.

Підтримка невизначеності дозволяє системі вносити великі зміни в оцінки навичок на ранньому етапі, але невеликі зміни після серії послідовних ігор (рис.2.2). В результаті система рейтингу може визначити навички окремих гравців з дуже невеликої кількості ігор.

Завдяки такому підходу метод системи рейтингової оцінки навиків досить точно підлаштовує складність гри до фактичних навичок гравців, що в результаті створює гравцям позитивний ігровий досвід та утримує їх у грі протягом довгого часу.

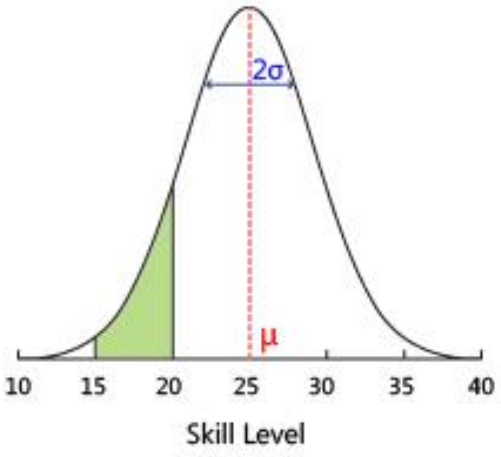

Рисунок 2.2 – Візуалізація невизначеності методу TrueSkill

У таблиці 2.1 наведено приклад мінімальної кількість ігор на одного гравця, яка потрібна системі для визначення рівня навичок у різних змагальних режимах.

Таблиця 2.1 – Кількість матчів на гравця для визначення навику за методом рейтингової оцінки

Ігровий режим	Кількість матчів на гравця
16 Гравців Кожен-за-себе	3
8 Гравців Кожен-за-себе	3
4 Гравця Кожен-за-себе	5
2 Гравця Кожен-за-себе	12
4 Команди / 2 Гравця на команду	10
4 Команди / 4 Гравця на команду	20
2 Команди / 4 Гравця на команду	46

Фактична кількість ігор, необхідних на одного гравця, може бути втричі вищою залежно від кількох факторів, таких як зміна результативності за гру, наявність суперників, які збігаються, шанси на нічию тощо.

Система рейтингу TrueSkill порівняє індивідуальні навички гравця (числа μ і σ) з навичками всіх ігрових хостів для певного ігрового режиму на Xbox Live і

автоматично порівняє гравця, для котрого розшукується матч, з гравцями з подібними навичками до його.

Основні переваги системи рейтингової оцінки навиків:

– ігровий процес точно підстроюється під рівень навиків гравця;

– легка інтеграція у гру.

Основні недоліки тригерної зміни балансу:

– у чистому вигляді підходить лише для корегування ігрового балансу багатокористувацьких змагальних ігор.

2.5 Метод проектування Match-3 рівнів

Основні особливості Match-3 ігор, що необхідні для розуміння підходу:

– гра складається з багатьох рівнів, які поступово відкриваються. Розблоковані рівні можна пройти кілька разів.

– у гравців є різні ресурси, баланс яких можна змінити піл час проходження ігрового рівня.

– у грі є різні бонусні предмети, які полегшують гравцеві проходження гри.

При дизайні не контролюється розміщення кожного елементу дошки, а створюється форма дошки, розміщення блокувальних елементів, кількість і тип фігур, які можуть впасти [6].

Після чого робиться встановлення коефіцієнтів випадіння та зміщення для тих частин, які випадають. Загалом цей процес випадковий, але заснований на упередженості, яку задає дизайнер.

Таким чином дизайнер отримує базовий баланс.

Після чого рівень проходить процес ручного тестування дизайнером та іншими членами команди, щоб перевірити, чи він приблизно працює і чи є це веселим досвідом.

Після завершення етапу ручного попереднього тестування, використовуються AI-боти з низьким інтелектом – ці проходження AI можна моделювати тисячі разів, що дає статистично значимі значення спроб на успіх.

Як тільки рівні будуть доступні реальним гравцям необхідно ввести зміни на основі реальних даних, щоб отримати рівень на кривій складності там, де необхідно.

Слід зазначити, що цей підхід зазвичай не передбачає перестановку порядку рівнів для значимих змін кривої складності, бо більшість рівнів додаються після запуску гри у лінійку, що слідує списку рівнів на старті проекту.

Основні переваги методу проектування Match-3 рівнів:

– базові етапі прості для інтеграції;

– використовує автоматизацію AI-ботами на етапі ітеративного корегування балансу;

– може бути покращений додатковою роботою над алгоритмом випадіння таким чином додавши автоматичне корегування балансу.

Основні недоліки тригерної зміни балансу:

– обмежений особливостями жанру та потребує адаптації для використання у інших жанрах;

– опирається на генератор випадковостей, що може давати розбіжність у складності на різних повтореннях.

3 ПОСТАНОВКА ЗАДАЧІ

3.1 Вихідні вимоги

Спроектувати концепт системи задання і авто-налаштовування ігрового балансу у відеогрі, що націлена на навчання та тренування навику сліпого друку.

Опис особливостей відеогри, котрий треба врахувати при розробці системи:

– дозволяє користувачам пройти повний курс сліпого друку: вивчити та практикувати прив’язку пальців до всіх клавіш клавіатури;

– курс повинен бути розділений на тематичні секції;

– уроки в рамках курсу мають включати механіку контролю засвоєння проміжного рівню навику;

– відеогра використовує елементи ігрофікації для мотивації здобуття майстерності навику;

– відеогра має ставити довгострокові цілі перед користувачем.

