

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»**

Спеціальність: 121 «Інженерія програмного забезпечення»

Освітньо-професійна програма: «Розробка програмного забезпечення»

Група: 4РП-08

Дипломний проект

**здобувача освіти денної форми навчання
РП.08.11.000.ДП**

***ЖМАЙЛО
АННИ АНАТОЛІЇВНИ***

**м. Одеса
2025 р.**

Спеціальність: 121 «Інженерія програмного забезпечення»

Освітньо-професійна програма: «Розробка програмного забезпечення»

Група: 4РП-08

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проекту на тему:

Розробка мобільної гри з монетизацією через інтеграцію реклами

Проектний матеріал складається з пояснювальної записки на 75 сторінках та графічного (презентаційного) матеріалу на 13 аркушах (слайдах)

Дипломник  (Жмайло А.А.)

Керівник  (Кунуп Т.В.)

Консультанти:


з економічного розділу  (Канський М.Ю)

з розділу охорони праці та техніки безпеки  (Чорновол Н.І.)

з нормоконтролю  (Петрашова В.І.)

старший консультант  (Кривченко Ю.В.)

До захисту допущений

Голова циклової комісії  (Кривченко Ю.В.)

Завідувач відділенням  (Краснокутська К.Г.)

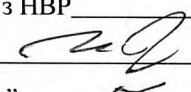
Захист «26» 06 2025 р. Протокол ЕК № 2

Оцінка ЕК 5/90

Секретар ЕК 

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Відділення комп'ютерних систем Комісія КТ та ПІ
Спеціальність 121 Інженерія програмного забезпечення
Освітньо-професійна програма Розробка програмного забезпечення

ЗАТВЕРДЖУЮ:
Заст. дир. з НВР Беркань І.В.

" 12 " 08 2025 р.

ЗАВДАННЯ

на дипломний проект

Здобувачеві освіти Жмайло Анні Анатоліївни
(прізвище, ім'я, по батькові)

1. Тема проекту Розробка мобільної гри з монетизацією через інтеграцію реклами

затверджена наказом по коледжу від " 14 " листопада 2025р. № 246

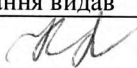
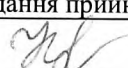
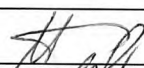
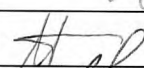
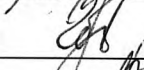

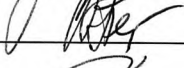



3. Вихідні дані до проекту Розробка мобільної гри з використанням рушія Unity;
Програмування гри мовою C# із застосуванням бібліотек та компонентів Unity;
Створення основних механік для жанру idle-ферми; Реалізація системи вирощування та збору врожаю; Реалізація логіки появи та поведінки NPC-покупців; Створення анімацій для персонажа-гравця; Реалізація системи продажу продукції та відкриття нових грядок; Додавання користувацького інтерфейсу; Впровадження механізмів монетизації через інтеграцію рекламного контенту.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які необхідно розробити)
Розробка ігрової концепції; Вибір жанру idle-ферми; Вибір рушія Unity та мови C#;
Розробка механіки росту і збору овочів; Реалізація NPC-покупців; Реалізація системи продажу та розблокування грядок; Додавання UI для гравця; Налаштування анімацій персонажів; Реалізація інтеграції реклами; Тестування ігрового процесу.

5. Перелік графічного (презентаційного) матеріалу (з точним зазначенням обов'язкових креслень, кількості слайдів)

Загальна ідея гри; Визначення жанру idle-ферми; Вибір інструментів та технологій;
Розробка механіки вирощування овочів; Впровадження системи збору та продажу;
Створення персонажів та покупців; Анімація руху гравця і NPC; Побудова ігрової сцени та грядок; Дизайн інтерфейсу користувача; Демонстрація ігрового процесу.

6. Консультанти по проекту, із зазначенням розділів проекту, що їх стосується

| Розділ | Консультант | Підпис, дата | |
|----------------------|----------------|--|---|
| | | Завдання видав | Завдання прийняв |
| Основний розділ | Кунуп Т.В. |  |  |
| Економічний розділ | Канський М.Ю. |  |  |
| Розділ охорони праці | Чорновол Н.І. |  |  |
| Нормоконтроль | Петрашова В.І. |  |  |
| Старший консультант | Кривченко Ю.В. |  |  |

7. Дата видачі завдання _____

Керівник

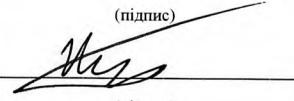
Кунуп Т.В.



