

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерних наук _____

(повна назва)

Кафедра _____ програмної інженерії _____

(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти _____ перший (бакалаврський) _____

Ігровий програмний застосунок у жанрі космічного симулятора.
Алгоритм поломок корабля, балансування, механіки поломок, level design.
(тема)

Виконав:

студент 4 курсу, групи ПЗПІ-20-6 _____

_____ Фільчаков Д.О. _____

(прізвище, ініціали)

Спеціальність 121 – Інженерія програмного
забезпечення _____

(код і повна назва спеціальності)

Тип програми _____ освітньо-професійна _____

Освітня програма Програмна інженерія _____

(повна назва освітньої програми)

Керівник _____ ст.викл. Новіков Ю.С. _____

(посада, прізвище, ініціали)

Допускається до захисту
Зав. кафедри _____

(підпис)

_____ З.В.Дудар _____

(прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наукКафедра Програмної ІнженеріїРівень вищої освіти перший (бакалаврський)Спеціальність 121 – Інженерія програмного забезпечення
(код і повна назва)Тип програми освітньо-професійнаОсвітня програма Програмна інженерія
(повна назва)

ЗАТВЕРДЖУЮ

Зав кафедри _____
(підпис)

« ___ » _____ 20__ р.

ЗАВДАННЯ**НА КВАЛІФІКАЦІЙНУ РОБОТУ**студентові Фільчакову Денису Олександровичу

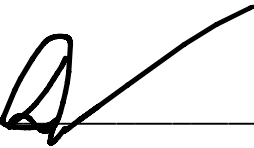
(прізвище, ім'я, по батькові)

1. Тема роботи: Ігровий програмний застосунок у жанрі космічного симулятора. Алгоритм поломок корабля, балансування, механіки поломок, level design
затверджена наказом університету від 20 травня 2024 р. № 471Ст
2. Термін подання студентом роботи до екзаменаційної комісії 07 червня 2024 р.
3. Вихідні дані до роботи: Розробити алгоритми для функціонування механік поломок корабля, розробити балансування всіх елементів гри, налаштувати та протестувати виконану роботу в програмному продукті.
4. Перелік питань, що потрібно опрацювати в роботі: Вступ, аналіз предметної галузі, формування вимог до програмної системи, архітектура та проектування програмного забезпечення, опис прийнятих програмних рішень, тестування програмної системи.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз предметної галузі	11.04.2024	<i>виконано</i>
2	Створення специфікації ПЗ	15.04.2024	<i>виконано</i>
3	Проектування ПЗ	20.04.2024	<i>виконано</i>
4	Розробка ПЗ	16.05.2024	<i>виконано</i>
5	Тестування ПЗ	23.05.2024	<i>виконано</i>
6	Оформлення пояснювальної записки	28.05.2024	<i>виконано</i>
7	Підготовка презентації та доповіді	31.05.2024	<i>виконано</i>
8	Нормоконтроль, рецензування	02.06.2024	<i>виконано</i>
9	Здача роботи у електронний архів	02.06.2024	<i>виконано</i>
10	Попередній захист	04.06.2024	<i>виконано</i>
11	Допуск до захисту у зав. кафедри	07.06.2024	<i>виконано</i>

Дата видачі завдання 08 квітня 2024 р.

Студент _____  _____ Фільчаков Д.О.
(підпис)

Керівник роботи _____ ст.викл. Новіков Ю.С. _____
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи бакалавра: 89 с., 66 рис., 5 табл., 11 джерел.

БАЛАНС, ЛЕВЕЛ-ДИЗАЙН, МЕХАНІКИ, ГРА, КООПЕРАТИВНИЙ РЕЖИМ, ОПТИМІЗАЦІЯ.

Об'єкт розробки – ігровий програмний застосунок в жанрі космічного симулятора.

Мета розробки – створення ігрового програмного застосунку, який дозволяє гравцям будувати тактику під час вирішення проблем на космічному кораблі, з акцентом на балансування гри, дизайні рівнів та механіках поломки модулів.

Метод рішення – середовище розробки Unreal Engine 5, Visual Studio 2022, Microsoft Excel, мови програмування Blueprint та C++.

У результаті розробки створено ігровий програмний застосунок, котрий має три режими – однокористувацький, розділений екран та онлайн кооператив. Гра дозволяє гравцям будувати стратегію для досягнення найкращого результату, розподіляючи ресурси, будуючи маршрути та цілі, а також забезпечуючи баланс гри та ефективний левел-дизайн під час вирішення проблем, що виникають на космічному кораблі, включаючи механіки поломки та їх лагодження.

BALANCE, LEVEL DESIGN, MECHANICS, GAME, COOPERATIVE MODE, OPTIMIZATION.

The object of development is a game software application in the genre of space simulator.

The purpose of the development is to create a gaming software application that allows players to build tactics while solving problems on a spaceship, with a focus on game balance, level design, and module breakdown mechanics.

The method of solution is the Unreal Engine 5 development environment, Visual Studio 2022, Microsoft Excel, Blueprint and C++ programming languages.

As a result of the development, a gaming software application was created that has three modes: single-player, split-screen, and online cooperative. The game allows players to build a strategy to achieve the best result by allocating resources, building routes and goals, ensuring game balance, and effective level design while solving problems that arise on the spacecraft, including breakdown mechanics and their repairs.

Я, Фільчаков Денис Олександрович, студент гр. ПЗП-20-6, здобувач вищої освіти на першому (бакалаврському) рівні кафедри «Програмна інженерія», заявляю: моя кваліфікаційна робота на тему «Ігровий програмний застосунок у жанрі космічного симулятора. Алгоритм поломок корабля, балансування, механіки поломок, level design», що буде представлена в екзаменаційну комісію для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату. Всі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений із 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.....	8
1 Аналіз предметної галузі.....	10
1.1 Розвиток предметної галузі.....	10
1.2 Виявлення проблем та актуалізація рішень.....	13
1.3 Цільова аудиторія.....	14
1.4 Аналіз бізнес-цілей.....	15
1.5 Аналіз потреб користувачів.....	16
1.6 Постановка задачі.....	17
2 Формування вимог до програмної системи.....	18
2.1 Загальний опис.....	18
2.2 Аналіз припущень та залежностей.....	19
3 Архітектура та проектування програмної системи.....	20
3.1 Опис ідеї та створення плану розробки.....	20
3.2 Розробка балансу гри.....	21
3.3 Механіки полумок.....	25
3.4 Левел-дизайн.....	26
4 Програмна реалізація.....	28
4.1 Параметри об'єктів.....	28
4.2 Реалізація левел-дизайну.....	31
4.3 Алгоритм полумок корабля та механіки полумок.....	34
5 Тестування програмного забезпечення.....	37
5.1 Розробка MindMap тестування.....	37
5.2 Розробка тестових випадків.....	38
Висновки.....	42
Перелік джерел посилання.....	43
Додаток А Звіт результатів перевірки на унікальність тексту в базі ХНУРЕ.....	44

Додаток Б Слайди презентації	45
Додаток В Геймдизайн-документ.....	53
Додаток Г Тест-план	70

ВСТУП

Ігрова індустрія вже майже півстоліття невпинно нарощує обсяги виробництва, охоплюючи всі можливі цифрові платформи, які пов'язані з життям кожної людини. Ігри діють комплексно: як нешкідливий спосіб відволіктись від проблем реального світу; як джерело унікального досвіду, як-от наприклад різноманітні симулятори управління машинами, літаками, космічними кораблями; як спосіб розважитись.

У сучасному світі, який дедалі більше обростає інформаційним шумом, ігри можуть стати невід'ємною частиною життя кожного з нас, даючи перепочинок від постійного бомбардування незв'язною інформацією, яку мозок не здатний обробити повністю. Гра дозволяє зосередити увагу на процесі, вигадувати стратегії розв'язку різних ігрових ситуацій.

Будь-яка гра з параметрами має містити в собі ключову складову, а саме баланс. Баланс – певний стан гри, при якій вона не відчувається занадто простою чи нечесною по відношенню до гравця. Гравець буде розчарований грою, яку можна пройти, кликнувши кілька разів по потрібним кнопкам; так само мало кому сподобається гра, яка вбиває тебе при першій-ліпшій нагоді, коли ще не освоєно базові навички (як ходити, як стріляти, як щось змастерити). Тому необхідно приділити цьому аспекту розробки гри певний час, щоб знайти золоту середину між цими двома станами, де гравець буде захоплено просуватися ігровим світом і не захоче виходити з гри до самого її кінця (якщо такий передбачено).

Розглядаючи питання коректного балансу гри з певними параметрами, проводиться комплексна робота з різними її аспектами. Необхідно врахувати, якою є сама гра: для одного гравця чи для кількох; якими є рівні чи локації, прописати та врахувати їх особливості в подальшому. Гра, баланс якої буде розроблено і розглянуто в цій роботі, є космічним симулятором жанру Survival з певними ресурсами та ворогами, тож необхідно провести балансування таким чином, щоб гравець розумів необхідність грамотно та економно витратити надані ресурси.

Результатом кваліфікаційної роботи є розв'язок поставленої розробниками задачі з продумування балансу для комфортної, але непростой гри. Її результатом стануть розроблені алгоритми для обрахування різних параметрів, станів речей, предметів, гравців та ворогів для отримання збалансованого та захоплюючого геймплею, що є метою кожного розробника ігор.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Аналіз предметної галузі

1.1.1 Розвиток предметної галузі

Аналіз предметної галузі, пов'язаної з розвитком кооперативних ігор, відкриває перед нами великі можливості та виклики, пов'язані зі створенням збалансованих та захоплюючого ігрового досвіду для гравців. У сучасному світі, де великий попит на ігри, які можна грати разом з друзями або онлайн-спільнотою, досягнення гармонії у геймплеї та взаємодії між гравцями є важливою задачею для розробників.

Інтеграція інноваційних геймплейних механік у кооперативні ігри дозволяє нам розглядати спільну гру з нової перспективи. Сучасні технології дозволяють створювати складні системи взаємодії між персонажами, розвивати унікальні стратегії гри та навіть створювати динамічно змінні ігрові світи, які реагують на дії гравців.

Зростання популярності онлайн громад, форумів та спільнот геймерів створює можливість для інтеграції кооперативних ігор у ці платформи. Можливість грати разом з іншими гравцями, обмінюватися досвідом та спілкуватися може значно збільшити захоплення гравців та залучити нових учасників до спільноти.

Враховуючи сучасні тенденції у розвитку ігрової індустрії, також важливо зосереджуватися на розробці інструментів для забезпечення балансу між різними персонажами та механіками гри. Використання методів аналізу даних та збору зворотного зв'язку від гравців може допомогти розробити систему, яка буде в змозі оптимізувати геймплей та забезпечувати задоволення від гри для всіх учасників.

Завданням нашої гри буде не лише створення цікавого та захоплюючого ігрового досвіду, але й сприяння спільній взаємодії та командній роботі гравців. Розробка такої гри віддзеркалить не лише різноманітність інтересів геймерів, але й наші зусилля у створенні високоякісних кооперативних ігрових продуктів.

1.1.2 Аналіз існуючих рішень

Прикладом кооперативної гри на чотири гравці з різноманітними параметрами можна навести гру Left for Dead 2. Задача гри – пробігти командою від точки А до точки Б, попутно відбиваючись від навали різних ворогів. Така команда має бути збалансованою: так, у ній є певні ролі, які гравці обирають собі самостійно, коли думають над тим, яку зброю обрати. В залежності від обраної зброї у гравця вибудовується певний геймплей, наприклад, із бензопилою гравець може косити цілі навали ворогів, однак його доволі просто відкинути на землю, звідки битися буде вже непросто, тому така зброя потребує певного рівня навичок.

Баланс у цій грі вибудовується інтранзитивним методом. Це означає, що різні стилі бою доповнюють одне одного в кооперативній грі, і гравці можуть самостійно підтримувати цей баланс, розподіляючи ролі в команді належним чином, банально обираючи людей, що займатимуться дальнім, середнім та ближнім боєм.

Баланс гри є асиметричним. Це означає, що початкові умови гравець обирає самостійно, вибудовуючи геймплей на основі обраної в початковій кімнаті зброї.

Гра має чотири рівні складності: Легкий, Середній, Мастер, Експерт. Кожний з рівнів пропонує свої особливості та параметри ворогів[1].

Наприклад, легкий рівень: як і випливає з назви, цей рівень складності призначений для новачків та недосвідчених гравців. Потрібно зовсім небагато зусиль, щоб перемогти. Заражені завдають мінімальної шкоди; особливі заражені поведуться пасивно. Швидкість їхньої реакції досить повільна, щоб гравець встиг усунути загрозу, доки стало занадто пізно; «Режисер» рідко генерує орди заражених крім кульмінаційних моментів; Відьма (і сидяча, і бродяча) отримує подвоєну шкоду від усіх джерел і починає атакувати тоді і тільки тоді, коли ті, що вижили більше 8 секунд, провокують її на атаку або ж вистрілюють по ній; рядові зомбі завдають 1 одиницю шкоди за удар спереду та 0,5 одиниць за удар зі спини; Танк має 3000 одиниць здоров'я. При успішному ударі по тому, хто вижив, завдає 24 одиниці втрат

— і 75 одиниць, якщо б'є лежачого. Підпалений танк бігає трохи швидше, ніж холодний; саморобні бомби, снаряди гранатомета, балони з пропаном і киснем не завдають шкоди іншим тим, хто вижив, що знаходяться в зоні поразки.

В той самий час середній режим пропонує складніше випробування: Звичайні заражені наносять 2 одиниці шкоди за удар спереду та 1 одиницю шкоди за удар зі спини; Танк має 4 000 очок здоров'я (6 000 в режимі «Бій»); завдає 24 одиниці шкоди при ударі - і 75 одиниць, якщо б'є лежачого, Відьма на 3 секунди швидше (тобто за 5 секунд) лютує і більше не отримує подвоєної шкоди від зброї; постріл з будь-якого типу зброї по своїх нанесе лише 10% від повної шкоди. Курець знімає по 10 очок здоров'я при кожному вдалому ударі під час задушення того, хто вижив. Удар кігтями завдає 2 одиниці шкоди. Час реакції для захоплення – 2,5 секунди; Мисливець завдає по 5 одиниць втрат кожні півсекунди при успішному кидку на жертву. Удар кігтями знімає 10 одиниць здоров'я.

Як можемо побачити, для кожного рівня складності маємо унікальні параметри для ворогів та зброї.

Також варто відмітити левел-дизайн карт: вони розраховані на те, щоб гравці шукали різні стратегічні позиції, з яких буде легше відбиватися від зомбі. Тобто, гра змушує шукати найбільш вигідне розташування гравців з різними типами геймплею для досягнення найбільш ефективного результату з найменшими втратами[2].

1.1.3 Аналіз способів подальшої монетизації

Опираючись на аналогічні проекти та особливості нашого продукту, основним методом монетизації буде продаж гри у Steam.

Основний функціонал гри буде доступний для всіх користувачів, забезпечуючи широке охоплення аудиторії. Проте, можна розглянути додаткові можливості для підвищення доходу, такі як випуск додаткового контенту у формі DLC (збільшення

обсягу гри, нові рівні, персонажі тощо) або спеціальних видань гри з ексклюзивними матеріалами.

Варто також врахувати можливість акцій та знижок для привертання нових гравців та збільшення продажів. Це може стати ефективним інструментом маркетингу та просування гри на ринку.

Головною метою є забезпечення якісного геймплею та задоволення потреб користувачів, а також підтримка стабільної та прибуткової діяльності проекту.

1.2 Виявлення проблем та актуалізація рішень

Після проведення аналізу предметної галузі та вивчення існуючих аналогів, ми розпочали дослідження для визначення цільової аудиторії та ідентифікації можливих проблем, що можуть виникнути для гравців нашої кооперативної гри. За допомогою опитування серед групи потенційних гравців, ми отримали більш глибоке розуміння їхніх потреб та очікувань. Цей підхід дозволив нам налаштувати наші ідеї з урахуванням реальних потреб та вимог користувачів у грі. В результаті були виявлені ключові проблеми, такі як складнощі з комунікацією та координацією між гравцями, нестача ресурсів та складнощі у вирішенні певних завдань гри, тобто нестача необхідних маркерів. Для вирішення цих проблем ми плануємо розробити гру, яка не лише відображатиме специфіку вимог гравців, але й надасть їм можливість спільно працювати над вирішенням завдань та виживанням у складних умовах. Система спільної комунікації, розподілу ресурсів та співпраці має на меті створити захоплюючий імерсивний досвід для всіх чотирьох гравців.

1.3 Цільова аудиторія

Цільова аудиторія для розроблюваної кооперативної гри-сурвайвалу орієнтована на широкий спектр гравців віком від 12 до 27 років. Гра зацікавить командних гравців, а також тих, хто любить динамічний геймплей та швидке прийняття рішень.

Користувачі, що шукають захоплюючий спосіб розваги та відпочинку.

Опис: особи, які люблять грати в ігри та шукають захоплюючі відчуття під час гри разом з друзями чи рідними.

Орієнтація: гра надає можливість користувачам насолоджуватися захоплюючим кооперативним геймплеем та спільними пригодами з друзями або рідними.

Користувачі, які активно користуються месенджерами:

Опис: гравці, які впевнені у використанні месенджерів та шукають ігрові варіанти, які легко інтегруються з цією платформою.

Орієнтація: гра пропонує можливість інтегрувати її в месенджер, що дозволяє гравцям зручно спілкуватися та координувати свої дії в грі безпосередньо через цю платформу.

Гравці, які шукають захоплюючий геймплей та цікаві виклики:

Опис: люди, які цінують емоційність інтенсивних ігрових сценаріїв та шукають захопливі враження від гри.

Орієнтація: гра пропонує простий та зрозумілий інтерфейс, що дозволяє гравцям швидко освоїти гру та насолоджуватися її геймплеем без особливих перешкод чи складнощів.

Цільова аудиторія гри різноманітна за ігровими вподобаннями та очікуваннями, тому гра розробляється з урахуванням потреб кожного гравця, щоб кожен учасник міг насолоджуватися ігровим процесом та отримувати задоволення від гри.

1.4 Аналіз бізнес-цілей

Аналіз бізнес-цілей для розроблюваної кооперативної гри-сурвайвалу включає наступні ключові аспекти.