Загальний ігровий цикл відеогри – головний елемент ігрової механіки, який визначає фундаментальний досвід гравця [7] – наведено на рисунку 3.1.

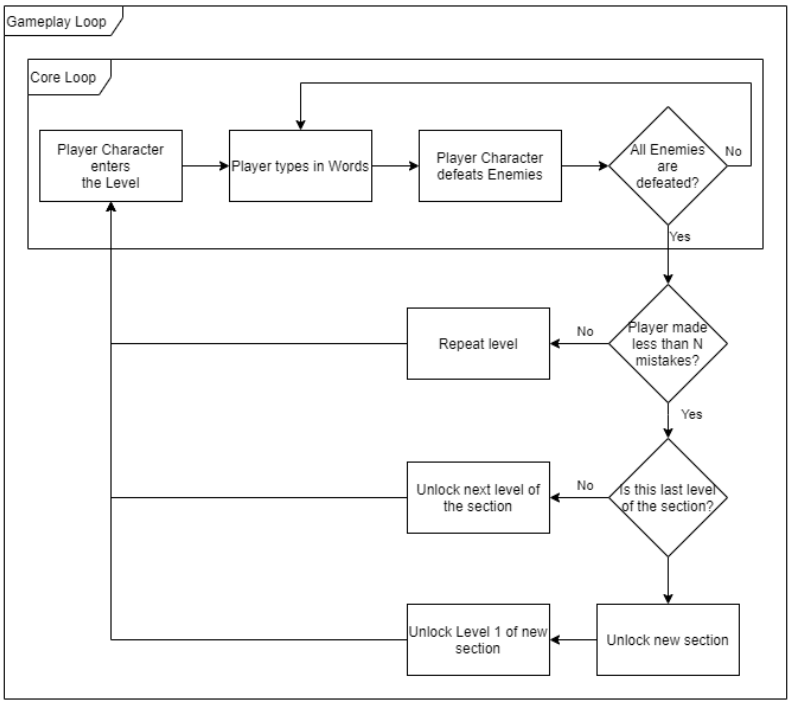

Рисунок 3.1 – Загальний ігровий цикл

Відеогра дозволяє проходити рівні, переглядати досягнення та статистику під час проходження та містить ігрові механіки, що розважають користувачів, але не перешкоджають процесу навчання.

Дії, що може виконувати гравець у грі наведено на рисунку 3.2.

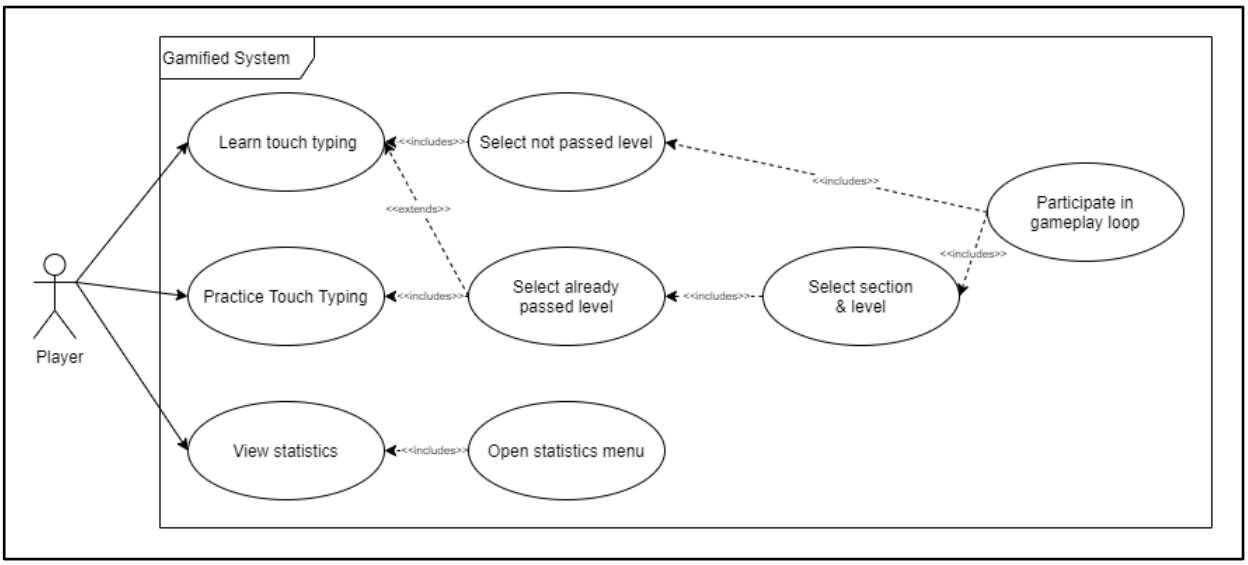

Рисунок 3.2 – Діаграма використання навчальної відеогри

Баланс рівнів має автоматично підстроюватися під навики гравця для пришвидшення освоєння техніки сліпого друку опираючись на минулі рівні, що пройшов гравець. При цьому порядок навчання має залишатися послідовним і вчасно переходити на вивчення наступної теми.

Основними критеріями успішності алгоритму є простота реалізації підтримки та швидкість освоєння техніки сліпого друку.