(підпис)

Завдання прийняв до виконання

Жмайло А.А.



(підпис)

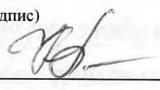
КАЛЕНДАРНИЙ ПЛАН

| № з/р | Назва етапів дипломного проекту | Термін виконання етапів дипломного проекту (роботи) | Відмітка про виконання |
|-------|--|---|------------------------|
| 1 | Вступ. Постановка мети та задач проектування | 20.05.2025 | виконано |
| 2 | Аналіз ігрового процесу розв'язання японських кросвордів | 23.05.2025 | виконано |
| 3 | Аналітичний огляд та вибір програмного рушія | 25.05.2025 | виконано |
| 4 | Формування концепту ігрового процесу гри | 28.05.2025 | виконано |
| 5 | Проектування основних елементів ігрового процесу | 30.05.2025 | виконано |
| 6 | Проектування графічної частини гри | 01.06.2025 | виконано |
| 7 | Реалізація графічної частини гри | 03.06.2025 | виконано |
| 8 | Реалізація основних елементів ігрового процесу | 05.06.2025 | виконано |
| 9 | Імплементація та відлагодження ігрових елементів | 07.06.2025 | виконано |
| 10 | Тестування працездатності елементів гри | 09.06.2025 | виконано |
| 11 | Виправлення виявлених помилок | 10.06.2025 | виконано |
| 12 | Аналіз результатів, підготовка слайдів презентації | 11.06.2025 | виконано |
| 13 | Економічні розрахунки та питання з охорони праці | 12.06.2025 | виконано |
| 14 | Підготовка графічної частини проекту | 13.06.2025 | виконано |
| 15 | Підготовка проекту до захисту та тестування ПП | 14.06.2025 | виконано |

Дипломник


(підпис)

Керівник


(підпис)

ЗМІСТ

| | |
|--|----|
| Вступ..... | 6 |
| 1. Основна частина..... | 7 |
| 1.1 Актуальність теми та мета розробки гри..... | 7 |
| 1.2 Аналіз аналогів та ринку мобільних ігор..... | 9 |
| 1.3 Аналіз цільової аудиторії гри..... | 12 |
| 1.4 Постановка задачі..... | 15 |
| 1.5Проектування архітектури гри..... | 16 |
| 1.6 Реалізація ключових механік гри..... | 25 |
| 1.7 Внутрішня економіка гри..... | 32 |
| 1.8 Система монетизації через відеорекламу..... | 35 |
| 1.9 Технології, що використовуються у грі..... | 40 |
| 1.10 Тестування, результати та ігровий процес..... | 42 |
| 1.11 Майбутні доопрацювання та перспективи розвитку..... | 47 |
| 2. Економічний розділ..... | 50 |
| 2.1 Резюме..... | 50 |
| 2.2 Визначення трудомісткості розробки програмного забезпечення..... | 51 |
| 2.3 Розрахунок ціни програмного продукту..... | 53 |
| 3. Розділ охорони праці та техніки безпеки..... | 55 |
| 3.1 Потенційні шкідливі фактори при роботі над дипломним проектом..... | 55 |
| 3.2 Технічні та організаційні заходи безпеки..... | 56 |
| 3.3 Облаштування безпечного робочого місця користувача ПК..... | 57 |
| 3.4 Основи пожежної безпеки при роботі з комп'ютерною технікою..... | 58 |
| Висновки,,,,,,..... | 60 |
| Перелік використаних джерел..... | 61 |
| ДОДАТОК А Фрагменти програмного коду ігрових механік..... | 62 |
| ДОДАТОК В Слайди мультимедійної презентації..... | 70 |

ВСТУП

Сучасні мобільні ігри є однією з найпопулярніших форм цифрових розваг. Завдяки доступності смартфонів та простоті встановлення, користувачі все частіше обирають саме мобільний формат — зручний, швидкий і зрозумілий. Ігри цього типу не потребують багато часу чи спеціальних знань, тому стають особливо привабливими для широкої аудиторії.

Одна з головних переваг мобільних ігор — можливість грати безкоштовно. Проте розробникам усе одно потрібно отримувати прибуток, тому важливу роль відіграє вбудована система монетизації. Найпоширенішим варіантом є реклама, яку гравець переглядає у зручний для себе момент. Таким чином, гра залишається безкоштовною, але приносить дохід через взаємодію з рекламним контентом.