БЦ-1: Забезпечення захоплюючого геймплею. Створення гри з захоплюючим сюжетом та динамічними ігровими сценаріями, що привертають увагу гравців і забезпечують їм цікаві виклики.

БЦ-2: Підтримка командного гравця. Розробка механік та інструментів, що сприяють спільній грі та співпраці між гравцями, щоб стимулювати командну взаємодію та тим самим забезпечити задоволення від гри.

БЦ-3: Забезпечення динамічності та швидкого розвитку геймплею. Реалізація механік, які стимулюють швидке прийняття рішень та активну реакцію гравців на зміни в ігровому середовищі, щоб забезпечити динаміку та непередбачуваність гри.

БЦ-4: Створення можливостей для особистого розвитку. Введення в гру системи прокачки або розвитку персонажів, що дозволить гравцям вдосконалювати свої навички та відчувати особистий прогрес.

БЦ-5: Забезпечення інтеграції з внутрішньоігровим голосовим чатом. Розробка можливості інтеграції гри з голосовим чатом для спрощення спілкування між гравцями та забезпечення легкого доступу до гри.

БЦ-6: Забезпечення безпеки гравців. Вдосконалення системи безпеки для захисту особистих даних гравців та запобігання можливих загроз для їхньої безпеки в грі.

Ці бізнес-цілі спрямовані на створення захоплюючої та цікавої гри, яка задовольняє потреби різноманітної аудиторії та забезпечує безпеку та комфорт гравців.

1.5 Аналіз потреб користувачів

Аналіз потреб користувачів для кооперативної гри-сурвайвалу включає такі ключові вимоги.

ПК-1: можливість командного спілкування. Гравці мають бажання легко та ефективно спілкуватися зі своїми командними товаришами під час гри, щоб координувати дії та розв'язувати завдання разом.

ПК-2: адаптація до різних стилів гри. Система повинна надавати можливість гравцям вибирати різні стилі гри та стратегії взаємодії з ігровим середовищем, щоб вони могли вибрати оптимальний підхід для своєї команди.

ПК-3: візуалізація геймплею. Гравці хочуть отримувати візуальну інформацію, яка допомагатиме їм краще розуміти геймплей та інтеракції з ігровим світом.

ПК-4: генерація стратегій та тактик. Система повинна надавати можливість гравцям генерувати та обговорювати стратегію гри з іншими членами команди для досягнення успіху в грі.

ПК-5: отримання аналітичної та статистичної інформації про гру. Гравці бажають мати можливість отримувати аналіз гри та статистичну інформацію про свої досягнення та результати гри для подальшого вдосконалення своїх навичок.

ПК-6: швидке прийняття рішень. Гравці мають бажання швидко та ефективно приймати рішення в грі, щоб реагувати на зміни в ігровому середовищі та досягати поставлених цілей.

Таким чином, потреби користувачів полягають у відносно невисокому порозі входження до гри, проте постійно тримати їх в напрузі та змушувати знаходити рішення у різних обставинах.

1.6 Постановка задачі

Мета цієї кваліфікаційної роботи – розробка ігрового застосунку, що спрямований на командну роботу гравців. Враховуючи мінімальні вимоги до функціональності програми, ми розробляємо комплексну систему, що надає користувачам цільний додаток з можливістю робити мінімум для отримання максимального ігрового досвіду.

Обрана архітектура системи базується на дворівневій структурі, що включає серверний рівень з централізованим API, клієнтську частину у вигляді користувацького інтерфейсу. Використовуються ефективні інструменти розробки, зокрема середовище Unreal Engine, Visual Studio.

Важливою складовою є оптимізація системи. Необхідно розробити такі моделі та методи оптимізації, щоб забезпечити працездатність додатку при заявлених мінімальних системних вимогах та плавний геймплей.

2 ФОРМУВАННЯ ВИМОГ ДО ПРОГРАМНОЇ СИСТЕМИ

2.1 Загальний опис

Космічні симулятори – це жанр відеоігор, який дозволяє гравцям відчувати себе членом космічного екіпажу та брати участь у керуванні кораблем. Цей жанр пропонує різноманітність геймплейних механік, від симуляції роботи корабля до бойових сцен з ворогами.

Ідея кооперативного космічного симулятора на чотири гравці – це поєднання елементів виживання, командної роботи та управління ресурсами. У цій грі команді необхідно буде лагодити корабель під час польоту та боротися з монстрами, що нападають через пробоїни.

Основні аспекти цього жанру:

- Кооперативний геймплей, де кожен гравець відповідає за свою роль у команді.
- Динамічна система пошкоджень корабля, що створює проблеми для гравців.
- Можливість ремонту пошкоджених частин корабля.
- Бойова система для відбивання атак монстрів.
- Управління ресурсами для ремонту та боротьби з монстрами.
- Різні ігрові режими, такі як кампанія та онлайн режим.
- Збалансований рівень складності, щоб гра була цікавою для всіх гравців.

Додаткові аспекти включають вибір графічного стилю, звуковий дизайн та інтерфейс користувача. Використання Unreal Engine 5 спростить розробку, забезпечуючи безперебійну синхронізацію даних між гравцями. Головне завдання полягає в тому, щоб механіки гри відповідали її концепції та забезпечували цікавий ігровий досвід для гравців.

2.2 Аналіз припущень та залежностей

Аналіз припущень та залежностей для програмної системи кооперативної гри включає наступні елементи:

П.1: Необхідність гнучкої архітектури гри: розробка гри передбачає наявність гнучкої архітектури, що дозволяє легко додавати та модифікувати ігрові елементи та механіки.

З.1: Залежність від інтегрованих середовищ розробки: використання інтегрованих середовищ розробки для створення гри буде ключовим, і важливо припустити доступність та підтримку цих сервісів.

П.2: Можливість мультиплатформенності: гра має бути розроблена з урахуванням можливості запуску на різних платформах, таких як ПК, консолі та мобільні пристрої.

З.2: Розробка ігрових компонентів для різних платформ: розробка ігрових компонентів повинна бути орієнтована на можливість легкої адаптації до різних платформ та середовищ, забезпечуючи єдинообразний ігровий досвід для всіх користувачів.

Такі припущення є властивими для багатьох кооперативних ігор, тому ресурси для виконання умов вже існують і активно використовуються розробниками.

3 АРХІТЕКТУРА ТА ПРОЕКТУВАННЯ ПРОГРАМНОЇ СИСТЕМИ

3.1 Опис ідеї та створення плану розробки

Ідея гри полягає у створенні кооперативного космічного симулятора для 4 гравців, де команда має управляти та лагодити корабель під час польоту, одночасно борючись з монстрами, які проникають через пробоїни, та намагаючись дістатися до пункту призначення навіть у злиденних умовах. Основні елементи гри:

- Гравці потрапляють на ушкоджений космічний корабель, де повинні втекти від переслідування;
- Корабель постійно зазнає пошкоджень, що призводить до пробоїн та поломок в системі корабля;
- Команда повинна взяти на себе певні ролі (капітан, інженер, офіцер, механік), щоб ефективно лагодити корабель, боротися з монстрами та забезпечувати життєзабезпечення;
- Успішність місії залежить від спільної роботи команди, управління ресурсами та оперативного реагування на критичні ситуації. Геймплей:
- Гравець спостерігає світ з першої особи, що дозволяє йому максимально відчувати себе членом екіпажу;
- Кожен гравець має свої унікальні здібності та інструменти, необхідні для виконання його ролі;
- Корабель має різні системи (двигун, електрика, життєзабезпечення), які потребують постійного ремонту та контролю;
- Бойова система передбачає використання різної зброї та тактики для подолання монстрів.

Вищеназвані елементи необхідно якісно спроектувати, щоб швидко і без проблем імплементувати всі ідеї в ігровий процес.

3.2 Розробка балансу гри

Для спрощення проектування балансу гри було створено MindMap. MindMap (рис 3.1) описує основні елементи гри та демонструє їх зв'язки між собою.

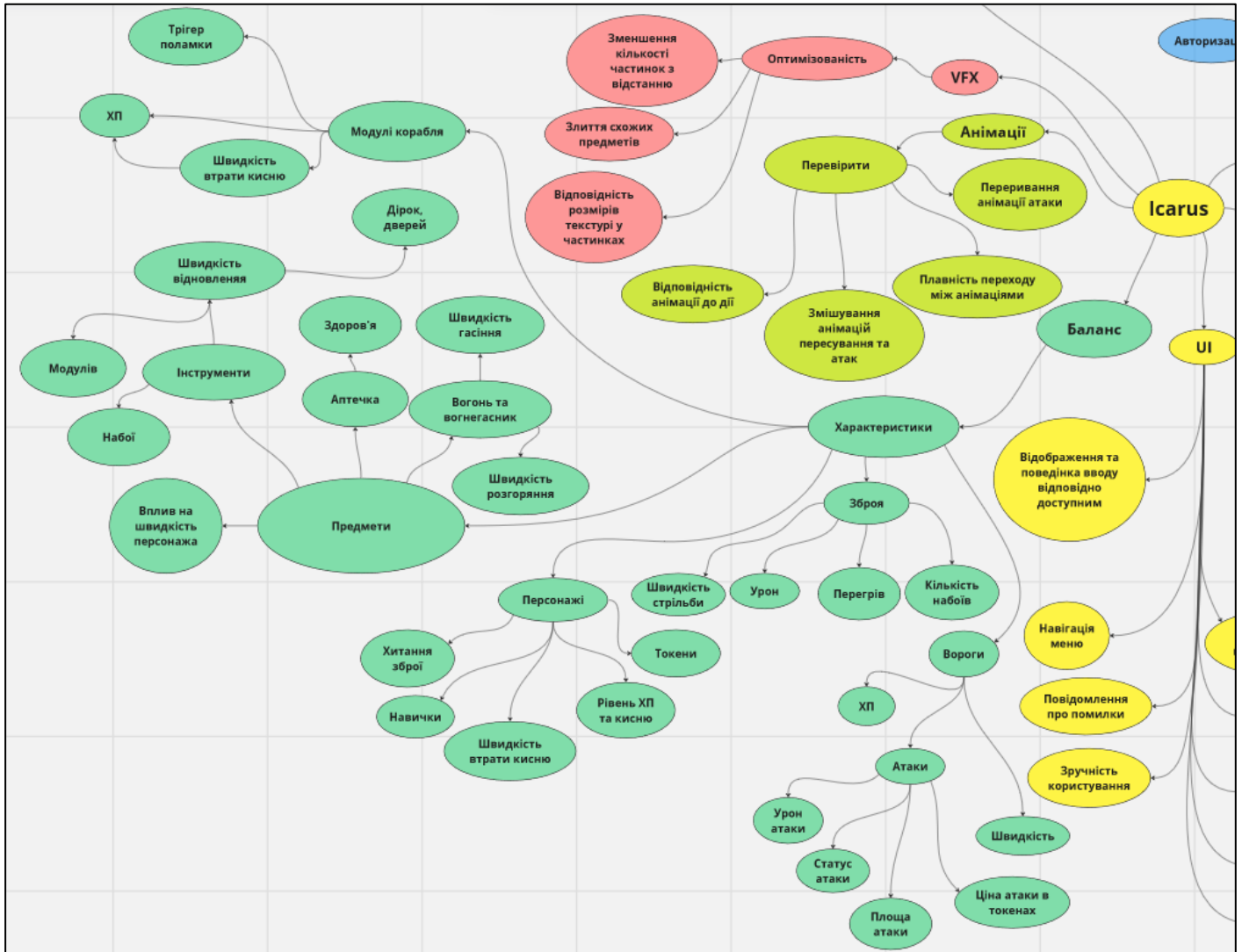


Рисунок 3.1 – MindMap частина балансу гри

На зазначеній діаграмі відображено основні характеристики параметрів гри: зброї, ворогів, персонажів, предметів та модулів корабля. Кожен пункт розбито на підпункти для уточнення конкретних параметрів.

Баланс може бути транзитивним та інтранзитивним[3]. Транзитивний баланс пов'язаний із порівнянням характеристик двох предметів напряму, а інтранзитивний

працює за принципом «камінь-ножиці-папір», тобто заснований на балансі багатьох елементів між собою в єдиній системі[4]. Зогляду на велику кількість елементів у грі було обрано інтранзитивний метод балансування.

Гра пропонує чотири основні ролі для гравця: Капітан, Офіцер, Інженер та Механік. Капітан слідкує за камерами на судні корабля та повідомляє екіпаж про те що відбувається на кораблі. Задача Офіцера полягає в тому, щоб винищувати ворогів, що лізуть в корабель через пробоїни в судні. Задача Інженера та Механіка полягає в тому, щоб оперативно лагодити корабель, а саме латати пробоїни, лагодити двері та апаратуру в модулях.

В цій ситуації найкращим шляхом є упор на балансування команди згідно з ролями, тобто: Офіцер та ефективніше відстрілює ворогів, Інженер та Механік краще справляються з поламками на кораблі, а Капітан має середні характеристики, щоб мати змогу допомагати кожному в команді. На основі цього було розроблено приблизну таблицю вагових коефіцієнтів до характеристик персонажів, збалансовану інтранзитивно. Результати розрахунків наведено в таблиці 3.1.

Таблиця 3.1 Приблизний баланс персонажів.

Роль	Здоров'я	Кисень	Лагодження	Хитання зброї	Токени атаки	Сума
Інженер	1	2	3	-1	1	6
Механік	1	2	3	-1	1	6
Офіцер	2	1	1	0	2	6
Капітан	1	2	2	-1	2	6

Балансування ворогів у кооперативному режимі має свою специфіку. Варто розраховувати на те, що з ворогом може зустрітися не тільки Офіцер, а й гравці з іншими ролями, які менше пасують для винищення прибульців. Загальний баланс було проведено з урахуванням того, що гра має три рівні. Параметром для вирахування складності буде урон всіх ворогів, які проводять одну атаку, поки не буде вичерпано токени гравців, після чого вони чекають на відновлення токена гравця певний час[5]. Було вибрано визначні параметри, що впливатимуть на розрахування складності ворогів, а саме: значення здоров'я, урон атаки ворога, кількість токенів (проведень атак без затримки), кількість ворогів на одному етапі.

Загальна формула має такий вигляд:

$$x = \sum_{i=1}^j \left(\frac{a_1 + a_2 + \dots + a_n}{n} * p_i * h \right)$$

де x – параметр складності;

j – кількість типів ворогів;

a – кількість здоров'я, яке віднімає певна атака;

n – кількість різних типів атак;

p – кількість ворогів на одному рівні;

h – коефіцієнт здоров'я ворога.

На таблиці 3.2 наведено загальні параметри ворогів.

Таблиця 3.2 Загальні параметри ворога

Ворог	h (коефіцієнт здоров'я)	Урон
Кракен	1,00	1
Елієн	1,50	2
Раптор	2,00	6

В таблиці 3.3 наведені розрахунки показників урону, який наносять вороги своїми атаками.

Таблиця 3.3 Показники атак ворогів

Ворог	a1	a2	a3	a4	a5	Середньостатистичне значення урону $(\frac{a_1 + a_2 + \dots + a_n}{n})$
Кракен	5	5	10	15	-	8,75
Елієн	20	20	30	-	-	23,33333
Раптор	20	40	30	80	40	42

В таблиці 3.4 наведено розрахунки приблизного параметру складності для кожного з рівнів, які проходить гравець.

Таблиця 3.4 Розрахунок параметру складності

Кількість на рівні	Етап 1 (p1)	Етап 2 (p2)	Етап 3 (p3)	Урон на етапі 1 $(\frac{a_1 + \dots + a_n}{n} * p_1 * h)$	Урон на етапі 2 $(\frac{a_1 + \dots + a_n}{n} * p_2 * h)$	Урон на етапі 3 $(\frac{a_1 + \dots + a_n}{n} * p_3 * h)$
Кракен	5	6	5	43,75	52,5	43,75
Елієн	0	1	2	0	35	70
Раптор	0	0	1	0	0	84
Сумарно (x)	5	7	8	43,75	87,5	197,75

Судячи з отриманих результатів, другий етап буде складнішим за перший в 1.6 рази, а третій етап буде складнішим за другий на 1.725 рази. Планка складності піднімається з часом гри, але можливо необхідно переглянути деякі параметри, щоб згладити підвищення складності на наступних етапах.

3.3 Механіки поломок

Час від часу секції корабля будуть отримувати поломки. Поломки можуть включати пошкодження корпусу, пожежу, поломку модуля або втрату цілого відсіку.

При пошкодженні корпусу відповідний відсік втрачає кисень. Якщо в одному відсіку виникає три або більше пошкоджень, швидкість переміщення в ньому зменшується. При пошкодженні дверей їх неможливо зачинити, що призводить до розповсюдження безкисневості у відкриті відсіки, якщо суміжні з ними відсіки мають пробоїни.

При тривалому неремонтуванні критичних поломок починається пожежа. Коли у модулі починається пожежа, кисень в ньому закінчується, і необхідно використовувати запаси кисню зі скафандру. Якщо пожежу не загасити, вона може перекинутись на інші відсіки.

Модулі корабля напряму впливають на роботу його систем, тому їх пошкодження впливає на якість та працездатність цих систем. Кожен модуль має два рівні поломок – звичайний та критичний.

При втраті відсіку його неможливо відновити, а всі модулі відсіку втрачаються. Деякі відсіки неможливо втратити, а втрата інших може призвести до миттєвої поразки.

Для лагодження пошкоджень необхідні відповідні предмети:

- Пошкодження корпусу та дверей лагодяться зварювальним апаратом (3 секунди).
- Пожежі гасяться вогнегасником (1 секунда за одиницю вогню).
- Пошкодження модулів лагодяться паяльником (звичайне - 4 секунди, критичне - 8 секунд).

Щоб полагодити пошкодження, потрібно знайти місце поломки, яке буде виділене vfx ефектами, і направити на нього зварювальний апарат (для пробоїн корпусу) або паяльник (для пошкоджень модуля). При цьому знижується заряд

відповідного інструмента. Якщо гравець не має потрібного обладнання або не хоче витратити його заряд, можна використати універсальний ресурс (синю ізоменту) для тимчасового ремонту будь-якої поломки на 10 секунд. Для гасіння пожежі потрібно взяти вогнегасник та розпилювати порошок на вогонь.

Усі розхідники в грі поділяються на три категорії:

- Заряди для інструментів: заряди для паяльника та заряди для зварювального апарату.
- Розхідники персонажів: аптечки та балони з киснем.
- Бойові розхідники: амуніція для зброї.