3.2 Додаткова інформація про архітектуру додатку

Прототип додатку вже містить базову реалізацію ігрового циклу.

Весь процес навчання розділений на навчальні секції, що складаються з окремих рівнів, котрі в свою чергу складаються з окремих хвиль ворогів.

Секції об’єднані тематично певним навчанням окремого під-навику сліпого друку, наприклад друку верхнього ряду вказівним та середнім пальцями.

Рівні ж містять окремі конкретні уроки по цьому напряму: від окремих рівнів для механічного запам’ятовування розміщення клавіш по темі, до рівнів випробувань, що можуть включати перевірку усіх попередніх вивчених під-навиків. Наприклад, випробування сліпого друку верхнього ряду вказівними та середніми пальцями може містити необхідність виконувати введення середнього ряду вказівними середніми, безіменними пальцями та мізинцями, якщо вони були вивчені до цього.

Хвилі ж реалізують появу ворогів і більш ігрову поведінку додатку (рис.3.3)

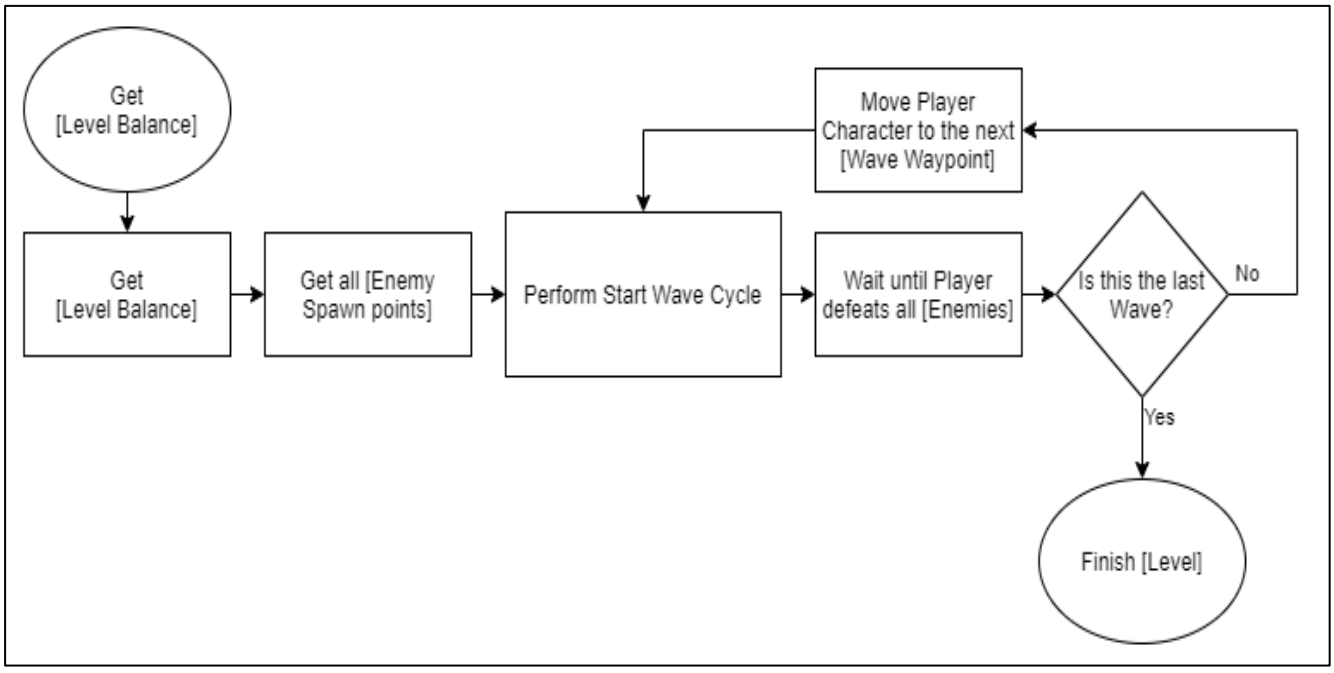

Рисунок 3.3 – Загальний алгоритм ігрового циклу рівня,
що складається з декількох хвиль

Кількість слів для рівня та точок появи ворогів – нерівна. Щоб уникнути проблем з повторенням слів на ворогах та появою декількох ворогів на одній точці, коли один з вищезгаданих списків вичерпується процес підготовки хвилі зупиняється і вона запускається (рис.3.4).