У межах таких моделей особливо добре працює формат idle-ігор. Це ігри, в яких частина процесів відбувається автоматично — навіть коли гравець не взаємодіє з додатком. Простота механік, приємний візуальний стиль і поступовий прогрес роблять idle-жанр привабливим як для новачків, так і для досвідчених користувачів.

У цьому проєкті реалізовано саме idle-гру. Гравець вирощує овочі на грядках, збирає врожай, продає продукцію, отримує віртуальні гроші й може переглянути рекламу, щоб подвоїти прибуток. Усі дії виконуються у комфортному темпі, що дозволяє грати у будь-який момент — без тиску та без обмежень.

Розробка гри велась у середовищі Unity з використанням мови програмування C#. У проєкті створено власну систему обліку ресурсів, простий інвентар, механіку замовлень від неігрових персонажів (NPC), таймер реклами, а також окрему систему інтерфейсу для взаємодії з гравцем.

Такий підхід дозволяє поєднати базові елементи геймдизайну, програмування, логіки монетизації та UI/UX у межах одного невеликого мобільного продукту. Це не лише корисна навчальна вправа, але й повноцінна технічна реалізація, яка може стати основою для більш складного застосунку або бути опублікованою у магазині мобільних ігор.

| | | | | | | |
|-----|------|----------|------|--------|--------------------------------|------|
| | | | | | <i>РП 08. 11 000. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Дата | Підпис | | 6 |

1. ОСНОВНА ЧАСТИНА

1.1 Актуальність теми та мета розробки гри

У цифрову епоху люди все частіше звертаються до мобільних ігор як до простого та доступного способу відпочити. Замість довгих сесій за комп'ютером багато користувачів обирають швидкий запуск гри прямо з кишені — на смартфоні. Те, що раніше вважалося звичайною розвагою, тепер стало невід'ємною частиною повсякденного життя. Ігри запускаються в транспорті, під час обіду чи ввечері, щоб просто розвантажити голову. Цей новий стиль взаємодії з додатками сформував попит на мобільні ігри з легким, зрозумілим та невимушеним геймплеєм.

Серед найпопулярніших жанрів мобільних ігор особливо вирізняються так звані idle-ігри. Їхня ключова особливість — автоматизований процес, який не вимагає постійної участі гравця. Гравець запускає гру, збирає результат, покращує свої ресурси, а потім може відкласти смартфон, поки процес триває сам. Саме ця простота й ненав'язливість зробили idle-жанр популярним серед людей, які хочуть грати у своєму ритмі.

Перші ігри з подібною логікою почали з'являтися ще в 2000-х роках, але справжній сплеск популярності трапився після релізу *Cookie Clicker* у 2013 році. Незважаючи на елементарний візуальний стиль, гра стала вірусною. Гравців приваблювало не лише натискання на “печенько”, а й те, як просто гра дозволяє відчувати прогрес без зайвих зусиль. З того часу жанр стрімко розвивається, особливо на мобільних платформах, де такі ігри ідеально підходять до формату “5 хвилин вільного часу”.

У 2025 році жанр idle-ігор знову підтвердив свою актуальність. За статистикою, загальна кількість встановлень ігор цього типу перевищила 10 мільярдів завантажень у світі, що більше, ніж у 2023 році (близько 8,7 мільярдів). Також спостерігається зростання прибутку: згідно з оцінками аналітиків, гравці

| | | | | | | |
|-----|------|----------|------|--------|-------------------------|------|
| | | | | | РП 08. 11 001. 00 ДП ПЗ | Арк. |
| Зм. | Арк. | № докум. | Дата | Підпис | | 7 |

витратили на ігри жанру idle понад 1,3 мільярда доларів лише за один рік. Це свідчить не тільки про популярність, а й про ефективну монетизацію, навіть без обов'язкових покупок чи платного доступу.

Важливою перевагою idle-ігор є те, що вони чудово поєднуються з моделлю монетизації через рекламу. Це означає, що гра лишається безкоштовною, а дохід розробника формується через перегляд гравцем відео за бонуси — наприклад, подвоєння прибутку або прискорення процесу. У сучасних умовах це оптимальне рішення, оскільки більшість гравців не готові платити за гру одразу, але охоче взаємодіють із рекламою, якщо це дає вигоду в ігровому процесі.

У контексті розробки саме idle-гри з темою фермерства рішення було продиктоване кількома факторами. По-перше, тема фермерства є інтуїтивно зрозумілою та знайомою гравцю. Навіть без глибоких пояснень користувач одразу розуміє, що робити: саджати овочі, чекати, збирати, продавати. По-друге, такий геймплей дозволяє поступово додавати нові механіки: замовлення від NPC, система інвентаря, монети, бонуси та багато іншого. По-третє, саме у фермерському сетингу логічно виглядає механіка подвоєння прибутку через рекламу — бо продаж і прибуток вже є частиною ігрової логіки.