Таким чином, у грі працюватимуть механіки, що спонукатимуть гравця до активних дій та розумного менеджменту ресурсів.

3.4 Левел-дизайн

Корабель розділений на секції, які з'єднані між собою шлюзами або дверима.

План корабля було розроблено на етапі проектування гри, його можна побачити у дизайн-документі проекту (див. рис. В.11).

Палуба є центральною секцією для моніторингу та управління кораблем. Вона оснащена системами спостереження, інструментами для розгерметизації та автопілотом. Система спостереження дозволяє виявляти поломки за допомогою камер та індикаторів. Інструмент розгерметизації дозволяє відстикувати пошкоджені модулі, що є критично важливим для запобігання поширення поломок. Автопілот допомагає уникати пробоїн корпусу, його поломка збільшує ризик пошкоджень.

Секція реактора є критичною для функціонування двигунів. При пошкодженні реактора починається пожежа, яка може призвести до зупинки двигуна. Втрата всіх реакторів призводить до неминучої поразки.

Секція двигуна забезпечує рух корабля. Пошкодження двигуна сповільнює корабель, а критична поломка зупиняє його рух. Втрата одного двигуна значно зменшує швидкість корабля, а втрата двох двигунів призводить до поразки.

Секція охолодження підтримує нормальну роботу реакторів, запобігаючи їх перегріву. Поломка охолоджувальної системи викликає часті перегриви, що призводить до поломок реакторів. Критична поломка або втрата секції значно підвищує ризик виходу з ладу реакторів.

Склад є місцем зберігання всіх необхідних інструментів для лагодження, розхідників та зброї. Ця секція забезпечує екіпаж всім необхідним для підтримки корабля в робочому стані та відбиття атак.

Секція життєзабезпечення відповідає за подачу кисню по всьому кораблю. Поломка системи життєзабезпечення призводить до зникнення кисню у всіх відсіках, окрім палуби, хабу та самої секції. Критична поломка або втрата відсіку призводить до повної відсутності кисню.

Генераторна секція забезпечує освітлення корабля. Поломка генератора викликає блимання світла, а критична поломка призводить до повного вимкнення освітлення. Втрата секції означає постійну відсутність світла на кораблі.

Хаб є основною секцією корабля, з якої можна перейти до інших секцій. Це велика та важлива частина корабля, що забезпечує комунікацію між відсіками.

Докер призначений для стикування з іншими кораблями. Через нього на борт потрапляють загарбники, а також здійснюється торгівля. Секцію неможливо розгерметизувати, що забезпечує її постійну функціональність.

Посадковий відсік використовується для висадки команди на поверхню. Це звичайний відсік, який забезпечує зручний вихід з корабля для проведення зовнішніх місій.

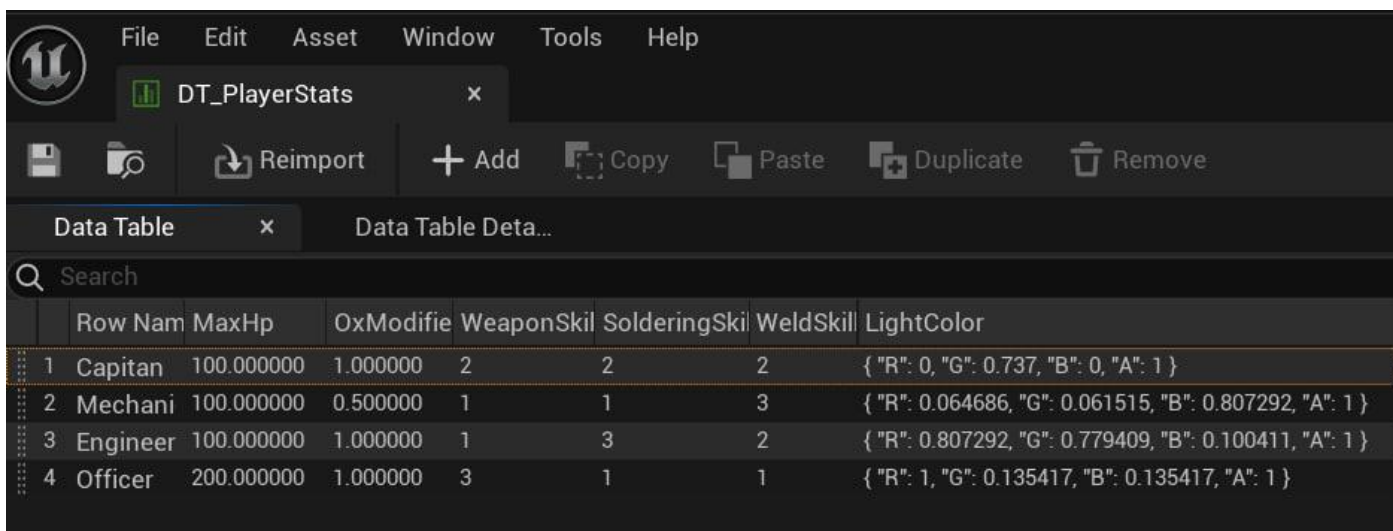
Таким чином, левел-дизайн корабля забезпечує функціональний та зручний розподіл різних систем, які підтримують його життєздатність, забезпечують захист та забезпечують ефективну роботу екіпажу.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ

4.1 Параметри об'єктів

Так як гра розробляється в середовищі Unreal Engine з використанням мови Blueprint, внести прораховані параметри можна швидко і без проблем на місці перевірити, як саме взаємодіють певні цифри певних параметрів один з одним у грі.

Внесемо загальні параметри гравців до файлу DT_PlayerStats, що є частиною динамічних класів і зберігає в собі інформацію про всі класи персонажів. Внесення параметрів зображено на рис. 4.1.

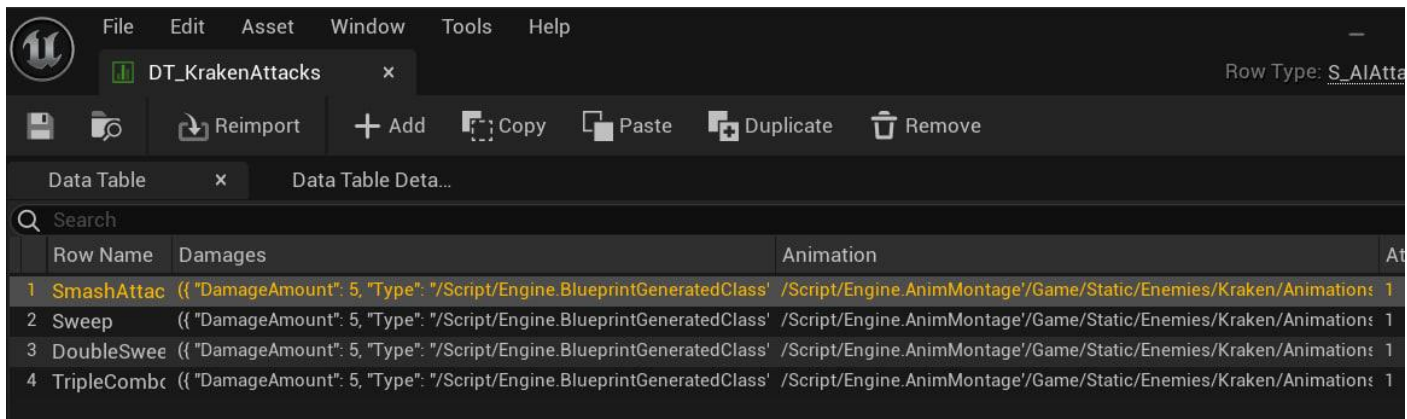


Row Nam	MaxHp	OxModifie	WeaponSkil	SolderingSkil	WeldSkil	LightColor
1 Capitan	100.000000	1.000000	2	2	2	{ "R": 0, "G": 0.737, "B": 0, "A": 1 }
2 Mechani	100.000000	0.500000	1	1	3	{ "R": 0.064686, "G": 0.061515, "B": 0.807292, "A": 1 }
3 Engineer	100.000000	1.000000	1	3	2	{ "R": 0.807292, "G": 0.779409, "B": 0.100411, "A": 1 }
4 Officer	200.000000	1.000000	3	1	1	{ "R": 1, "G": 0.135417, "B": 0.135417, "A": 1 }

Рисунок 4.1 – Внесення параметрів для гравця

Ці параметри пов'язані з уміннями кожного із гравців ефективно користуватись певними механіками гри, тобто, капітан це збалансований персонаж, який може підмінити будь-кого з команди за їх відсутності, офіцер спеціалізується на винищенні потвор, а механік та інженер займаються різними типами поломок на кораблі, яких чимало. У гру також був доданий «коефіцієнт кисню», оскільки механіку часто доводиться працювати у відсіках з пробоїнами, крізь які виходить весь кисень модуля., тому він і отримав скафандр із вдвічі більшим його вмістом.

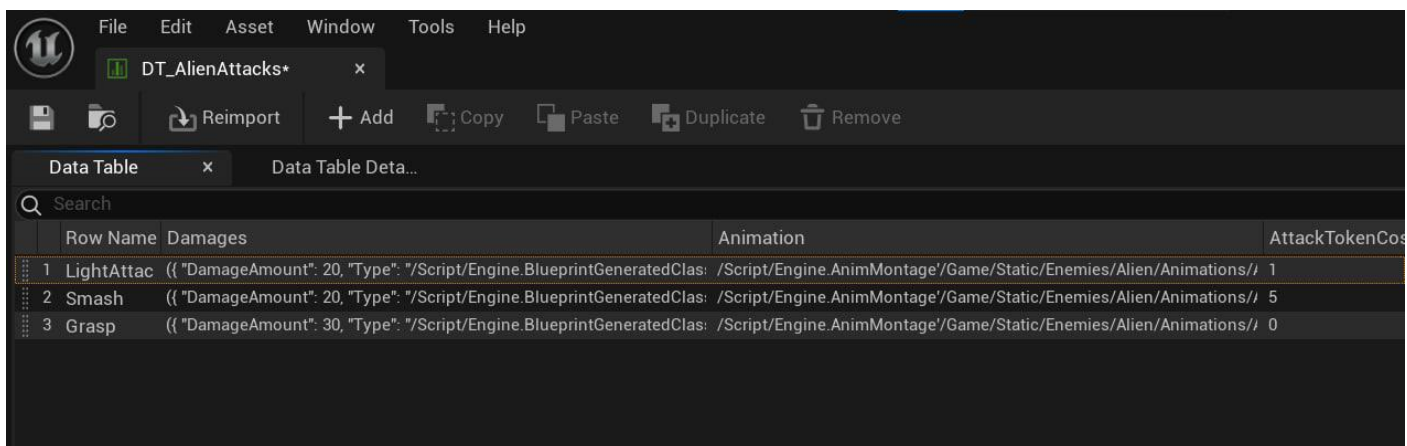
Наступними пропишемо характеристики для кожного з ворогів. Першим є «кракен». Він малий і швидкий, а його атака невелика. Він має чотири різні типи атак, але всі вони не наносять багато шкоди гравцю і в основному дають виключно своїм числом а не якістю, тому такий ворог добре підходить для початку гри. На рис. 4.2 представлені заповнені параметри для нього у файлі DT_KrakenAttacks.



Row Name	Damages	Animation	AttackTokenCos
1 SmashAttac	{ "DamageAmount": 5, "Type": "/Script/Engine.BlueprintGeneratedClass' /Script/Engine.AnimMontage'/Game/Static/Enemies/Kraken/Animations/ 1		
2 Sweep	{ "DamageAmount": 5, "Type": "/Script/Engine.BlueprintGeneratedClass' /Script/Engine.AnimMontage'/Game/Static/Enemies/Kraken/Animations/ 1		
3 DoubleSwee	{ "DamageAmount": 5, "Type": "/Script/Engine.BlueprintGeneratedClass' /Script/Engine.AnimMontage'/Game/Static/Enemies/Kraken/Animations/ 1		
4 TripleComb	{ "DamageAmount": 5, "Type": "/Script/Engine.BlueprintGeneratedClass' /Script/Engine.AnimMontage'/Game/Static/Enemies/Kraken/Animations/ 1		

Рисунок. 4.2 – Внесення параметрів для «кракена»

Для ворога «елієн» атак всього три, проте вони всі є більш потужними. До того ж, у одній з його атак з'являється пасивний ефект, який оглушає гравця на короткий термін часу. З ним необхідно бути обережним і не підпускати до себе близько. На рис. 4.3 показано параметри для цього ворога.



Row Name	Damages	Animation	AttackTokenCos
1 LightAttac	{ "DamageAmount": 20, "Type": "/Script/Engine.BlueprintGeneratedClas: /Script/Engine.AnimMontage'/Game/Static/Enemies/Alien/Animations// 1		
2 Smash	{ "DamageAmount": 20, "Type": "/Script/Engine.BlueprintGeneratedClas: /Script/Engine.AnimMontage'/Game/Static/Enemies/Alien/Animations// 5		
3 Grasp	{ "DamageAmount": 30, "Type": "/Script/Engine.BlueprintGeneratedClas: /Script/Engine.AnimMontage'/Game/Static/Enemies/Alien/Animations// 0		

Рисунок. 4.3 – Внесення параметрів для «елієна»

Останнім та найбільш небезпечним ворогом гри, що з'являється на останньому рівні, є «раптор». Цей ворог є найбільшим, має атак більше за інших, а їх потужність змусить гравця розстрілювати його з-за кута та ухилятися від кожної атаки, які чимало шкодять. Деякі атаки раптора наносять постійний урон, поки хітбокс атаки контактує з гравцем. На рис. 4.4 представлено всі параметри п'яти різних атак «раптора».

Row Name	Damages	Animation	AttackTokenCos
1 Bite	(("DamageAmount": 10, "Type": "/Script/Engine.BlueprintGeneratedClass	/Script/Engine.AnimMontage'/Game/Static/Enemies/Raptor/Animations 1	
2 Posion	0	/Script/Engine.AnimMontage'/Game/Static/Enemies/Raptor/Animations 0	
3 DoubleTentacleStri	(("DamageAmount": 15, "Type": "/Script/Engine.BlueprintGeneratedClass	/Script/Engine.AnimMontage'/Game/Static/Enemies/Raptor/Animations 2	
4 Fatality	(("DamageAmount": 20, "Type": "/Script/Engine.BlueprintGeneratedClass	/Script/Engine.AnimMontage'/Game/Static/Enemies/Raptor/Animations 3	
5 Swing	(("DamageAmount": 10, "Type": "/Script/Engine.BlueprintGeneratedClass	/Script/Engine.AnimMontage'/Game/Static/Enemies/Raptor/Animations 1	

Рисунок. 4.4 – Внесення параметрів для «раптора»

Друга атака «раптора» не наносить урон сама по собі, а є пасивним ефектом, унаслідок якого при спрацюванні навколо ворога на 15 секунд з'являється отруйна хмара, яка кожену секунду наносить шкоду гравцеві, що знаходиться у ній. Всі пасивні атаки ворогів були прописані в окремому класі, що знаходиться у папці DamageTypes серед Blueprints.

Внесені параметри істотно впливають на гру та змушують гравця шукати різні тактики для подолання ворогів в умовах постійного ремонту корабля, борючись з оточенням. Кількість ворогів на різних рівнях регулюється згідно з запланованою складністю гри, щоб тримати гравця в «потоці»[7].

4.2 Реалізація левел-дизайну

Мапу корабля було спроектовано на стадії дизайн-документації, за нею був створена основа для всього світу гри, який замкнений у трирівневому кораблі. За задумкою, таким чином гравцям не одразу вдається зорієнтуватися на місцевості, проте невелика кількість кімнат та спокійний темп гри на початку дають можливість ознайомитися з оточенням і примітити можливі укриття від ворогів чи можливі проблемні місця, до яких часто треба буде повертатися і лагодити їх. Таким чином вибудовується ознайомлення гравця з оточенням та правилами гри.

На рис. 4.5 показано вигляд всієї мапи зовні з боку палуби, де капітан контролює стан корабля по камерах.

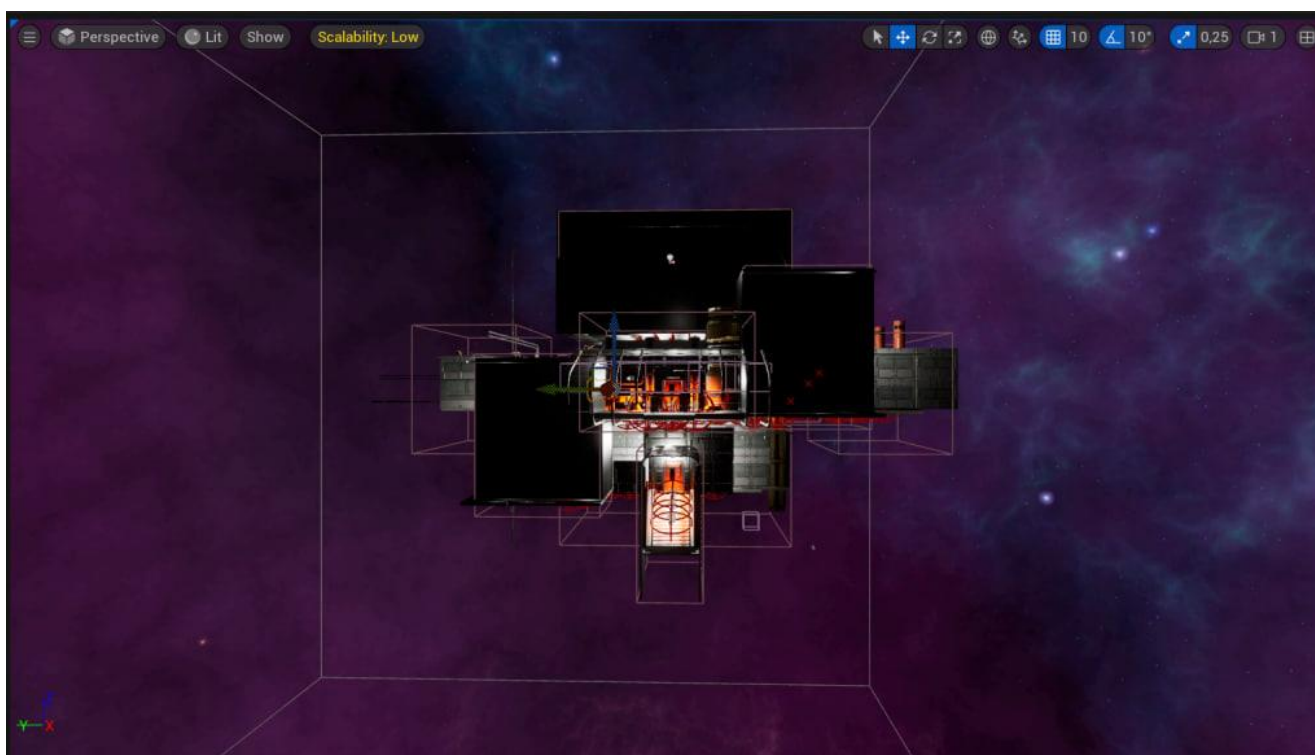


Рисунок. 4.5 – Вигляд мапи гри зовні

Палуба корабля це місце капітана, який слідкує за порядком. Вигляд палуби корабля представлений на рис. 4.6.