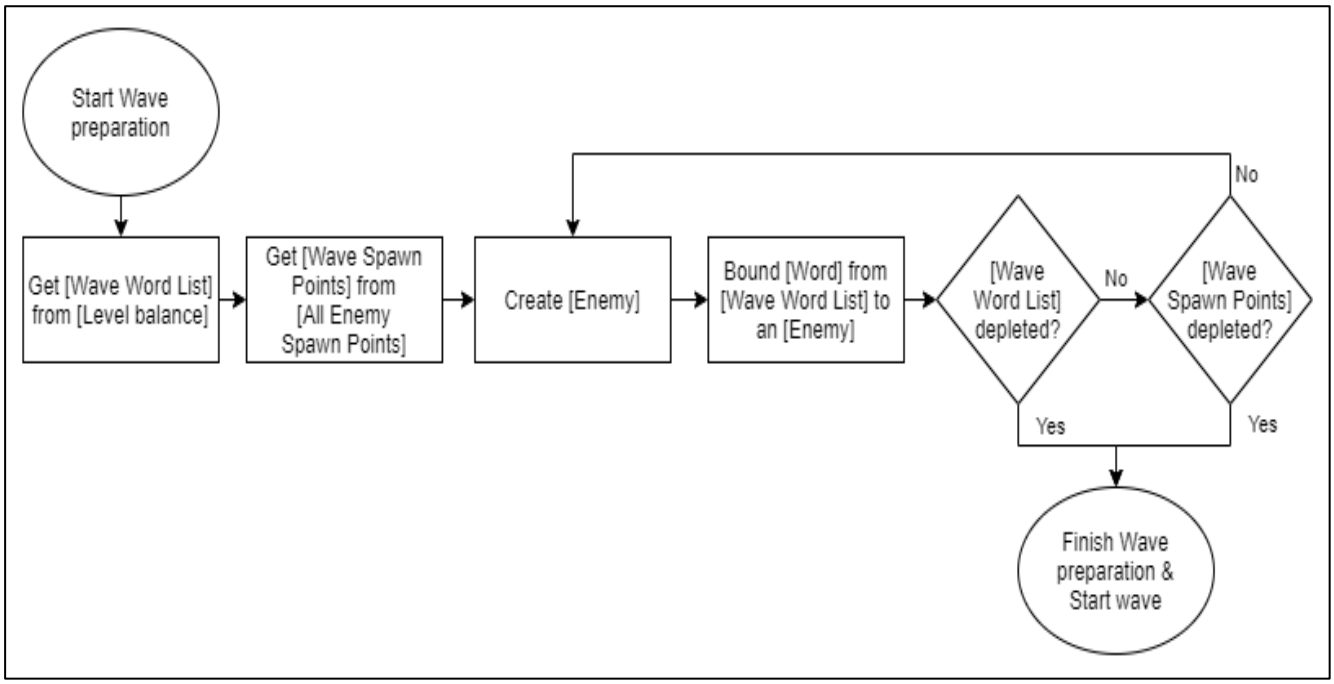

Рисунок 3.4 – Загальний алгоритм запуску хвилі

В основі задання балансу лежить підхід статичного балансу з елементами випадковості – зі списки слів обирається певне окреме слово.

Для цього використовується вбудований в Unreal Engine 4 клас «DataTable» [8]. Він дозволяє зберігати дані у форматі схожому до електронних таблиць.

Для зберігання балансу створена структура, котра містить індекс рівня, індекс хвилі та список слів цієї хвилі (рис 3.5).

Data Table | Data Table Details

Search

	Row Name	LevelIndex	WaveIndex	WordList
1	NewRow	0	0	("jkkjk","kjkkj","jkjjk","kkjkj","jjkjk","jkjkk","kjkjj","jkjkj")
2	NewRow_0	0	1	("dffdf","fdffd","dfddf","ffdfd","ddfdf")
3	NewRow_1	0	2	("dkfdj","jdfkf","kjdkf","fkjdk","fkdkj","fkfdj","kjdkf","kfkdj","fjkjd")

Рисунок 3.5 – Приклад балансу рівня

Таким чином задаючи баланс через будь який додаток електронних таблиць можна точно керувати досвідом гравця на кожному рівні.

4 РЕАЛІЗАЦІЯ

4.1 Визначення оптимального варіанту реалізації

На основі аналізу існуючих методів проведемо попередню оцінку методу на котрий опиратися при розробці.

Виділимо 4 альтернативних методи:

– метод статичного балансу – для кожного рівня і хвилі ворогів заздалегідь заданий список слів; для кожної хвилі список відрізняється і визначений на основі плану навчання сліпому друку: яким саме дія, в якому порядку треба навчитися; під час розміщення ворога йому вибирається випадкове слово (рівномірна вірогідність) з заданого списку, котре не може повторитися для іншого ворога;

– тригерний метод – заздалегідь заданий список слів більший для кожної хвилі, між хвилями міститься більше повторень слів, задані тригери на здійснення помилки чи зменшення швидкості друку, котрі на певний відсоток збільшують шанс випадіння слів, що схожі зі словом на котрому спрацював тригер;

– адаптована система рейтингової оцінки – слова у наступних хвилях та рівнях сприймаються як гравці опоненти у рейтингу TrueSkill. Результат рейтингу інвертуються і гравцю підбираються слова з заданого списку проти котрих він з найбільшою вірогідністю програє;

– адаптований алгоритм Match-3 – для підбору слів на наступних хвилях використовується адаптована версія алгоритму створення балансу рівнів, що використовується у Match-3 іграх.

Виділимо 5 основних критерій вибору та їх шкали оцінювання:

а) Простота реалізації – вимоги до технічних знань та час на підготовку реалізації – чим проще тим краще:

1) Дуже проста – 5 балів;

2) Проста – 4;

3) Середня – 3;

4) Складна – 2;

5) Дуже складна – 1 бал;

б) Кількість параметрів – параметри алгоритму, котрі необхідно налаштувати заздалегідь – чим менше тим краще:

1) 50 і менше – 5 балів;

2) 100 – 4;

3) 150 – 3;

4) 200 – 2;

5) 250 і більше – 1 бал;

в) Складність тестування – вимоги до технічних знань та час на проведення тестування – чим простіше тим краще:

1) Дуже проста – 5 балів;

2) Проста – 4;

3) Середня – 3;

4) Складна – 2;

5) Дуже складна – 1 бал;

г) Імовірність виникнення помилкових рішень – очікування того, що гра обиратиме нерелевантні на окремому етапі навчання слова – чим нижче тим краще:

1) Дуже низька – 5 балів;

2) Низька – 4;

3) Середня – 3;

4) Висока – 2;

5) Дуже висока – 1 бал;

д) Ефективність навчання системою – кількість повторень слів, котрі гарантують, що гравець досяг певний порогу навиків – чим вище тим краще:

1) Дуже низька – 1 бал;

2) Низька – 2;

3) Середня – 3;

4) Висока – 4;

5) Дуже висока – 5 балів.