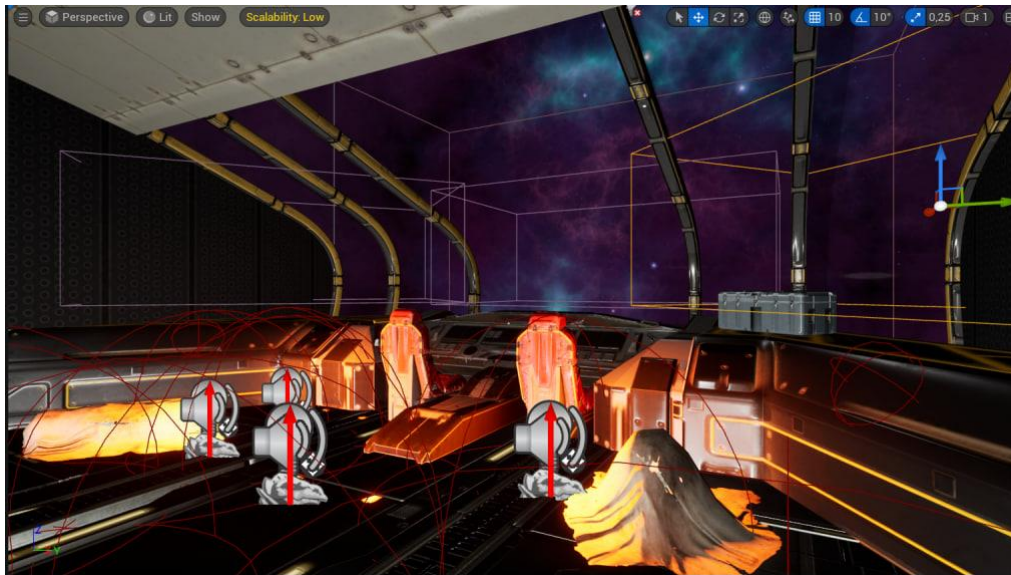


Рисунок. 4.6 – Палуба корабля

Палуба корабля містить в собі панель управління камерами. Капітан може швидко зреагувати на поломку в кораблі, сповістивши про неї гравців через голосовий чат. Вигляд панелі управління представлено на рис. 4.7.

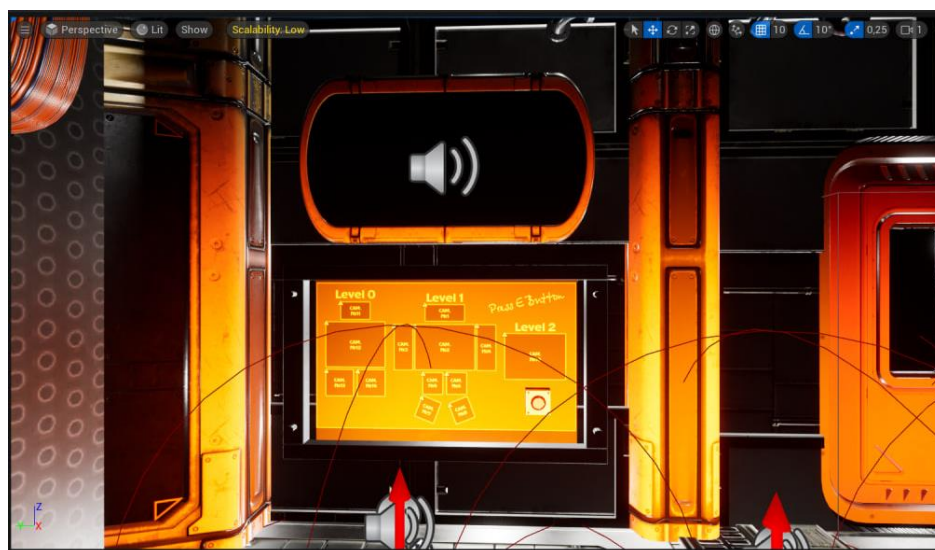


Рисунок. 4.7 – Панель управління камерами на палубі

Хол корабля є широким приміщенням, з якого можна швидко отримати доступ до всіх критично важливих модулів корабля на другому його ярусі. Він також є найбільш пожежонебезпечним, а також не є пустим, а з коробками та хламом, який може стати укриттям від ворогів. Вид холу представлено на рис. 4.8



Рисунок. 4.8. – Хол корабля

Кисневий відсік є важливою частиною корабля. Якщо він не функціонуватиме, всі матимуть обмежений часом запас кисню. Вигляд модулю показано на рис. 4.9.

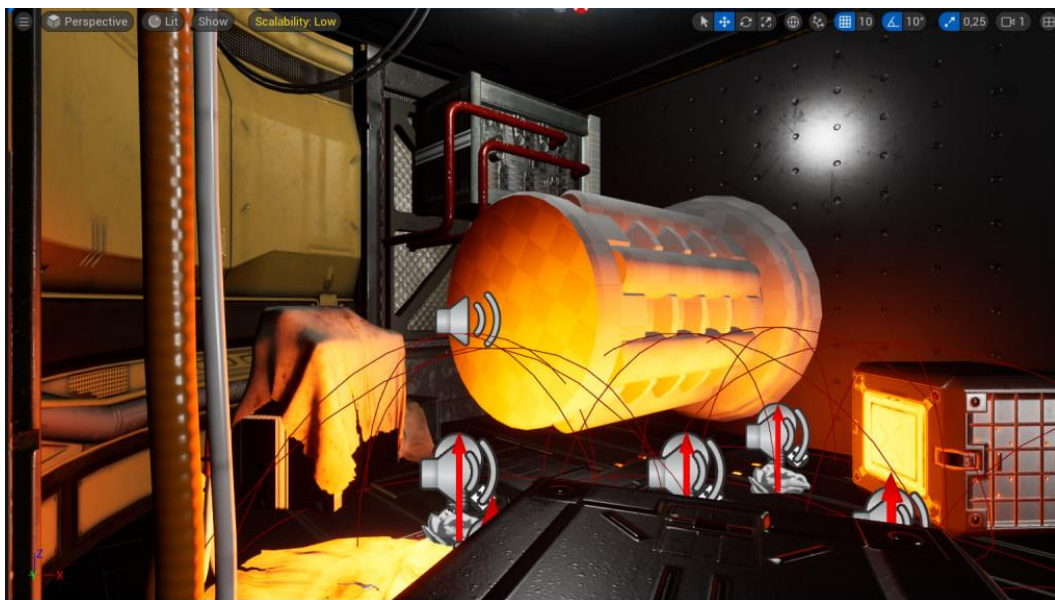


Рисунок. 4.9. – Кисневий відсік

У ході розробки дизайну корабля було також приділено увагу тому, як і де відбуватимуться події. Деякі з них є прописаними у скрипті, на першому рівні складності, для того щоб контролювати процес навчання гравця незнайомим йому механікам. Також прописані події є й на другому і третьому рівнях, але там додається відсоток випадковості події у ході гри.

Існують такі збіги подій, які є дуже неприємними, такі як пробоїни у двох чи більше віддалених відсіках корабля, що буде заскладним викликом для гравця, тому ці події необхідно контролювати.

Вороги також можуть бути заскладними для недосвідченого гравця, особливо якщо їх кілька і вони різні. Їх поведінка контролюється наявністю параметру токенів у гравця, які витрачаються, коли гравець отримує урон. Якщо вартість атаки ворога в токенах перевищує кількість наявних токенів у гравця, то ворог вичікує, поки токени відновляться, просто переслідуючи гравця.

Таким чином, все це разом утворює збалансований підхід до регулювання складності гри на різних рівнях складності, який можна легко змінювати за потреби на етапі тестування.

4.3 Алгоритм поломок корабля та механіки поломок

Для імплементації алгоритму поломок корабля було використано візуальну мову програмування Blueprint. Вона дозволяє візуалізувати процеси, які відбуваються під час гри, за допомогою блоків виконуваного коду, які можна під'єднувати послідовно та паралельно.

Під час написання алгоритму було розроблене рішення, яке складається з двох частин. Перша частина – про випадкові поломки, які можуть статись будь-де і будь-коли, проте між ними має бути певний часовий проміжок. Друга частина – про «заскриптовані», або заплановані поломки, які мають на меті навчити гравця основним механікам на перших рівнях.

На рисунку 4.10 зображений імплементований алгоритм частини випадкових поломок на кораблі.

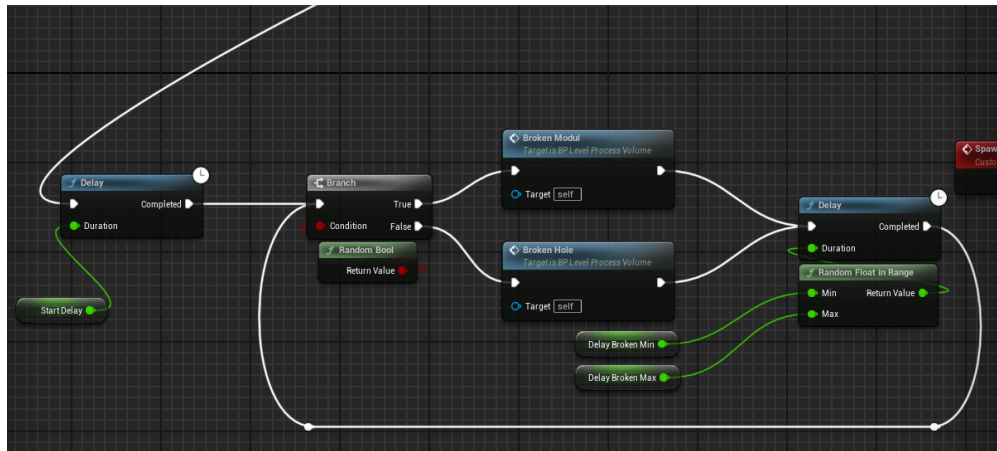


Рисунок 4.10 – Алгоритм випадкових поломок на кораблі

Алгоритм виставляє випадкову поломку через певний часовий проміжок, обираючи поломку модуля чи діру в кораблі. Алгоритм працює до завершення рівня.

Самі механіки поломок детально показано на рис. 4.11.

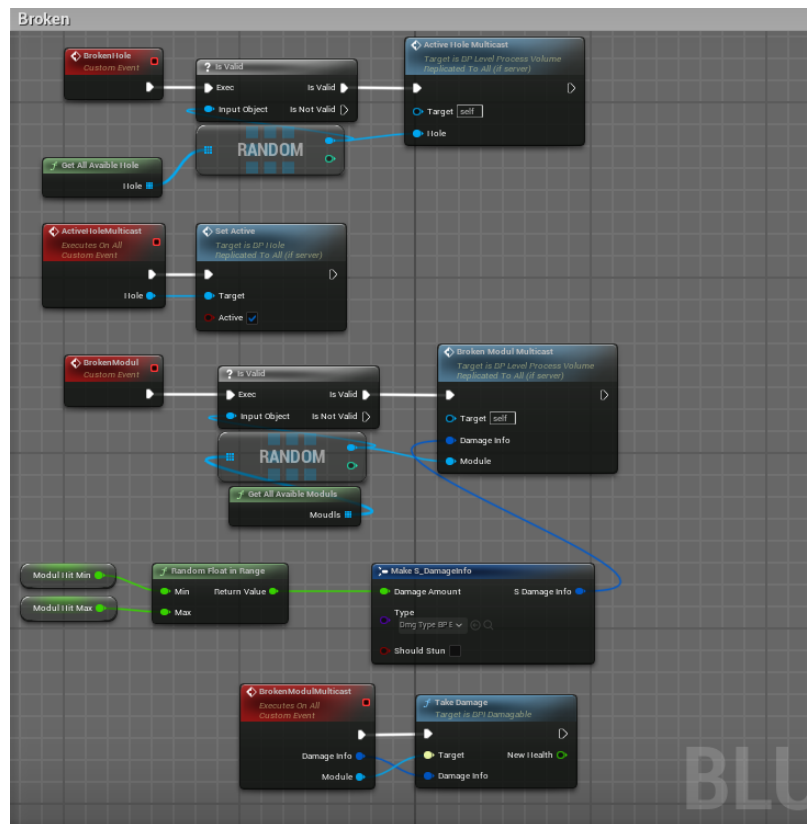


Рисунок 4.11 – Механіки поломок корабля, описані мовою Blueprint

Ця частина коду описує властивості поломок, а саме поломок модулів та дір, а також характеризує їх поведінку. Наприклад, при появі у модулі чорної діри, з модуля зникає весь кисень, а при поломці модуля той перестає функціонувати та викликає пожежу на кораблі.

На мапі виділено основні елементи гри, які підлягають тестуванню. Необхідно провести контроль над тим, наскільки левел-дизайн відповідає потребам гравця, чи є він зручним, чи працює він як частина геймплею («боротьба з оточенням»). Також необхідно проконтролювати доречність кожного параметру гри, щоб гравець вважав гру не надто простою чи складною, а збалансованою. Необхідно перевірити справність механік поламок корабля як основну частину геймплею за механіка та інженера.

Тестування проводиться за принципом «білої скрині» - тестувальник має розуміння роботи логіки системи та знає, як саме все працює, а також має повний доступ до всіх параметрів системи та має змогу у будь-який час змінити їх. Тестування включає в себе мануальні перевірки окремих елементів системи, оскільки результат, якого необхідно досягнути, не є повністю об'єктивним: досягнення захопленості гравця на етапі розробки неможливо передати чіткими формулами.

5.2 Розробка тестових випадків

Для перевірки елементів програмної системи були розроблені тест-кейси. Кожен кейс містить у собі важливу інформацію про те, що саме відбувається, яка серйозність недоліку, описане покрокове відтворення ситуації. Серйозність поділяється на п'ять рівнів, починаючи з S5 – це дрібні помилки в менш важливих системах, і закінчуючи S1 – критичні помилки, які впливають на основний ігровий процес і мають високу ймовірність зустрічі гравцями. Крім серйозності, помилки класифікуються за пріоритетом: від P3 – найнижчий пріоритет, що впливає лише на візуальні аспекти, до P1 – найвищий пріоритет для виправлення помилок, які можуть блокувати ігровий процес. Приклади тестових випадків наведені в таблиці 5.1.

Таблиця 5.1 – Приклади тестових випадків

Тест № 1	
Назва:	Перевірка балансу гри на середньому рівні складності
Опис:	Оцінка складності гри на середньому рівні складності для середньостатистичного гравця
Елемент системи:	SpaceShip
Пріоритет:	S3
Критичність:	P2
Як відтворити:	Повністю пройти середній рівень складності, використовуючи наявні матеріали та ресурси
Очікування:	Гравець стикається з певними труднощами у координації роботи на кораблі та відбивання від ворогів, проте доволі швидко з усім впорається
Фактичний результат:	Вогонь на кораблі з'являється занадто швидко і занадто довго гаситься, що постійно змушує гравця нервувати, аж до відстикування відсіку.
Результат:	Знайдено баг, виправлено 25.04.2024
Тест № 2	
Назва:	Перевірка левел-дизайну на наявність «мертвих зон»
Опис:	Виявлення зон на рівнях, де гравець може застрягти або де відсутні корисні дії.

Продовження таблиці 5.1

Елемент системи:	SpaceShip
Пріоритет:	S1
Критичність:	P1
Як відтворити:	Запустити рівень SpaceShip і уважно пройти кожен закуток розробленої мапи.
Очікування:	Гравець ніде не провалюється крізь текстури і всюди може знайти укриття.
Фактичний результат:	Гравець провалюється під текстури в місцях стику в крайніх кімнатах, у шлюзі, на палубі
Результат:	Знайдено баг, виправлено 25.04.2024
Тест № 3	
Назва:	Перевірка механіки поломок модулів кораблів
Опис:	Необхідно перевірити правильність роботи механіки поломки корабля, процес починки.
Елемент системи:	SpaceShip
Пріоритет:	S4
Критичність:	P3
Як відтворити:	Під час проходження рівня дочекатись поломки у модулі, спробувати полагодити поломку різними інструментами, ізоленотою. Модулі повинні припиняти роботу під час поломки та відновлювати її після починки.

Продовження таблиці 5.1

Очікування:	Механік краще справляється зі своїми поломками, інженер – зі своїми. Ізолента рятує лише на короткий час.
Фактичний результат:	Можна повторно використовувати ізоленту на поломці кожен раз, не витрачаючи час на починку. Таким чином гра стає занадто легкою.
Результат:	Знайдено баг, виправлено 26.04.2024

Таким чином, було проведено тестування окремих елементів системи та виявлено кілька недоліків, які впливали на геймплей. Їх було виправлено в короткі терміни.

ВИСНОВКИ

Під час проведення роботи було здійснено аналіз сфери кооперативних ігор для чотирьох гравців, виявлено проблеми існуючих рішень та визначено завдання розробки нового ігрового продукту з унікальною концепцією.

Було встановлено цільову аудиторію гри та сформульовано вимоги до програмної системи, приділяючи особливу увагу балансу параметрів та характеристик гри.

Проектування включало створення детального опису ідеї гри, розробку плану робіт, визначення ключових ігрових механік, в тому числі системи персонажів, їх характеристик та взаємодії з оточенням, а також розробку інтерфейсу користувача, що відображає важливу інформацію для гравців. Було виконано UML-проектування характеристик гри для подальшого їх балансування. Було створено дизайн-документ гри, на який орієнтується подальша розробка гри.

На основі цього була сформована концептуальна основа для розробки кооперативного космічного симулятора з використанням Unreal Engine 5, де особлива увага була приділена балансуванню параметрів та характеристик гри, левел-дизайну та механікам, що кидають виклик досвідченому гравцеві. Вийшов цікавий, повноформатний ігровий програмний продукт, який потенційно може привернути до себе увагу його цільової аудиторії.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Difficulty | Left 4 Dead Wiki | Fandom [Електронний ресурс] – URL: <https://left4dead.fandom.com/wiki/Difficulty> (дата звернення: 09.03.2024)
2. Jesse Schell, Carnegie Mellon University. The Art of Game Design - 2008. ISBN: 978-0-12-369496-6
3. Баланс гри: 5 уроків для геймдизайнерів-початківців [Електронний ресурс] – URL: <https://vokigames.com/ua/balans-gry-5-urokiv-dlya-gejmduzajneriv-pochatkivcziv/> (дата звернення: 02.05.2024)
4. Ian Schreiber, Brenda Romero. Game Balance – 2021. ISBN: 978-1-49879-957-7
5. Реф Костер. Розробка ігор та теорія розваг. – 2021. ISBN: 978-5-97060-478-6
6. Джейсон Шрейєр. Кров, піт і пікселі. По той бік створення відеоігор. – 2020. ISBN: 978-966-993-265-5
7. Video Game Balance: A Definitive Guide [Електронний ресурс] – URL: <https://gamedesignskills.com/game-design/game-balance/> (дата звернення: 13.05.2024)
8. Design 101: Balancing Game – URL: <https://www.gamedeveloper.com/design/design-101-balancing-games> (дата звернення: 13.05.2024)
9. Game Balance: A Pivotal Issue in Game Design [Електронний ресурс] – URL: <https://www.innovescgames.com/blog/game-balance-a-critical-issue-in-designing-top-titles/> (дата звернення: 13.05.2024)
10. How to test game for balance? [Електронний ресурс] – URL: <https://pinglestudio.com/knowledge-base/how-to-test-for-game-balance> (дата звернення: 14.05.2024)
11. Understanding Balance in Video Games [Електронний ресурс] – URL: <https://www.gamedeveloper.com/design/understanding-balance-in-video-games> (дата звернення: 14.05.2024)

ДОДАТОК А

Звіт результатів перевірки на унікальність тексту в базі ХНУРЕ



Ім'я користувача:
Олійник Олена Володимирівна каф. ПІ

ID перевірки:
1016301314

Дата перевірки:
30.05.2024 22:51:48 EEST

Тип перевірки:
Doc vs Library

Дата звіту:
30.05.2024 22:59:52 EEST

ID користувача:
100012353

Назва документа: 2024_Б_ПІ_ПЗПІ_20_6_Фільчаков_Д_О_скорочений

Кількість сторінок: 39 Кількість слів: 6294 Кількість символів: 50328 Розмір файлу: 1.28 MB ID файлу: 1016096863

2.72%
Схожість

Найбільша схожість: 0.65% з джерелом з Бібліотеки (ID файлу: 1008296726)

Пошук збігів з Інтернетом не проводився

2.72% Джерела з Бібліотеки

212

Сторінка 41

0% Цитат

Вилучення цитат вимкнено

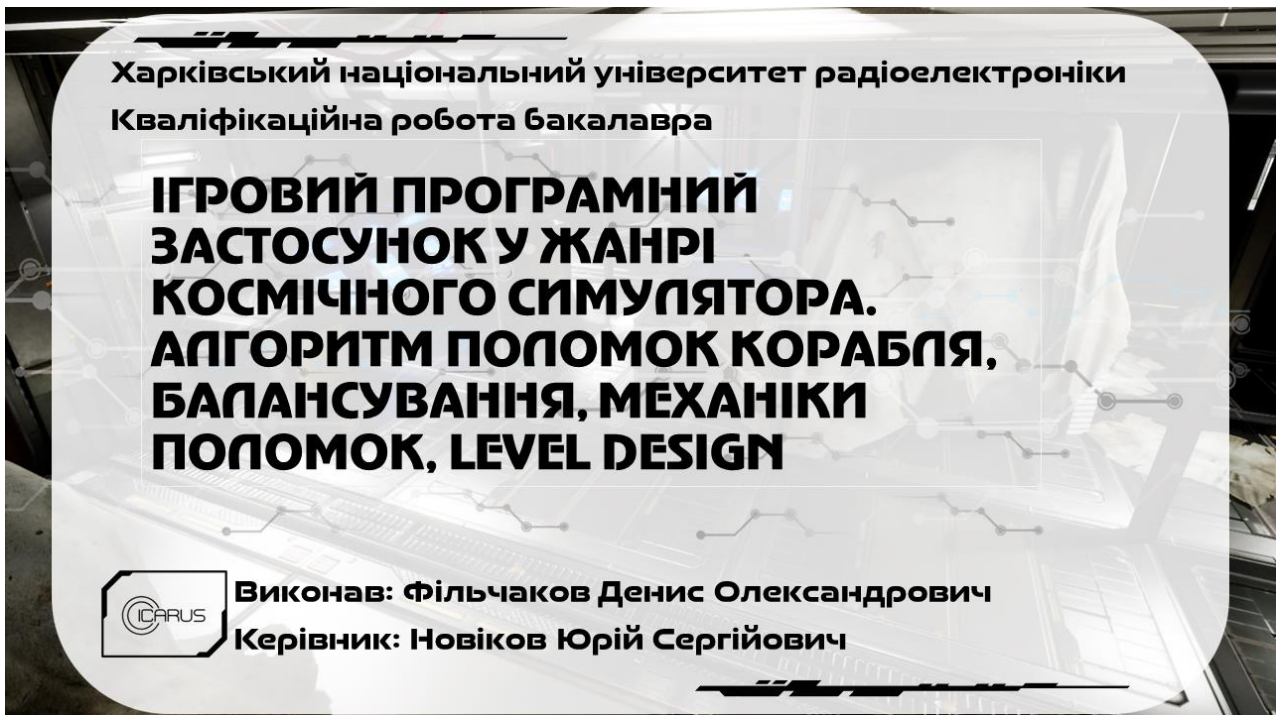
Вилучення списку бібліографічних посилань вимкнено

0%
Вилучень

Немає вилучених джерел

ДОДАТОК Б

Слайди презентації



Харківський національний університет радіоелектроніки
Кваліфікаційна робота бакалавра

**ІГРОВИЙ ПРОГРАМНИЙ
ЗАСТОСУНОК У ЖАНРІ
КОСМІЧНОГО СИМУЛЯТОРА.
АЛГОРИТМ ПОПОМОК КОРАБЛЯ,
БАЛАНСУВАННЯ, МЕХАНІКИ
ПОПОМОК, LEVEL DESIGN**


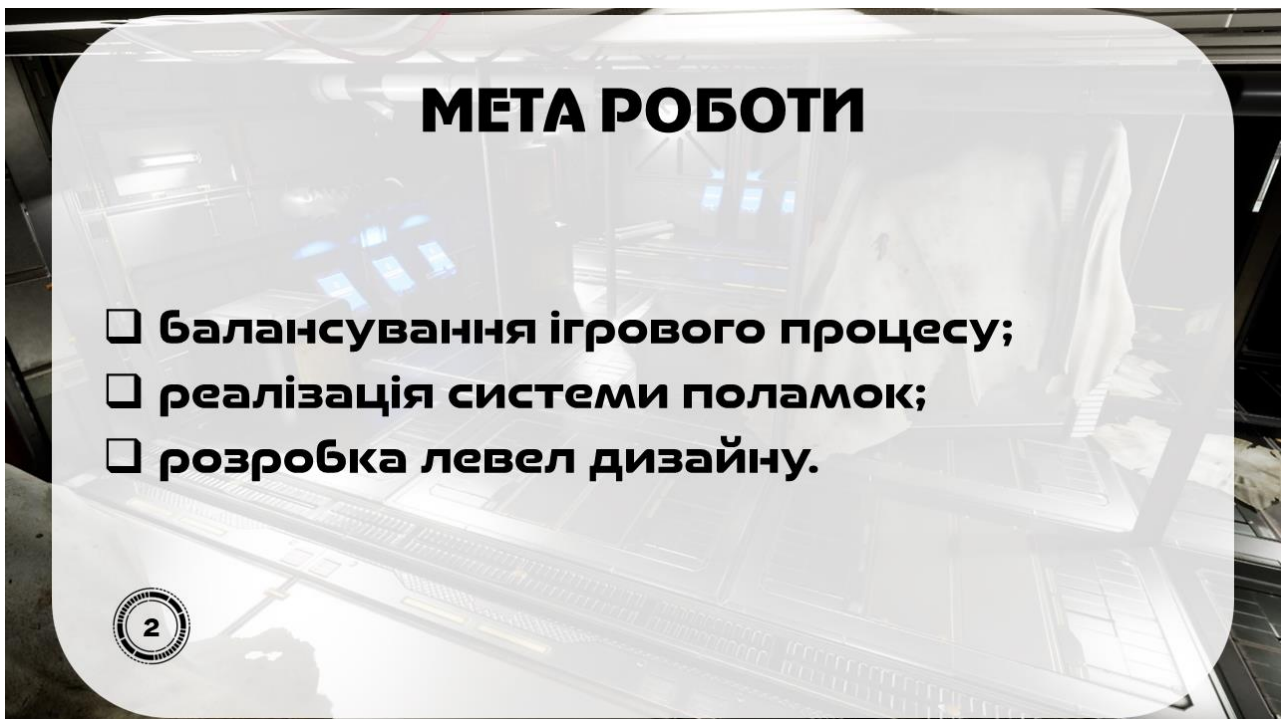
 Виконав: Фільчаков Денис Олександрович
Керівник: Новіков Юрій Сергійович

Рисунок Б.1 – Слайд 1



МЕТА РОБОТИ

- балансування ігрового процесу;
- реалізація системи поламок;
- розробка левел дизайну.




Рисунок Б.2 – Слайд 2

АКТУАЛЬНІСТЬ

- Гравці мають отримувати задоволення від проходження гри
- Нинішній стан ігрової індустрії
- Механіки - основа



Рисунок Б.3 – Слайд 3

ПОСТАНОВКА ЗАДАЧІ

В ході виконання роботи необхідно створити ігровий застосунок «Isagus» в жанрі космічного симулятора з системою кооперативного проходження.

Також необхідно розробити баланс параметрів гри, спроектувати мапу та рівні, розробити механіки та алгоритми для поломок корабля.

Рисунок Б.4 – Слайд 4

АКТУАЛЬНІСТЬ

- Об'єднує людей;**
- Розвиває навички взаємодії;**
- Дозволяє будувати стратегії стратегічні навички;**



Рисунок Б.5 – Слайд 5

ПРО ПРОЕКТ

В ході виконання роботи було створено ігровий застосунок «Icarus» в жанрі космічного симулятора з системою кооперативного проходження.



Рисунок Б.6 – Слайд 6

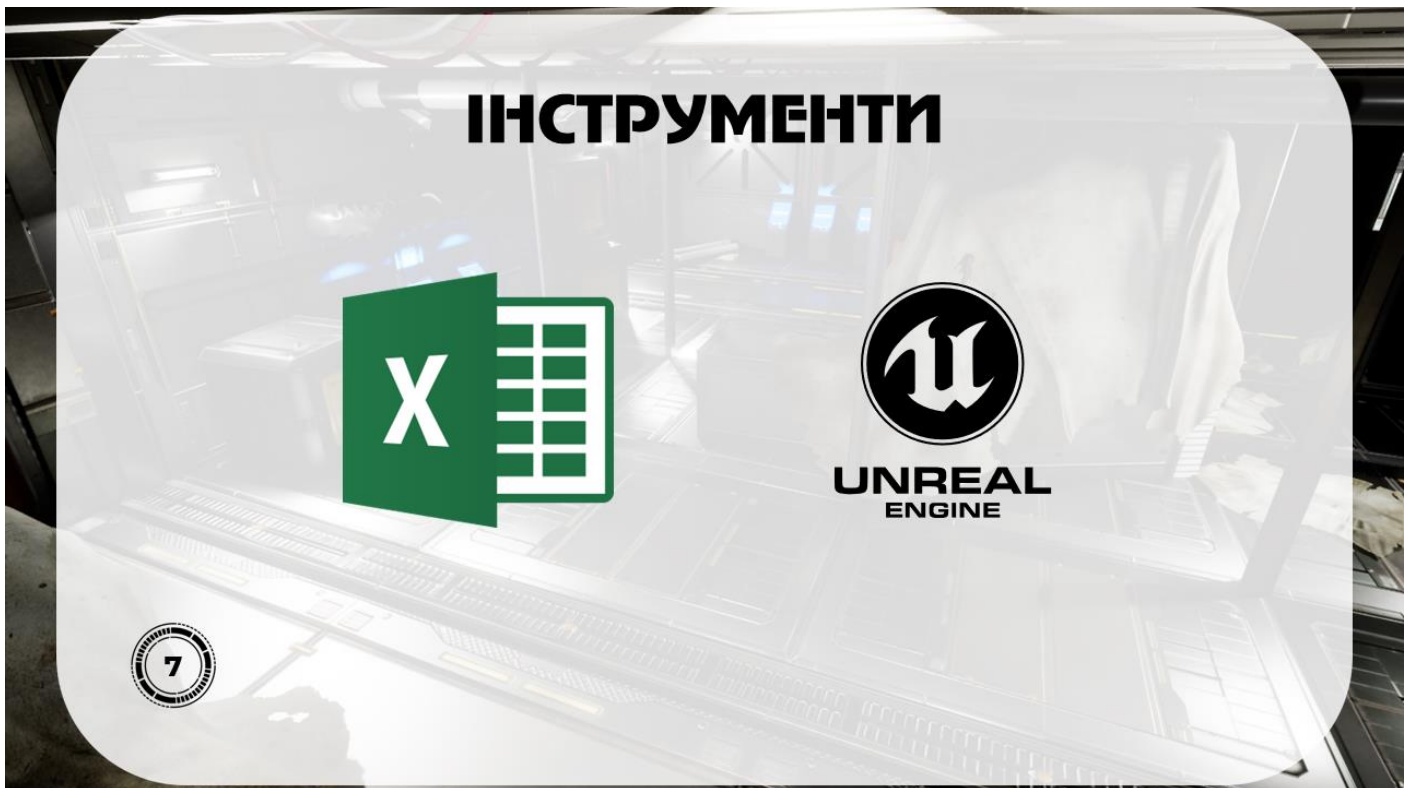


Рисунок Б.7 – Слайд 7

	Хп	Кисень	Лагодження	Хитання	Токени атаки	Сума
Інженер	1	2	3	-1	1	6
Механік	1	2	3	-1	1	6
Офіцер	2	1	1	0	2	6
Капітан	1	2	2	-1	2	6