Таблиця 4.1 – Векторний опис альтернатив по критеріям

Альтернативи	Критерії				
	Простота реалізації	Кількість параметрів	Складність тестування	Імовірність виникнення помилкових рішень	Ефективність навчання системою
Статичний метод	Дуже проста	50	Складна	Дуже висока	Низька
Тригерний метод	Проста	150	Складна	Висока	Середня
Адаптована рейтингова оцінка	Складна	100	Середня	Низька	Висока
Адаптований Match3	Складна	150	Проста	Середня	Висока

Таблиця 4.2 – Числові значення критеріїв та вагові коефіцієнти

Альтернативи	Критерії				
	Простота реалізації	Кількість параметрів	Складність тестування	Імовірність виникнення помилкових рішень	Ефективність навчання системою
Статичний метод	5	5	2	1	2
Тригерний метод	4	3	2	2	3
Адаптована рейтингова оцінка	2	4	3	4	4
Адаптований Match3	2	3	4	3	4
Вагові коефіцієнти	0,3	0,1	0,15	0,15	0,3

Після проведення порівняння між параметрами домінуючих альтернатив не знайдено.

Кращу альтернативу обрано за допомогою лінійної адитивної згортки з ваговими коефіцієнтами (1) (2) (3) (4).

$$w(\text{Статичний метод}) = 0{,}3 * 5 + 0{,}1 * 5 + 0{,}15 * 2 + 0{,}15 * 1 + 0{,}3 * 2 = 3{,}05, \quad (1)$$

$$w(\text{Тригерний метод}) = 0{,}3 * 4 + 0{,}1 * 3 + 0{,}15 * 2 + 0{,}15 * 2 + 0{,}3 * 3 = 3, \quad (2)$$

$$w(\text{Адаптований рейтинг}) = 0{,}3 * 2 + 0{,}1 * 4 + 0{,}15 * 3 + 0{,}15 * 4 + 0{,}3 * 4 = 3{,}25, \quad (3)$$

$$w(\text{Адаптований Match3}) = 0{,}3 * 2 + 0{,}1 * 3 + 0{,}15 * 4 + 0{,}15 * 3 + 0{,}3 * 4 = 3{,}15 \quad (4)$$

Найкращим алгоритмом для використання при реалізації є адаптований алгоритм рейтингової оцінки.

4.2 Опис дизайну реалізації

Для задання базового ігрового балансу за основу береться статичний метод: для кожного рівня і хвилі ворогів заздалегідь заданий список слів; для кожної хвилі список відрізняється і визначений на основі плану навчання сліпому друку.

Кількість ворогів (а звідси і кількість слів на хвилі), що може буди розміщена одночасно, обмежена кількістю точок розміщення ворогів для хвилі.

На основі вивчення ринку виведена необхідність розміщати від 10 ворогів (слів) на секцію для стабільного засвоєння (частину рівня, що акцентує на одному під-навику теми рівня). В свою чергу секція може складатися з декількох хвиль. Кількість слів на секцію має перевищувати суму кількості точок появи ворогів на всіх хвилях секції (5).

$$WordsPerSection = \sum_{i=0}^{WaveAmount} TotalSpawns_i * 1{,}5 \quad (5)$$

де $WordsPerSection$ – кількість слів на секцію;

$WaveAmount$ – кількість хвиль у секції;

$TotalSpawns_i$ – кількість точок появи ворогів у хвилі i.

Вибір слова при присвоєнні ворогу відбувається у момент його розміщення на точці появи та відбувається на основі адаптованого рейтингу TrueSkill. При цьому всі наступні рівні містять такі слова, що стимулюють повторення навиків, що вивчені на попередніх рівнях.

За допомогою присвоєння рейтингу відбувається динамічне ітеративне корегування балансу кривої складності.

Попередні ігрові тести та вивчення ринку додатків навчання техніці сліпого друку визначили, що проблеми у користувачів виникають, коли специфічна літера, з котрою виникають проблеми друку, слідує за певною попередньою.

Для присвоєння рейтингу слова діляться на комбінації з двох букв, котрим безпосередньо присвоюється рейтинг. Кількість комбінацій на слово відповідає формулі (6).

$$CombinationsPerWord = TotalLettersInWord - 1 \quad (6)$$

Кожна комбінації автоматично присвоюється рейтинг μ_c – навик гравця при зустрічі з цією комбінацією, за замовчанням $\mu_c = 25$ для всіх комбінацій; та рейтинг невпевненості навику гравця $\sigma_c = 8{,}333$.