Рисунок Б.8 – Слайд 8

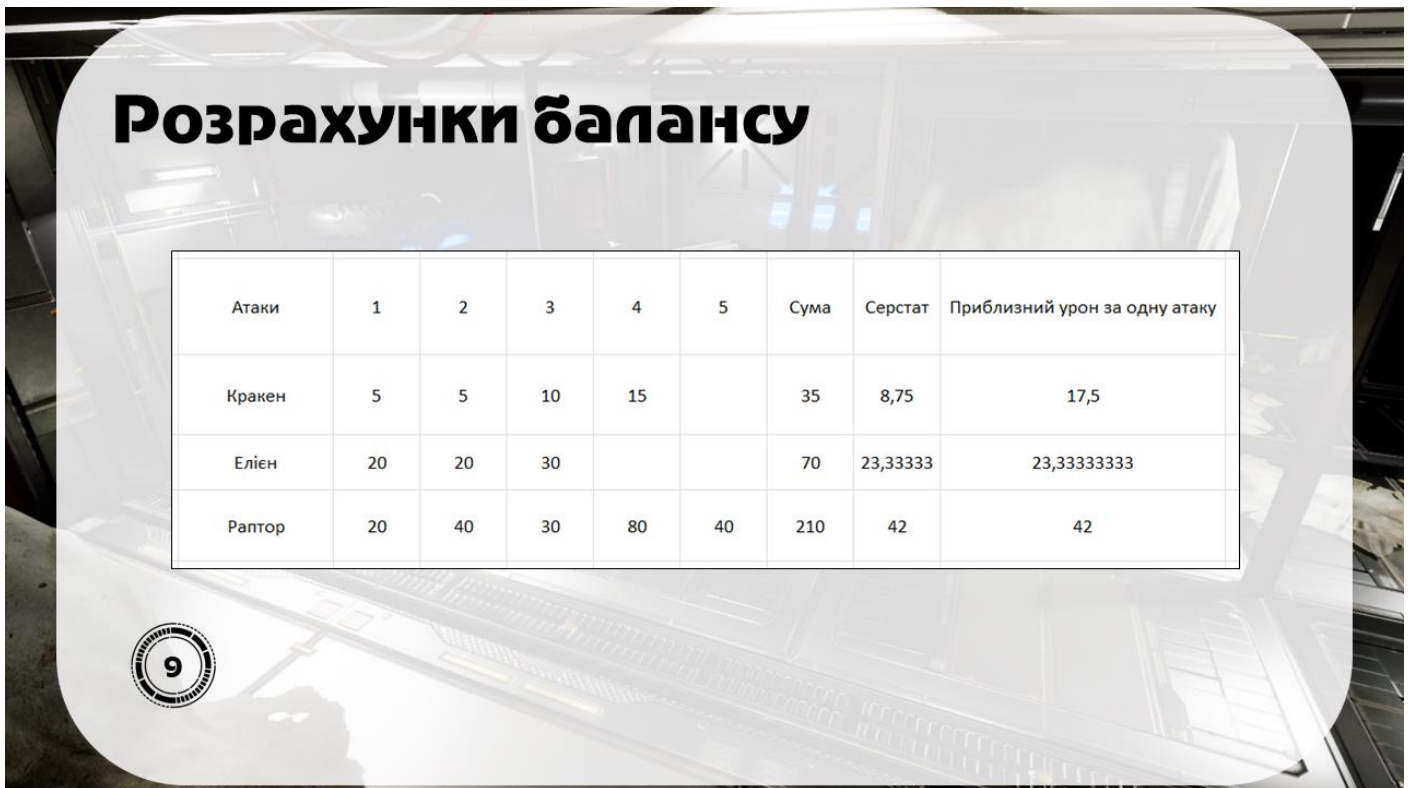


Рисунок Б.9 – Слайд 9

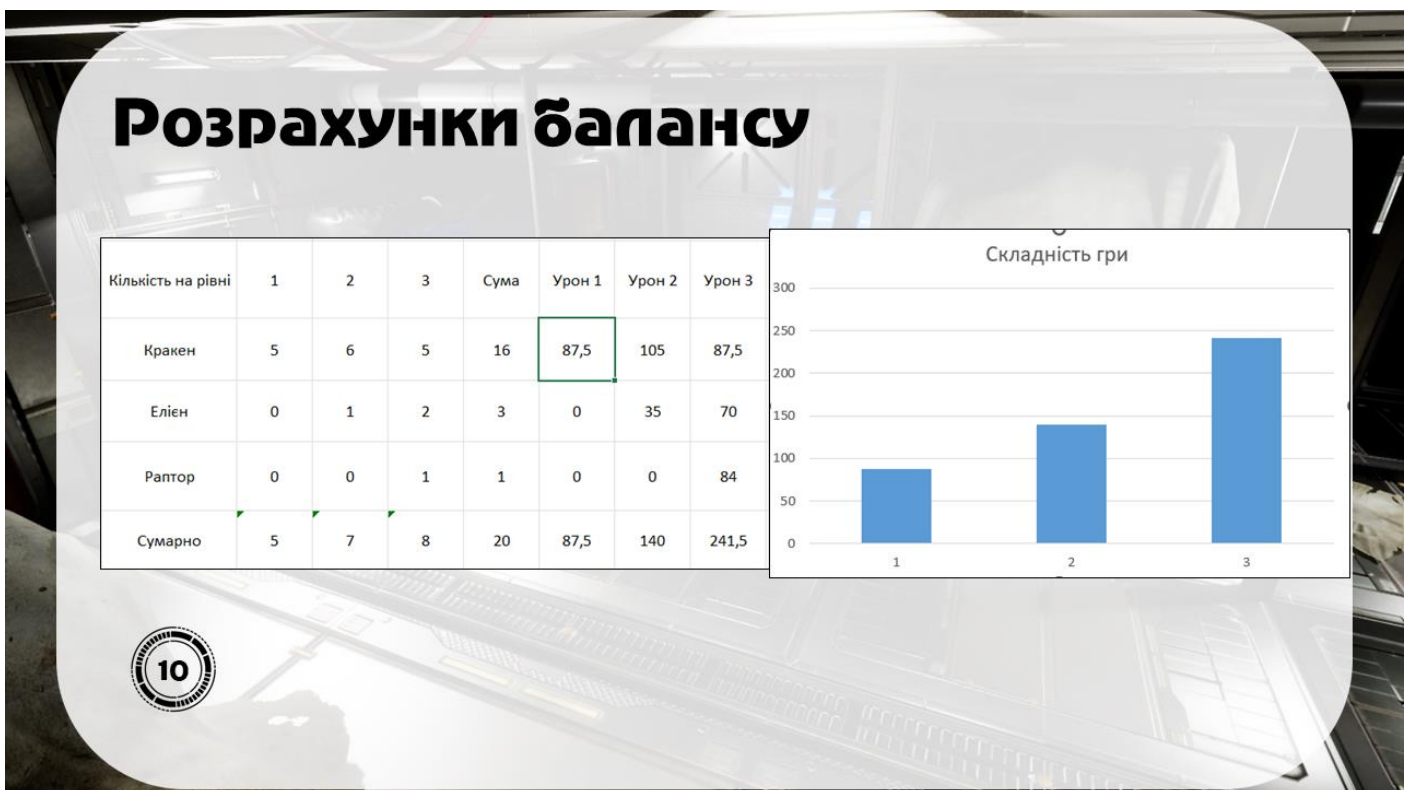


Рисунок Б.10 – Слайд 10

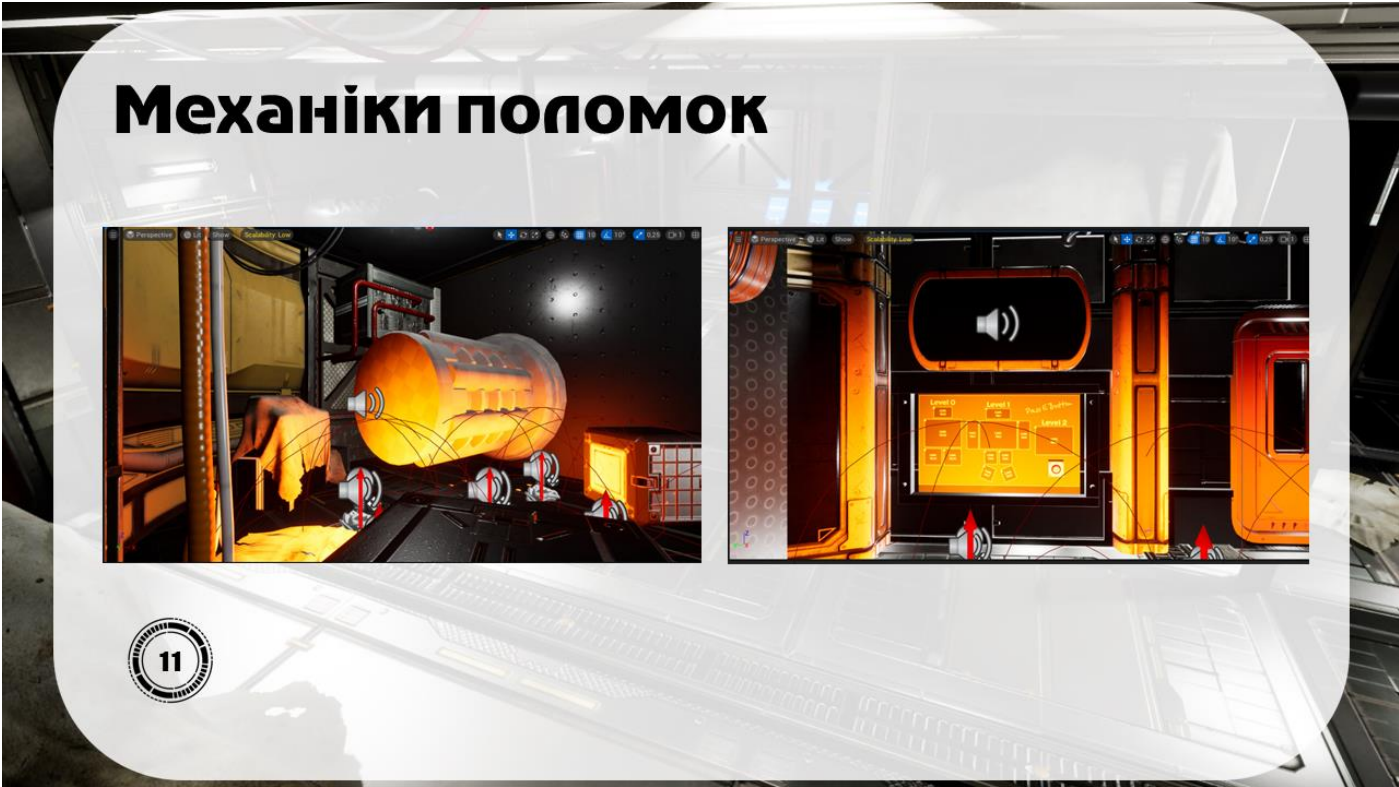


Рисунок Б.11 – Слайд 11

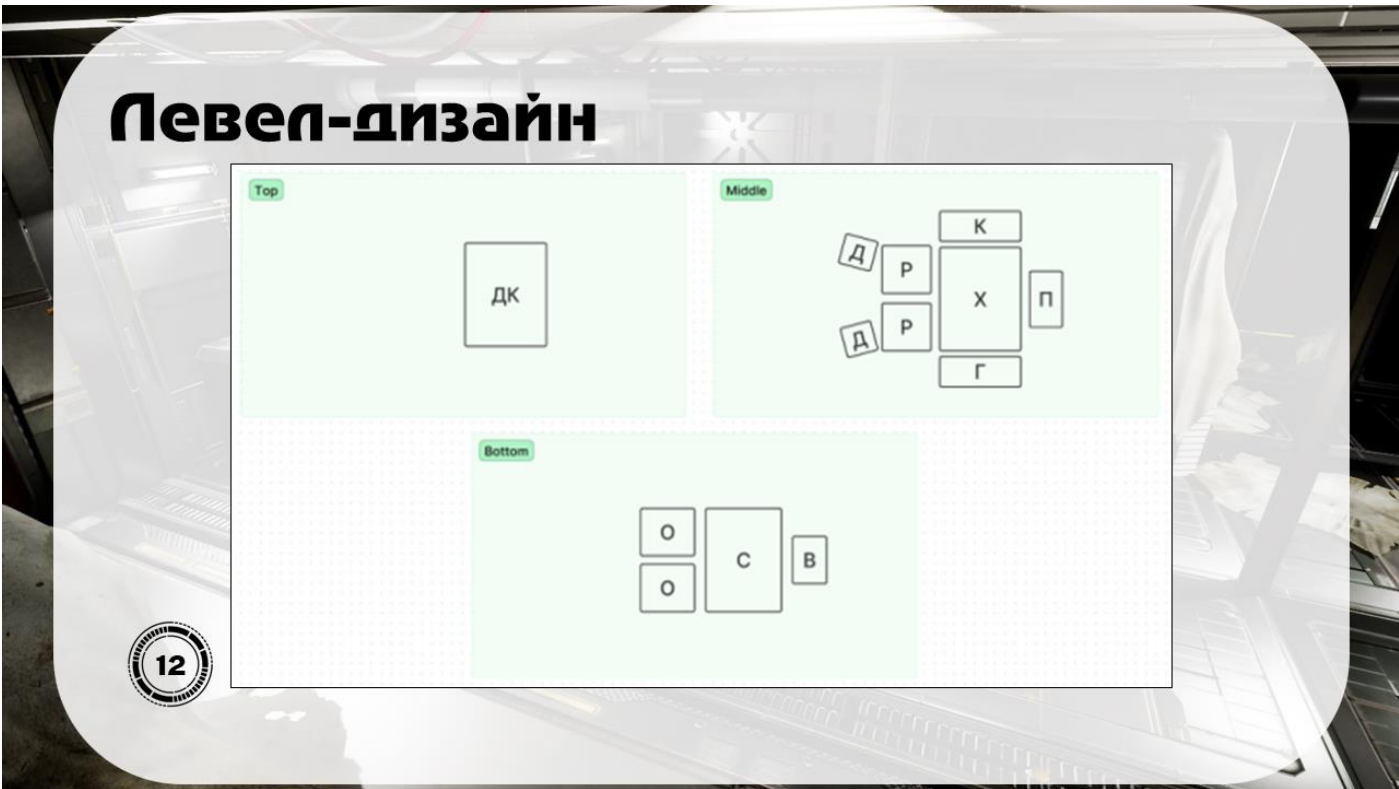


Рисунок Б.12 – Слайд 12

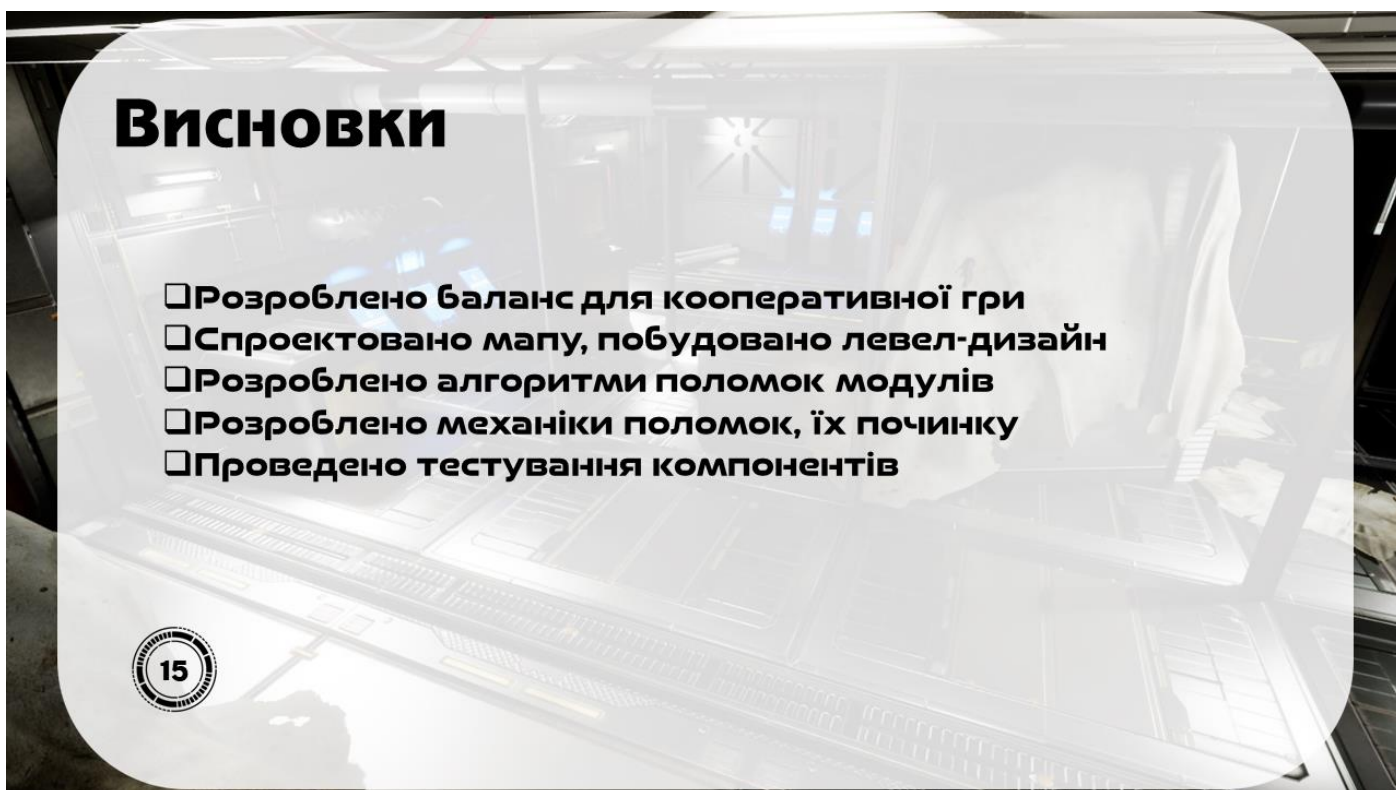


Рисунок Б.15 – Слайд 15

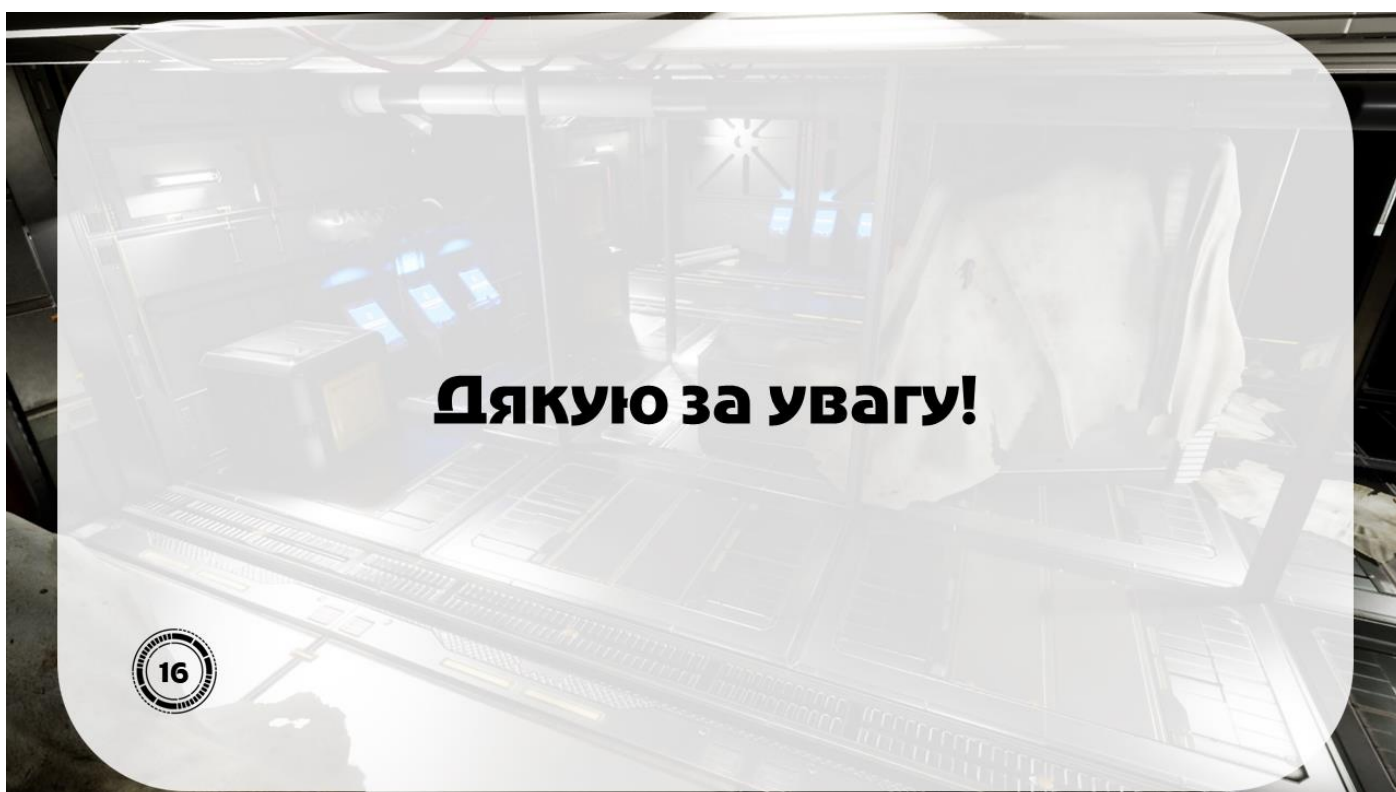


Рисунок Б.16 – Слайд 16

ДОДАТОК В

Геймдизайн-документ



Project Icarus

Game Design Document

by
Черепко Є.Ю.
Яковенко Д.О.
Фільчаков Д.О.

Рисунок В.1 – Сторінка 1 дизайн-документу

1. Загальні положення	3
1.1 Тетра	3
1.2 Введення.....	3
1.3 Цільова аудиторія.....	3
1.4 Unique selling proposition	3
1.5 Час ігрової сесії	4
1.6 Технічні характеристики	4
2. Ігровий процес	5
2.1 Характеристики персонажів.....	5
2.2 Класи персонажів	6
2.3 Механіки	7
2.3.1 Полумки	7
2.3.2 Лагодження систем корабля	8
2.3.3 Розхідники	8
2.3.4 Гроші	9
2.3.5 Магазин	9
2.4 Корабель.....	10
2.4.1 Палуба	11
2.4.2 Секція реактора	12
2.4.3 Секція двигуна.....	12
2.4.4 Секція охолодження	12
2.4.5 Склад	13
2.4.6 Секція життєзабезпечення	13
2.4.7 Генераторна	13
2.4.8 Хаб	13
2.4.9 Докер	13
2.4.10 Посадковий відсік	14
2.5 Зброя	14
3. Референси	16
3.1 Гравці.....	16
4. Інтерфейс	17
4.1 Основний геймплей	17

Рисунок В.2 – Сторінка 2 дизайн-документу

1. Загальні положення

1.1 Тетра

- 1) технологія – ПК;
- 2) механіка – симулятор виживання на космічному кораблі;
- 3) історія – злочинець;
- 4) естетика – Sci-fi.

1.2 Введення

Космічні пірати потрапляють у чергову пригоду, під час якої їх корабль зазнає пошкоджень і з кожною секундою починає розвалюватись. Їм треба досягти фінальної точки до того моменту, як від космічного корабля залишиться сама стеля, об'єднавшись з командою, або взявши все у свої руки. Слідкуйте за станом корабля, лагодіть що можливо полагодити, відбивайтеся від загарбників.

1.3 Цільова аудиторія

Основна аудиторія цієї гри це люди віком від 12 до 27 років. В першу чергу гра зацікавить командних гравців, або людей які люблять грати з рідними чи друзями. Також гра може залучити людей яким подобається динамічний геймплей зі швидким прийняттям рішень.

1.4 Unique selling proposition

- Проривайтеся крізь космічні простори у спробі втекти від космічної федерації на кораблі, що кожної миті розвалюється на шматочки!
- Зберіть команду спеціалістів у кооперативі на розділеному екрані та організуйте свою роботу, або спробуйте вижити на кораблі наодинці;

- Вчасно реагуйте на збої свого космічного судна та відбивайте атаки космічних загарбників.

1.5 Час ігрової сесії

Гра буде поділена на рівні, кожен з яких буде тривати по 10-15 хв. З кожним рівнем складність буде зростати, а в перервах можна буде купити потрібне обладнання та озброєння. Всього буде 3 рівні.

1.6 Технічні характеристики

Мінімальні системні вимоги для ПК:

Windows 10 64-bit, Quad Core 2.4 GHz, 8 GB RAM, GeForce GTX 1050;

Управління: клавіатура та миш, геймпад.

2. Ігровий процес

Гра починається на космічному кораблі, який рухається до наступної контрольної точки. Всього контрольних точок 3, включаючи фінальну. Основна задача гри вижити дійшовши до фінальної точки. Вживанню персонажів буде сприяти стан корабля, а також працюючі модулі на ньому. Під час мандрівки, корабель буде обстрілюватися, що може наносити шкоди корпусу та модулям корабля, екіпажу, а також викликати пожежі (більш детально про загрози на кораблі описано у пункті 2.3.1). Крім того на корабель можуть проникати загарбники, які також несуть загрозу як екіпажу так і кораблю. При досягненні фіналу гри, гравці отримують результат в очках, який буде залежати від кількості грошей, які команда змогла зберегти під час мандрівки, та від цілісності корабля.

2.1 Характеристики персонажів

Ігрові персонажі будуть мати наступні характеристики:

- НР – здоров'я персонажа. При зниженні до 0, персонаж помирає, тим самим вибуває з гри повністю, або якщо гра виконувалась у со-ор інший гравець може викупити нового персонажа в магазині для вільного гравця;
- Кисень. Під час перебування у секції з пожежею або пробоїною він буде знижуватися (кожні 2 сек на 1 од). Після вичерпання запасів кисню, почне знижуватись НР (кожні 1 сек на 1 од), що може призвести до смерті. Якщо персонаж опускає забрало шолома скафандру, то кисень спочатку буде витрачатися з нього.
- Швидкість переміщення. Вона залежить від предмета, який тримає гравець.
- Рівень бойових навичок – впливає на якість персонажа у бою - додатковий урон від зброї, розкид стрільби (хитання ствола), відстань відкидання.

- Рівень навичок Інженера – впливає на здатність персонажу лагодити складні електричні системи, такі як реактор, двигуни, систему охолодження, систему подачі кисню, систему автопілота, генератори, камери та систему спостереження.

- Рівень навичок Механіка – впливає на здатність персонажу лагодити механічні пошкодження корабля, такі як пробоїни, дверей та пожеж.

2.2 Класи персонажів

В грі є чотири класи персонажів: Офіцер, Механік, Інженер, Капітан. У кожного рівень навичок розподілений відповідно свого класу:

В одиночній грі

	Рівень навичок Механіка	Рівень навичок Інженера	Рівень бойових навичок	Здоров'я	Кисень
Капітан	3	3	3	200	200

В коопі

	Рівень навичок Механіка	Рівень навичок Інженера	Рівень бойових навичок	Здоров'я	Кисень
Офіцер	1	1	3	200	100
Капітан	2	2	2	100	100
Механік	3	1	1	100	200
Інженер	2	3	1	100	100

Рисунок В.6 – Сторінка 6 дизайн-документу

2.3 Механіки

2.3.1 Поломки

Час від часу секції корабля будуть отримувати поломки. Поломками може бути пошкодження корпусу, викликана пожежа, поломка модуля, або втрата цілого відсіку.

2.3.1.1 Пошкодження корпусу та дверей

При пошкодженні корпусу відповідний відсік втрачає кисень. При трьох і більше пошкодженнь в одному відсіку, швидкість переміщення в ньому зменшується.

При пошкодженні дверей їх буде неможливо зачинити, що буде призводити до розповсюдження безкисневості у відкриті відсіки, якщо суміжні з ними відсіки мають пробоїни.

2.3.1.2 Пожежа

При тривалому не ремонтуванні критичних поломок, почнеться пожежа. Коли у модулі почалась пожежа, у ньому закінчиться кисень, і потрібно буде використовувати його запаси зі скафандру. При тривалій пожежі, вона може перекинутись на інші відсіки.

2.3.1.3 Поломка модуля

Модулі корабля напряму впливають на ті чи інші його системи, тому їх пошкодження буде впливати на якість та працездатність цих систем. Кожен модуль має два рівні поломок – звичайний та критичний (про особливості поломок дивись відсіки корабля в яких він розташований у пунктах 2.5.1 - 2.5.9).

2.3.1.4 Втрата відсіку

При втраті відсіку, його вже ніяк не можна буде повернути, при цьому втрачаються всі модулі відсіку. Деякі відсіки неможливо втратити, деякі викликають миттєву поразку. Про наслідки для кожного відсіку див. п. 2.5.1 - 2.5.9.

2.3.2 Лагодження систем корабля

Для лагодження пошкоджень потрібні відповідні для цього предмети:

- пошкодження корпусу та дверей лагодяться зварювальним апаратом (3 сек);
- пожежі гасяться вогнегасником (секунда за одиницю вогню);
- пошкодження модулів лагодяться паяльником (звичайне - 4 сек, критичне - 8 сек).

Щоб полагодити пошкодження, треба знайти місце поломку, яка буде виділена vfx ефектами і направити на неї зварювальним апаратом, якщо це пробоїна корпусу, або паяльником, якщо це пошкодження модуля. При цьому знижується заряд відповідного інструмента.

Якщо у гравця в руках немає потрібного обладнання, або він не хоче витратити заряд відповідного обладнання, можна використати універсальний ресурс (синю ізоленту), щоб відремонтувати будь-яку поломку, але на дуже короткий термін (10 секунд).

Щоб загасити пожежу, треба взяти вогнегасник та почати розпилювати порошок на вогонь.

2.3.3 Розхідники

Всі розхідники у грі можна розділити на 3 категорії:

- 1) заряди для інструментів – заряди для паяльника та заряди для зварювального апарату.
- 2) розхідники персонажів – аптечки та балони з киснем;

3) бойові розхідники – амуніція для зброї.

2.3.4 Гроші

На початку гри гравці мають фіксовану суму грошей. Гроші витрачаються у магазинах, а також від неї залежить фінальний результат гри.

2.3.5 Магазин

Магазин відкривається на контрольних точках і доступний біля докера. Товари магазину включають наступні:

Товар	Ціна
заряди для паяльника	15
заряди для вогнегасника	
заряди для зварювального апарату	10
аптечка	40
найняти нового члена екіпажу(воскресити померлого)	100
балони кисню	25
патрони для пістолету	10
напівавтоматичний автомат	100
патрони напівавтоматичного для автомату	15
гвинтівка	200

Рисунок В.9 – Сторінка 9 дизайн-документу

патрони для гвинтівки	20
-----------------------	----

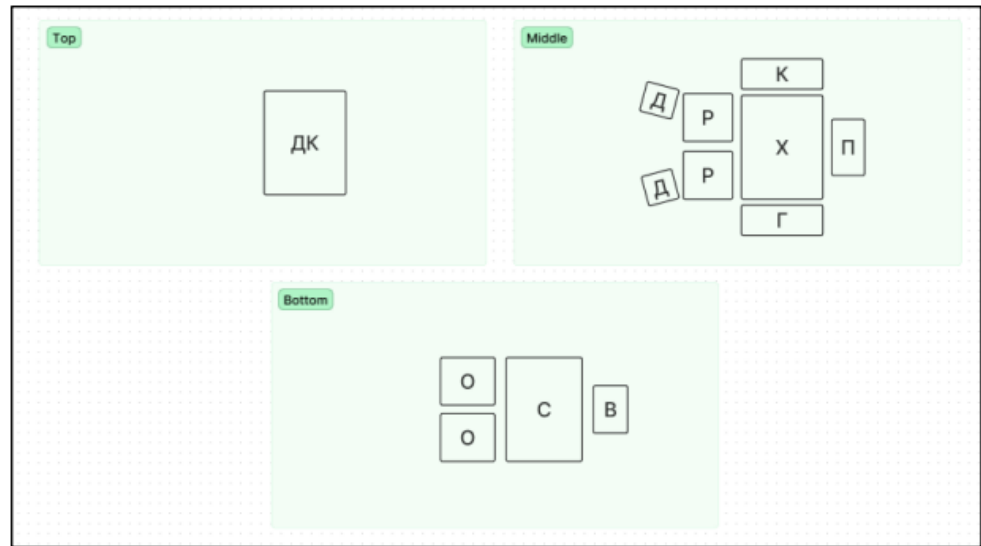
заряди для паяльника;
 заряди для зварювального апарату;
 аптечка;
 найняти нового члена екіпажу(воскресити померлого);
 балони кисню;
 патрони для пістолету;
 напівавтоматичний автомат;
 патрони напівавтоматичного для автомату;
 гвинтівка;
 патрони для гвинтівки.

2.4 Корабель

Корабель розділений на секції, які з'єднані між собою шлюзом, або дверима. До основних відсіки корабля належать: палуба, секція реактора, секція двигуна, секція охолодження, склад, секція життєзабезпечення, генераторна, хаб, докер. Більшість з цих секції мають модулі, які можуть бути пошкоджені.

Загальна схема корабля має наступний вигляд:

Рисунок В.10 – Сторінка 10 дизайн-документу



де П - палуба

Р - реактор

Д - секція двигуна

О - секція охолодження

С - склад

К - секція життєзабезпечення

Г - генераторна

Х - хаб

ДК - докер

В - посадковий відсік

2.4.1 Палуба

Секція, на якій буде розташована система спостереження для аналізу стану корабля та моніторингу вторгнень, інструменти розгерметизування частин корабля та система автопілота.

Система спостереження представлена у вигляді зображень з камер кожного відсіку. Коли виникає поломку у відсіку, індикатор загоряється біля

Рисунок В.11 – Сторінка 11 дизайн-документу

відповідної камери. Дізнатися деталі поломки вже можна через камеру. При поломці, камери перестають працювати, але індикатори працюють. При критичній поломці не працюють як камери, так і індикатори. Також цей відсік неможливо втратити.

Інструмент розгерметизації дозволяє відстикувати модуль корабля, щоб наслідник його поломки цього не перейшли на інші модулі. При його поломці, відстикування модулів корабля буде проводитись повільніше. При критичній поломці, капітан не зможе відстикувати модулі корабля

При поломці автопілота, корабель почне частіше отримувати пошкодження корпусу, тобто пробоїни. В залежності від тривалості поломки, частота появи пробоїн почне збільшуватись.

2.4.2 Секція реактора

На цій секції будуть розташований реактор, який підтримує життєздатність двигуна. При його пошкодженні спалахує пожежа. При виведенні зі строю, зупиняється відповідний двигун. При втраті цієї секції, втрачається секція з відповідним двигуном. При втраті всіх реакторів гра закінчується поразкою.

2.4.3 Секція двигуна

Від двигуна корабля напряму залежить його корабля. Пошкодження двигуна сповільнює корабель, а критична поломка зупинить переміщення корабля. При втраті двигуна, швидкість корабля зменшується вдвічі. Всього на кораблі дві секції з двигунами. При втраті двох двигунів гра закінчується поразкою.

2.4.4 Секція охолодження

На цій секції будуть розташовані системи охолодження реактора. На кораблі дві такі секції, по кожному на реактор. При поломці, реактори час від

часу почнуть перегріватися, що буде викликати їх середню поломку. Якщо ж реактори вже перегріті, то це буде викликати критичну їх поломку.

При критичній поломці, або втраті цієї секції, реактор буде ще частіше перегріватися.

2.4.5 Склад

Секція з усім знаряддям для лагодження, розхідниками та зброєю.

2.4.6 Секція життєзабезпечення

Секція, яка відповідає за подачу кисню на кораблі. При частковій поломці кисень зникає на всіх відсіках корабля, окрім палуби, хабу та самої секція життєзабезпечення. При критичній поломці, або втраті відсіку кисень зникає усюди.

2.4.7 Генераторна

Секція з генератором, який відповідає за освітлення корабля. При поломці, світло на кораблі почне блимати, а при критичній поломці вимкнеться повністю. При від'єднанні цієї секції, світло на кораблі вже ніколи не повернеться.

2.4.8 Хаб

Велика основна секція корабля. Звідси можна перейти до інших секцій.

2.4.9 Докер

Невелика секція корабля, призначена для стикування до інших кораблів. Через неї на борт будуть потрапляти загарбники (у більшості випадках). Також через неї виконується торгівля у магазинах. Не може бути розгерметизована.

2.4.10 Посадковий відсік

Звичайний відсік, через який команда висаджується на землі.

2.5 Зброя

2.5.1 Пістолет

Урон	10
Магазин	16
Скорострільність	3 пулі/сек
Режим стрільби	По 1 пулі



2.5.2 Напівавтоматичний автомат

Урон	14
Магазин	27
Скорострільність	3 пулі/сек
Режим стрільби	По 3 пулі

Рисунок В.14 – Сторінка 14 дизайн-документу



2.5.3 Гвинтівка

Урон	15
Магазин	30
Скорострільність	5 пулі/сек
Режим стрільби	Автоматичний



Рисунок В.15 – Сторінка 15 дизайн-документу

3. Референси

3.1 Гравці

Кожен гравець матиме свій унікальний колір підсвітки.



Рисунок В.16 – Сторінка 16 дизайн-документу

4. Інтерфейс

4.1 Основний геймплей

На екрані у гравця відображається інформація про стан його здоров'я (у верхньому лівому куті червоний статус бар), стан балонів з киснем (у верхньому лівому куті синій статус бар), стан здоров'я та балонів з киснем команди (у верхньому правому куті), залишок заряду його поточного предмета в руці предмету, якщо він має заряди (внизу ліворуч), обраний предмет (внизу праворуч), індикатор шляху, який корабль подала, і який йому ще треба подолати, для успішного проходження рівня (зелений статус бар зверху).



Рисунок В.17 – Сторінка 17 дизайн-документу

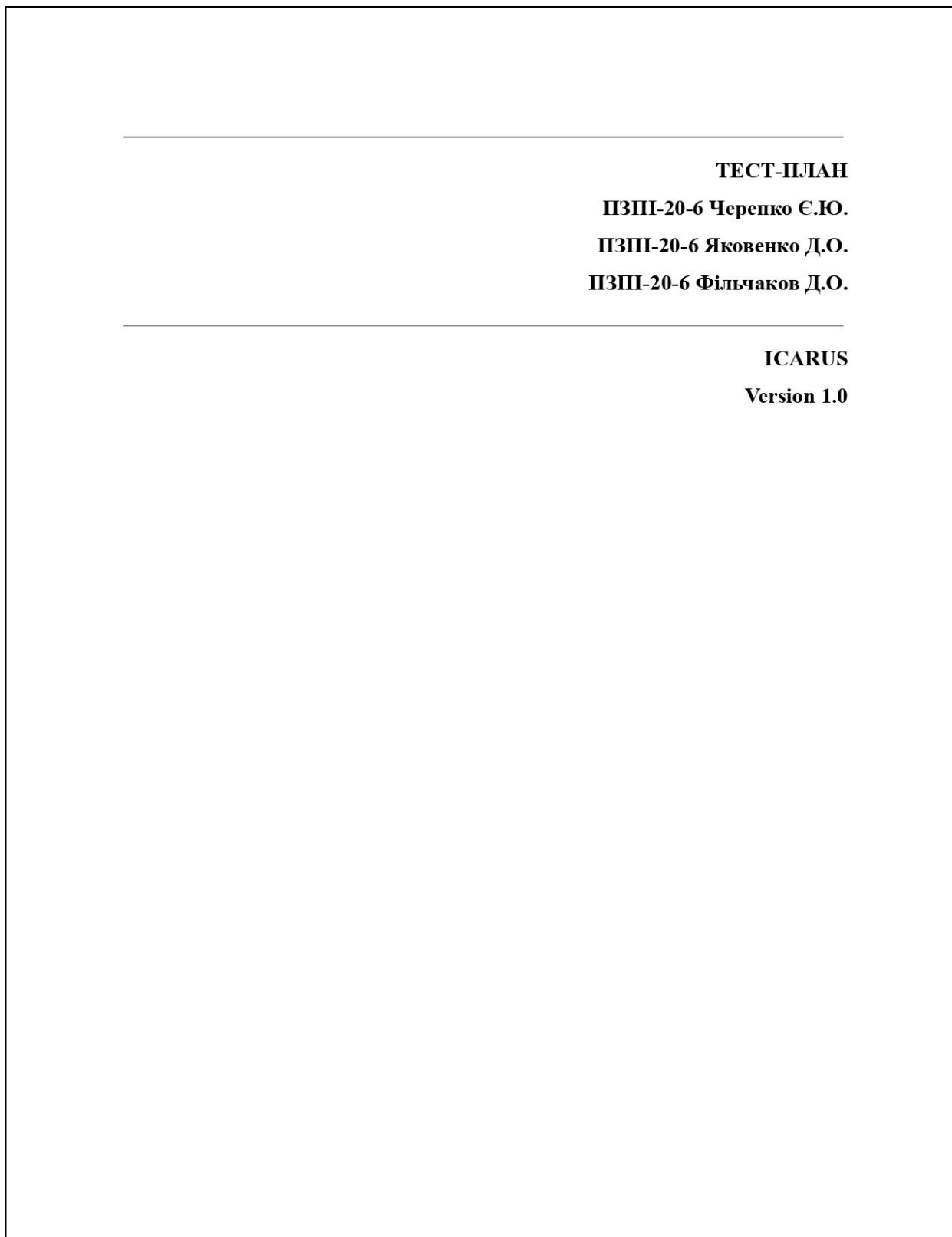
ДОДАТОК Г**Тест-план**

Рисунок Г.1 – Сторінка 1 тест-плану

Revision History

Date	Version	Description	Author
26/02/2024	1.0	Initial revision	Черепко Є.Ю. Яковенко Д.О. Фільчаков Д.О.