Після зустрічі з комбінацією, якщо гравець безпомилково проходить її рейтинги змінюється за рядом формул (7) (8) (9).

$$\mu_c = \mu_c + \frac{\sigma_c^2}{c}, \quad (7)$$

$$\sigma_c^2 = \sigma_c^2 * \left(1 - \frac{\sigma_c^2}{c^2}\right), \quad (8)$$

$$c^2 = 2\beta + \sigma_c^2, \quad (9)$$

де μ_c – навик гравця проходження комбінації літер;

σ_c – рейтинг невпевненості навику гравця;

β – відхилення від середньої швидкості друку гравця.

Для підбору слів при розміщенні ворогів для всього списку слів відбується підрахунок $\mu_{average}$ – середнього значення μ_c для всіх комбінацій, а далі відбувається аналіз потенційної успішності проходження:

$$SuccessRate = e^{-\frac{(\mu_{average}-\mu_c)^2}{2c^2}} * \sqrt{d}, \quad (10)$$

$$d = \frac{2\beta^2}{c^2} \quad (11)$$

де $SuccessRate$ – очікувана успішність проходження комбінації (від 0 до 1);

$\mu_{average}$ - середнього значення μ_c для всіх комбінацій у секції;

μ_c - навик гравця при зустрічі з цією комбінацією;

β – середня швидкість друку гравця.

З отриманого списку $SuccessRate$ при розміщенні вибираються, що містять найменше середнє значення $SuccessRate \to 0$. Таким чином алгоритм автоматично обирає слова з якими у гравця виникатиме найбільше проблем при друці.

5 ДОСЛІДЖЕННЯ

5.1 Планування дослідження

Аналіз методів задання та автоматичного корегування балансу можливий за двома головними напрямами:

– складність розробки методів;

– успішність навчання гравців навику сліпого друку.

Перший напрям дозволяє провести поверхневе визначення які методи потенційно дадуть краще відношення успішності навчання до вкладених зусиль. Цей аналіз необхідний для того, щоб визначити пріоритети розробки – який метод розробляти в якості головного методу задання та корегування балансу. Інші ж методи формують пріоритезований перелік методів на які переходити, якщо в ході розробки стане очевидно, що поверхневий аналіз містив недоліки. Дослідження в цьому напрямі було пророблено ще на етапі проектування реалізації (підрозділ 4.1).

Другий напрям – визначення успішності навчання – направлений на детальне вивчення успішності реалізованої системи авто-корегування балансу. Це дозволяє визначати точно визначати які дизайн рішення успішні, а які ні та планувати внесення коректив у коригуючу модель.

Найпопулярнішим базовим підходом для точкового аналізу в сфері відеоігор є проведення ігрових тестів з реальними гравцями [9]. На жаль аналіз ігрових тестів містить велику кількість вроджених проблем такі як людські помилки через упереджену суб’єктивну оцінку та трудність аналізу великої кількості тестів через майже повністю ручну організацію оцінки ігрових тестів [10]. Деякі популярні рішення другої проблеми – залучення великої кількості експертів до аналізу великої кількості тестів – в свою чергу циклічно повертають нас до першої проблеми, бо у різних людей різне суб’єктивне враження. Тим не менш, цей підхід на малій кількості респондентів, що грають у гру дозволяє швидко проводити загальну оцінку окремих рішень під час розробки перших версій гри.

Для успішної точкової оцінки моделі авто-корегування балансу рекомендується використовувати метод AB-тестування. Цей метод заснований на

поділенні великої кількості гравців на декілька груп та є широко розповсюдженим у сфері розробки умовно-безкоштовних ігрових проектів [11].

Загальна схема використання підходу АВ-тестування:

– визначення цілей тесту;

– планування моделі даних на основі котрої буде проводитись аналіз;

– дизайн та розробка в проекті точок заміру даних, котрі формують модель;

– розділення гравців на дві або більше групи (звідси назва АВ-тест) одна з котрих є контрольною групою, котра демонструє звичайну поведінку користувачів а інші тестовими;

– збір даних протягом виділеного проміжку часу;

– аналіз даних та формування висновків щодо тестових груп.

Додатково слід зауважити, що при проведенні АВ-тестів гравці не знають до якої групи вони належать, бо це може додати суб’єктивний фактор в їх поведінку та зруйнувати об’єктивність результатів аналізу. Також не рекомендується змінювати модель балансування під час тесту і якщо це зроблено розділяти в часових рамках отримані дані.

Цей метод є рекомендований для аналізу успішності вивчення навику сліпого друку за допомогою ігрофікованого програмного забезпечення, бо навик має виражені числові параметри за якими можна його оцінювати – відсутність помилок та швидкість друку – а АВ-тестування використовує виключно об’єктивні числові дані.

Також вагомим аргументом є популярність цього методу серед умовно безкоштовних ігрових проектів. Бо умовно безкоштовні ігрові проекти є здебільшого одразу бізнес-орієнтовані про що свідчить доля виручки цих проектів на загальному ринку відеоігор [12]. Тому використання їх досвіду може потенційно покрити бізнес ризики при розробці гейміфікованих проектів.

5.2 Постановка вимог до експерименту

Експеримент проводиться з використанням методики АВ-тестування.

Ціль проведення АВ-тесту – визначення зміни ефективності навчання користувачів методу сліпого друку за використання динамічної зміни кривої складності, що адаптується до поточних навиків гравці.

Гіпотеза, котру націлений підтвердити чи спростувати тест в ствердженні того, що динамічна система швидше навчає навику ніж система з незмінним однаковим для всіх порядком навчання.

Для порівняння розбиваємо користувачів на три групи:

– група А – контрольна – навчаються зі статичним заздалегідь заданим балансом з випадковим;

– група В – тестова – навчаються з тригерним методом;

– група С – тестова – навчаються з використанням адаптованої системи рейтингової оцінки навиків.

Для розширеного покриття експерименту у групі А рекомендується створювати баланс орієнтуючись на вже існуючі додатки. Таким чином на основі групи А можна робити приближені висновки щодо ефективності роботи додатків на ринку.

Також для спрощення порівняння вводимо обмеження, що під час роботи алгоритмів корегування кривої складності у групах В та С система обирає з того ж балансу, що й в групі А.

Кількість слів, котрі гравець фактично зустрічає на кожному етапі має бути в 3 рази менша ніж кількість заданих слів у балансі для цього етапу. Це умовне обмеження щоб зробити попередню оцінку алгоритмів.

Ефективність буде оцінюватися за допомогою замірів наступних показників:

– кількість зроблених помилок на кожному етапі;

– середня швидкість друку на кожному етапі;

– кількості помилок до відношення середньої швидкості друку на кожному етапі навчання;

– відхилення швидкості друку на певному тексті від середньої швидкості друку користувача;

– кількість спроб на кожний етап;

– довжина окремої сесії навчання;

– кількість сесій навчання щодня;

– щоденне повернення до додатку (Retention).

При аналізі, якщо різниця між групами становить менше 5% вважаємо різницю статистично незначимою [13].

АВ-тест проводиться протягом місяця, або до моменту коли 1000 гравців у кожній групі – всього 3000 гравців – завершить запропонований для проходження курс навчання чи перестане його проходити. Вважаємо, що гравець перестав проходити курс, якщо він не повертається до додатку протягом 5 днів.

Для проведення тесту обов’язковою умовою є публікація проекту на площадку розповсюдження Steam або Itch де гравці отримають доступ до завантаження проекту.

Збір даних анонімний – не зберігаються та не передаються жодні дані, що дозволяють встановити особу окремого гравця – тому додаткової узгодження з юридичної сторони не потрібно.

5.3 Опис реалізації експерименту

Для реалізації експерименту орієнтуємось на поточну архітектуру проекту (розділ 3.2) та вимогу до АВ-тесту (розділ 5.2).

Для подальшого аналізу результатів експерименту нам потрібно отримувати окремі дані про різні дії гравця. Для цього будуємо систему подій котрі будуть фіксувати окремі дії гравця для подальшої агрегації та аналізу (таблиця 5.1).

Таблиця 5.1 – Система подій для логування дій гравця

#	Момент фіксації	Параметри	Цільовий KPI
1	Зроблена помилка	– Слово – Номер літери слова – Номері рівня та хвилі – Номер слова у хвилі	– Кількість помилок
2	Завершення хвилі	– Середня швидкість – Номері рівня та хвилі	– Середня швидкість друку
3	Зміна швидкості друку відносно середньої на 30%	– Середня швидкість до зміни – Швидкість друку після зміни – Слово – Номер літери слова – Номері рівня та хвилі – Номер слова у хвилі	– Середня швидкість друку – Відхилення швидкості друку на певному тексті
4	Виграш рівня	– Номері рівня та хвилі – Дата й час – Статус чи пройдено	– Кількість спроб на кожний етап
5	Програш рівня	– Номері рівня та хвилі – Дата й час – Статус чи пройдено	– Кількість спроб на кожний етап
5	Початок сесії	– Дата й час	– Довжина окремої сесії – Кількість сесій на день – Щоденне повернення
6	Кінець сесії	– Дата й час	– Довжина окремої сесії – Кількість сесій на день – Щоденне повернення

Через те, що гравець може обирати порядок знищення ворогів у хвилі для параметру номеру слова у хвилі визначаємо як значення на один більше кількості знищених ворогів хвилі.

Запис робиться у виділений файл логів, що відправляється по мережі після кожного пройденого рівня. Якщо немає можливості відправити дані після проходження рівня через відсутність інтернету або іншу причину – файл логів поповнюється і відправляється при наступній можливості. Приймається ризик того, що окремі дані окремих гравців будуть втрачені – вважаємо цей ризик низьким і невпливаючим на подальший аналіз.

Запис даних виконується за допомогою вбудованого макросу Log Info. В якості значення передається прописані у таблиці 4 параметри конвертовані до формату JSON.

5.4 Подальше використання результатів дослідження

Спроектований АВ-тест можна не змінюючи перезапускати заміняючи при цьому параметри ігрового балансу: корегування підбору слів під час помилок, допустиме відхилення середньої швидкості друку та інші щоб додатково налаштовувати використовувану модель динамічної зміни параметрів ігрового балансу.

Результати цього аналізу цього АВ-тесту можуть використовуватися в подальшому для інших ігрофікованих додатків навчання.

Так само можуть використовуватися окремі елементи цього АВ-тесту для інших ігрових застосувань і звичайних навчальних додатків для аргументвації використання тіх чи інших рішень. Наприклад результати підвищеного щоденного повернення до гри у тестових групах можуть бути використані для звичайних ігор щоб аргументувати використання систем самонавчання.