Рисунок Г.2 – Сторінка 2 тест-плану

Table of Contents

1. Вступ (Introduction).....	4
1.1 Мета (Purpose).....	4
1.2 Довідкова інформація (Background)	4
1.3 Галузь застосування (Scope).....	5
1.4 Визначення проекту (Project Identification).....	8
2. Вимоги до тестування (Requirements for Test)	9
3. Стратегія тестування (Test Strategy)	9
3.1 Типи тестування (Testing Types)	10
3.1.1 Дані і БД Інтеграційне тестування (Data and Database Integrity Testing).....	10
3.1.2 Функціоанльне тестування (Function Testing)	10
3.1.3 Бізнес-цикл тестування (Business Cycle Testing).....	11
3.1.4 Тестування інтерфейсу користувача (User Interface Testing).....	11
3.1.5 Тестування продуктивності (Performance Profiling)	12
3.1.6 Завантажувальне тестування (Load Testing).....	12
3.1.7 Стресове тестування (Stress Testing)	13
3.1.8 Навантажувальне тестування (Volume Testing)	13
3.1.9 Тестування безпеки і контролю доступу (Security and Access Control Testing).....	13
3.1.10 Тестування відмовостійкості та відновлення (Failover and Recovery Testing).....	13
3.1.11 Тестування конфігурації (Configuration Testing).....	14
3.1.12 Тестування інсталяції (Installation Testing).....	16
3.2 Інструменти (Tools)	16
4. Ресурси (Resources)	16
4.1 Ролі (Roles).....	16
4.2 Система (System)	19
Задачі проекту (Appendix A Project Tasks)	20

Рисунок Г.3 – Сторінка 3 тест-плану

Test Plan

1. Вступ (Introduction)

1.1 Мета (Purpose)

Метою складання Тест Плану є опис процесу тестування комп'ютерної гри Icarus, що є нашою дипломною роботою. Розглянемо визначення існуючої інформації про гру, визначення рекомендованих вимог до тестування, опис стратегій тестування, рекомендацій щодо використання, визначення необхідних ресурсів і забезпечення оцінки випробувань, перелік тестових елементів гри.

1.2 Довідкова інформація (Background)

У якості об'єкта тестування було обрано гра Icarus, що є нашою дипломною роботою.

В ході тестування необхідно буде перевірити найважливіші складові гри . А саме:

1. Графічний інтерфейс;
 2. Анімації персонажів;
 3. Взаємодія персонажа з оточенням;
 4. Ворогів;
 5. Геймплей у різних режимах(одиночний\split-screen\multiplayer);
- Саме ці функції є основою онлайн гри, що тестується.

Тестування гри буде виконано в ручному режимі. Якщо в результаті тестування буде знайдено 75 багів, то тест можна вважати вдалим.

1.3 Галузь застосування (Scope)

Повний набір складових частин, що будуть протестовані:

1. Графічний інтерфейс:
 - Головне меню;
 - Меню налаштувань;
 - Меню створення сесії;
 - Меню обору ролі в split-screen;
 - Меню обору ролі в мультиплеєрі;
 - Магазин;
 - Екран завантаження гри;
 - HUD персонажа;
 - Меню паузи.

2. Анімації персонажів:
 - Анімація пересування персонажа;
 - Анімації персонажа з різними предметами;
 - Анімації персонажа при використанні споживаних предметів;
 - Анімація пересування ворогів;
 - Анімація атак ворогів;

3. Мультиплеер:
 - Авторизація;
 - Створення сесії;
 - Під'єднання до сесії;
 - Реплікація пересування;
 - Реплікація станів персонажів;
 - Реплікація модулів;

- Реплікація ворогів;
 - Реплікація предметів;
 - Реплікація магазину;
 - Реплікація інтерфейсу в лобі;
 - Реплікація полумок;
4. Штучний інтелект ворогів:
- перевірка реакції ворогів на персонажів;
 - перевірка реакції ворогів на модулі;
 - перевірка коректності пошуку ворогом цілі;
 - коректність пересування ворогів по навігаційній сітці;
 - перевірка узгодженості атак ворогів між собою;
5. Магазин:
- коректність списання ігрової валюти та начислення предметів, або ресурсу;
 - коректність оновлення наявних предметів з переходом на новий рівень;
 - коректність відродження загіблених персонажів при їх викупі;
6. VFX
- Коректність відображення частиць.
 - Оптимізованість візуальних ефектів.
 - Коректність взаємодії частиць з навколишнім середовищем.
7. Баланс
- перевірка балансу характеристик персонажів;
 - перевірка балансу характеристик ворогів;

Рисунок Г.6 – Сторінка 6 тест-плану

- перевірка балансу систему токенів атаки;
- перевірка балансу атак ворогів: швидкості, частоти, сили особливих ефектів;
- перевірка балансу характеристик модулів;
- перевірка збалансованості алгоритму поломок;
- перевірка балансу зброї;
- перевірка балансу предметів починки;
- перевірка балансу споживаних предметів;
- перевірка балансу пожеж;
- перевірка балансу витрачання кисню;

8. Геймплей у різних режимах:

- Вибір класу персонажа в split-screen;
- Вибір класу персонажа в мультиплеері;
- Геймплей в однокористувацькому режимі;
- Геймплей в split-screen;
- Геймплей в мультиплеері;
- Спавн персонажей;
- Винекнення поломок;
- Спавн ворогів;
- Респавн;
- Виграш та програш;

Ризиком, або непередбачуваною обставиною, що може вплинути на тестування може бути відмова роботи Epic Online Services з технічних причин або технічні роботи на сервері.

1.4 Визначення проекту (Project Identification)

Документ і версія / дата (Document and version / date)	Створено або Доступно (Created or Available)	Поступил о або перегляну то (Received or Reviewed)
Специфікація вимог (Requirements Specification)	Так	Ні
Функціональна специфікація (Functional Specification)	Так	Так
Звіти використання - випадок (Use-Case Reports)	Ні	Ні
План проекту (Project Plan)	Так	Ні
Специфікація дизайну (Design Specifications)	Так	Так
Прототип (Prototype)	Так	Ні
Керівництво користувача (User's Manuals)	Ні	Ні
Бізнес модель (Business Model or Flow)	Ні	Ні
Модель даних (Data Model or Flow)	Ні	Ні
Бізнес-функції (Business Functions and Rules)	Так	Ні

Рисунок Г.8 – Сторінка 8 тест-плану

Оцінка ризиків (Project or Business Risk Assessment)	Ні	Ні
--	----	----

2. Вимоги до тестування (Requirements for Test)

В ході тестування необхідно буде перевірити найважливіші складові гри

. А саме:

1. Графічний інтерфейс;
2. Анімації персонажів;
3. Взаємодія персонажа з оточенням;
4. Ворогів;
5. Геймплей у різних режимах(одиночний\split-screen\multiplayer);

Саме ці функції є основою онлайн гри, що тестується.

Тестування гри буде виконано в ручному режимі. Якщо в результаті тестування буде знайдено 75 багів, то тест можна вважати вдалим.

3. Стратегія тестування (Test Strategy)

Для тестування було прийнято рішення використати наступні типи:

- Функціональне тестування;
- Тестування інтерфейсу користувача;
- Тестування продуктивності;
- Тестування відмовостійкості;
- Тестування конфігурації;

Інтеграційне тестування, бізнес-цикл тестування, стресове тестування, тестування безпеки і контролю доступу, тестування навантаження, завантажувальне тестування та тестування інсталяції не будуть проведені з

ряду причин, що будуть наведені нижче у відповідних пунктах.

3.1 Типи тестування (Testing Types)

3.1.1 Дані і БД Інтеграційне тестування (Data and Database Integrity Testing)

Цей тип тестування неможливо провести через відсутність БД. Від тесту відмовляємося, через непотрібність проведення.

3.1.2 Функціональне тестування (Function Testing)

Було прийнято рішення обрати цей тип тестування з метою перевірки правильності прийняття і обробки даних, а також належного виконання бізнес-правил. Тестування буде проведено за методом «білої скрині», тобто перевірки гри, занурюючись у код, щоб зрозуміти яка частина коду працює некоректно.

Мета випробування (Test Objective):	Виявити баги
Технічний прийом (Technique):	Методом «білої скрині» перевірити кожну функцію, виявити дефекти.
Критерії завершення (Completion Criteria):	Кожна складова гри протестована, виявлені помилки записані у баг-репорт
Спеціальні рекомендації (Special Considerations):	Зафіксувати процес тестування на відео

Рисунок Г.10 – Сторінка 10 тест-плану

3.1.3 Бізнес-цикл тестування (*Business Cycle Testing*)

Бізнес цикл тестування не буде використовуватися, оскільки ми не маємо необхідної документації. Цей тип тестування потребує документації, де визначені бізнес-процеси, що представляють собою сценарії щоденного використання. В нашому випадку такої документації немає у наявності. Від цього тестування відмовляємося.

3.1.4 Тестування інтерфейсу користувача (*User Interface Testing*)

Тестування інтерфейсу користувача було обрано з метою перевірки взаємодії користувача з інтерфейсом гри. В ході тестування необхідно перевірити зручність та правильність роботи інтерфейсу, а також виявити візуальні баги та баги анімацій.

Мета випробування (Test Objective):	Перевірити всі доступні користувачеві сторінки, діалогові вікна та елементи ігрового інтерфейсу під час геймплею.
Технічний прийом (Technique):	Створення та редагування випробувань для кожного елемента, щоб перевірити коректність його роботи та наявність візуальних помилок.
Критерії завершення (Completion Criteria):	Кожен елемент інтерфейсу був перевірений, виявлені візуальні баги занесені у баг-репорт.

Рисунок Г.11 – Сторінка 11 тест-плану

▲ 3.1.5 Тестування продуктивності (*Performance Profiling*)

Тестування продуктивності було обрано з метою виміру оцінки часу відгуку, перевірки швидкості транзакції, й інших вимог. Тестування продуктивності реалізується і виконується згідно з профілем, а також розглядається залежно від робочого навантаження або апаратної конфігурації.



Мета випробування (Test Objective):	Перевірка продуктивності різних функцій та час їх відгуку, порівняння з нормою.
Технічний прийом (Technique):	- В ході тестування проводити додаткові перевірки на наявність проблем з продуктивністю та багів, що викликають «лаги»; - Тестування проводити на конфігурації, що відповідає рекомендованим системним потребам, або краще.
Критерії завершення (Completion Criteria):	В ході перевірки всі показники продуктивності знаходяться в межах норми, під час тестування не виникає збоїв.



3.1.6 Завантажувальне тестування (*Load Testing*)

Завантажувальне тестування не буде проведено, через неможливість залучити до тестування максимально можливу кількість користувачів. Цей вид тестування передбачає перевірку роботи гри при великій кількості користувачів.

3.1.7 *Стресо́ве тестування (Stress Testing)*

Тестування неможливо провести через відсутності виділених серверів, оскільки вся мережева взаємодія відбувається через сервери, які хостят гравці, і стоїть обмеження на 4 гравців на один сервер.

3.1.8 *Навантажувальне тестування (Volume Testing)*

Навантажувальне тестування неможливо для проведення, оскільки гра не передбачає поля вводу де, є можливість ввести велику кількість даних для тестування. Крім того цей вид тестування не є обов'язковим для функцій, що необхідно перевірити.

3.1.9 *Тестування безпеки і контролю доступу (Security and Access Control Testing)*

Даний тип тестування неможливо провести через відсутність різних ролей у грі. Існує лише роль звичайного гравця.

3.1.10 *Тестування відмовостійкості та відновлення (Failover and Recovery Testing)*

Було прийнято рішення обрати цей тип тестування з метою перевірки

відмовостійкості гри, при втрачанні з'єднання з сервером, або з клієнтом.

Мета випробування (Test Objective):	Перевірити всі етапи гри, на яких можуть виникнути проблеми з мережею.
Технічний прийом (Technique):	Створення та редагування випробувань для кожного етапу, щоб перевірити коректність його роботи.
Критерії завершення (Completion Criteria):	Кожен випадок був перевірений, виявлено баги, та занесено їх в баг-репорт.

3.1.11 Тестування конфігурації (*Configuration Testing*)

Конфігурація тестування перевіряє через тест роботу гри при різних умовах апаратної конфігурації. Цей тип тестування буде проведений з метою перевірки коректності роботи гри на мінімальній та рекомендованій конфігурації.

Мета випробування (Test Objective):	Тестування проводиться належним чином на мінімальній та рекомендованій конфігурації.
Технічний прийом (Technique):	Проведення тестових випадків на різних конфігураціях та порівняння результатів виконання однакових функцій. Виявлення відмінностей у результатах, що будуть вважатися багами.
Критерії завершення (Completion Criteria):	Робота гри на різних конфігураціях перевірена, виявлені баги занесені в баг-репорт.
Спеціальні рекомендації (Special Considerations):	Для тестування використати наближену до Мінімальна: <ul style="list-style-type: none"> - Процесор: amd ryzen 5 1600; - Відеокарта: GTX 1650; - Озу: 8гб; А також конфігурацію, що буде вище за рекомендована: <ul style="list-style-type: none"> - Процесор: amd ryzen 5 3600; - Відеокарта: GTX 2070 - Озу: 16гб;

Рисунок Г.15 – Сторінка 15 тест-плану

3.1.12 Тестування інсталяції (*Installation Testing*)

Тестування інсталяції не потребується для тестування зазначених вище функцій, крім того інсталяція відноситься до тестування цифрового магазину, що не розглядається в тест плані.

3.2 Інструменти (Tools)

The following tools will be employed for this project:

	Tool
Test Management	Microsoft Excel
Defect Tracking	Jira
ASQ Tool for performance testing	FPS Monitor
Створення тест кейсів (Test case creation)	Microsoft Excel
Відстеження тест кейсів (Test case tracking)	Microsoft Excel

4. Ресурси (Resources)

4.1 Ролі (Roles)

Ця таблиця показує, кадрові забезпечення для проекту.

Людський ресурс (Human Resources)		
Працівник (Worker)	Конкретні обов'язки або Коментарі (Specific Responsibilities or Comments)	Конкретні обов'язки або Коментарі (Specific Responsibilities or Comments)
Тест-менеджер, Менеджер з тестування проекту (Test Manager, Test Project Manager)	1	Забезпечує управління наглядом. Обов'язки: - технічна підтримка, - придбання відповідних ресурсів, - забезпечення управлінської звітності

Рисунок Г.17 – Сторінка 17 тест-плану

Конструктор тестів (Test Designer)	1	<p>Визначення, пріоритетів, і реалізація тестів.</p> <p>Обов'язки:</p> <ul style="list-style-type: none"> - створення плану тестування, - генерація тестових моделей, - оцінка ефективності тестових зусиль.
Тестувальник (Tester)	1	<p>Виконання тестів.</p> <p>Обов'язки:</p> <ul style="list-style-type: none"> - виконання тестів, - журнал результатів, - відновлення в журналі реєстрації після помилок, - документ зміни.
Виконавець (Implementer)	1	<p>Реалізує модульні тести і тестові класи</p> <p>Обов'язки:</p> <ul style="list-style-type: none"> - створює тестові класи і пакети, - виконує тестові моделі

Рисунок Г.18 – Сторінка 18 тест-плану

4.2 Система (System)

Нижче в таблиці представлені системні ресурси для тестування гри.

System Resources	
Resource	Name / Type
Апаратна конфігурація	Мінімальна: - Процесор: amd ryzen 5 1600; - Відеокарта: GTX 1650; - Озу: 8гб;
	Рекомендована: - Процесор: amd ryzen 5 3600; - Відеокарта: GTX 2070 - Озу: 16гб;

Рисунок Г.19 – Сторінка 19 тест-плану

Задачі проекту (Appendix A Project Tasks)

Нижче наведені завдання, пов'язані з тестом:

1. Спланувати тест
 - визначити вимоги до тесту;
 - розробити стратегію тестування;
 - створюємо розклад;
 - генеруємо план тестування.
2. Проектування тесту
 - визначаємо та описуємо тестові кейси;
3. Реалізація тесту
 - записувати або програмувати тестові скрипти;
 - визначити специфічну для тестів функціональність в моделі проектування та реалізації;
 - створення зовнішніх наборів даних.
4. Виконання тесту
 - виконати тестові процедури;
 - реєстрація дефектів.
5. Оцінити тест
 - аналіз дефектів;
 - визначити, чи були досягнуті критерії завершення тесту та критерії успіху.