ВИСНОВКИ

Результатом виконання роботи є дизайн системи задання і авто-налаштовування кривої складності балансу у відеогрі, що націлена на навчання та тренування навику сліпого друку та сформовані вимоги для дослідження ефективності розробленої системи.

Протягом роботи проведення аналітичний огляд сфери ігрофікованих проектів з навчання навикам сліпому друку та огляд різноманітних підходів до задання та автоматичного корегування ігрового балансу.

Система задання балансу заснована на методі статичного задання балансу. Автоматичне корегування балансу засноване на адаптованому алгоритмі рейтингової оцінки навиків, що засновується на рейтинговій системі TrueSkill Matchmaking. Остання вивчена на основі відкритих джерел.

Вибір підходів зроблений на основі порівняльного аналізу 4 альтернативних методів роботи з ігровим балансом: статичного методу, тригерного, адаптованого методу рейтингової оцінки навиків та адаптованого методу Match3 ігор. Для перелічених альтернатив визначені основні критерій вибору та задані шкали їх оцінювання. Після чого проведена перевірка Парето, котра не виявила домінуючих альтернатив. В кінці найкраща альтернатива виявлена за допомогою лінійної адитивної згортки з ваговими коефіцієнтами.

Система автоматичного корегування ігрового балансу дозволяє оцінювати навики гравця на основі успішності проходження ним кожної пари літер слів. Комбінаціям з пар літер присвоєно значення рейтингу котре змінюється за адаптованими формулами після безпомилкового та помилкового проходження їх гравцем.

Для дослідження ефективності цієї системи пропонується використання підходу АВ-тестування з розділенням користувачів на окремі групи, що будуть проходити навчання з різними системами корегування балансу та без них. Подібний підхід є широко популярним серед індустрії умовно-безкоштовних

ігрових додатків, котра є переважно бізнес-орієнтовано, що дуже важливо для будь-якого ігрового чи ігрофікованого додатку.

Дослідження націлене на підтвердження чи спростування гіпотези про те, що системи автоматичного корегування балансу підвищують швидкість освоєння навику, що вивчається.

Для підтвердження цього сформований список з ключових показників ефективності, котрі націлені як безпосередньо на ефективність навчання так і на бажання повертатися до навчання.

Для збору показників ефективності спроектований список подій котрі треба фіксувати під час ігрового процесу.

Під час дизайну системи корегування балансу, формування списку ключових показників та списку подій бралися в увагу поточна архітектура проекту та вимоги до АВ-тесту. Ціллю цього є скорочення витрат на розробку системи та підготовки дослідження.

Сфера застосування результатів дослідження – проектування та розробка ігрофікованих додатків з навчання та інших ігрових застосувань.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Araya Roberto. Does Gamification in Education Work – 2019. – 47 c. doi: 10.18235/0001777

2. Sus, B., Tmienova, N., Revenchuk, I., Bauzha, O., Stirenko. Gamification approach to the creation of virtual laboratory works and educational courses // S.CEUR Workshop Proceedings, 2020, 2711, c. 68-78.

3. Jesse Schell. The Art of Game Design – CRC Press, 2014. C.33–48. doi: 10.1201/b17723

4. Oleksandr Topchii, Oleksandr Samantsov, Oksana Mazurova, Mariia Shirokopetleva. A Study of Optimization Models for Creation of Artificial Intelligence for The Computer Game in The Tower Defense Genre // Problem of Infocommunications. Science and Technology (PIC S&T′2020), Kharkiv, Ukraine.- 6-9, October 2020. doi: 10.1109/PICST51311.2020.9468057

5. Microsoft Research [Електронний ресурс] – Режим доступу до ресурсу: www.microsoft.com/en-us/research/project/trueskill-ranking-system (дата звернення: 01.05.2022)

6. Game Dev Conference Vault [Електронний ресурс] – Режим доступу до ресурсу: www.gdcvault.com/play/1023799/Level-Design-Saga-Creating-Levels (дата звернення: 01.05.2022)

7. Wendy Despain. 100 Principles of Game Design – New Riders, 2012. C.70–71. ISBN 0-321-90249-1

8. Data Driven Gameplay Elements. [Електронний ресурс] – Режим доступу до ресурсу: https://docs.unrealengine.com/en-US/Gameplay/DataDriven/index.html (дата звернення: 01.05.2022)

9. Best Practices: Five Tips for Better Playtesting. [Електронний ресурс] - Режим доступу до ресурсу: https://www.gamedeveloper.com/design/best-practices-five-tips-for-better-playtesting (дата звернення: 01.05.2022)

10. Playtesting for indie studios – 20th International Academic Mindtrek Conference, 2016. doi: 10.1145/2994310.2994364

11. A/B Testing for Game Design Iteration: A Bayesian Approach. [Електронний ресурс] - Режим доступу до ресурсу: https://www.gdcvault .com/play/1020473/A-B-Testing-for-Game (дата звернення: 01.05.2022)

12. The Games Market and Beyond in 2021: The Year in Numbers. [Електронний ресурс] - Режим доступу до ресурсу: https://newzoo.com/insights/articles/the-games-market-in-2021-the-year-in-numbers-esports-cloud-gaming/ (дата звернення: 01.05.2022)

13. How to Determine if Your A/B Test is Statistically Significant. [Електронний ресурс] - Режим доступу до ресурсу: www.investisdigital.com/blog/conversion-rate-optimization/how-determine-if-your-ab-test-statistically-significant (дата звернення: 01.05.2022)