З урахуванням усього вищесказаного, метою цього проекту стало створення мобільної гри у жанрі idle-симулятора з фермерським ухилом. У грі реалізовано вирощування овочів, збір урожаю, взаємодію з неігровими персонажами, продаж продукції та перегляд реклами як спосіб збільшити дохід. Ідея полягала не лише у створенні цікавого геймплею, але й показати, як працює внутрішня економіка, коли гравець добровільно взаємодіє з рекламою, отримуючи бонуси.

Уся логіка реалізована за допомогою рушія Unity та мови програмування C#. Сценарій гри побудовано навколо поступового розвитку: гравець починає з однієї грядки і кількох овочів, але поступово відкриває нові види продукції, вдосконалює ферму і отримує все більший прибуток. Механіка перегляду реклами працює як додатковий варіант для пришвидшення прогресу, але не є обов'язковою. Це дозволяє залишити вибір за гравцем і не нав'язувати рекламу.

| | | | | | | |
|-----|------|----------|------|--------|--------------------------------|------|
| | | | | | <i>РП 08. 11 001. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Дата | Підпис | | 8 |

Таким чином, idle-ігри не лише не втратили своєї актуальності, а навпаки — показують стабільне зростання популярності. Поєднання зрозумілої механіки, ненав'язливого геймплею та вдало впровадженої реклами дозволяє створити продукт, який відповідає сучасним тенденціям ринку мобільних ігор.

1.2 Аналіз аналогів та ринку мобільних ігор

У сучасному світі мобільні ігри стали не просто способом розваги, а справжнім культурним феноменом. Щодня мільйони користувачів відкривають свої смартфони не лише для спілкування чи перегляду новин, а й для того, щоб хоча б на кілька хвилин зануритися у віртуальний світ — посіяти моркву, зібрати буряк або обмінятися товарами на фермерському ринку. Саме такі прості на перший погляд дії створюють ефект залучення, з яким важко зрівнятися іншим жанрам. Ринок ігор формує особливий мікросвіт, де кожна дія — це не просто натискання кнопки, а частина великої гри.

Попри велику кількість вже існуючих ігор, розробники не припиняють створювати нові продукти, що свідчить про актуальність жанру. Справді, здається, ніби ринок уже заповнений подібними іграми: десятки фермерських симуляторів, сотні idle-проектів, тисячі клонів тих самих механік. Та насправді це лише поверхнєве враження. Більшість таких проектів не мають глибокої ідеї чи продуманої реалізації. Вони зникають так само швидко, як і з'являються. Водночас якісні, унікальні ігри, що враховують потреби гравця та пропонують щось справді нове, стабільно тримаються в топах і приносять прибуток роками.

Щоб глибше зрозуміти, якими мають бути успішні мобільні ігри у жанрі idle/ферма, звернемо увагу на два флагмани цього напрямку — Idle Farming Empire та Hay Day. Саме вони стали для нас джерелом натхнення, з якого ми почерпнули ідеї, а також прикладом того, як правильно працює ігрова економіка, монетизація та інтерфейс.

| | | | | | | |
|-----|------|----------|------|--------|--------------------------------|------|
| | | | | | <i>РП 08. 11 001. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Дата | Підпис | | 9 |

Idle Farming Empire — це приклад гри, яка вдало поєднує простоту та ефективність. У ній гравець керує фермою, що поступово автоматизується, генерує дохід і розвивається без активної участі. Такий формат особливо привабливий для зайнятих людей, які хочуть грати без постійного контролю. Візуальний стиль гри приємний для ока, інтерфейс інтуїтивно зрозумілий, а винагорода за перегляд реклами не нав'язується, а сприймається як приємний бонус.

Hay Day, натомість, орієнтується на інший тип гравців — тих, хто любить глибше занурення у процес. Гравець не лише вирощує рослини, а й займається виробництвом, торгівлею, будівництвом і навіть соціальними взаємодіями. Гра має складнішу структуру, вимагає більше часу, але й дарує значно більше варіативності. Вона візуально яскрава, захоплива і дає відчуття справжнього фермерського життя.

Таблиця 1.1 Порівняння ігор

| Параметр | Idle Farming Empire | Hay Day |
|--------------------|---------------------------------------|--------------------------------------|
| Жанр | Idle-симулятор | Фермерський симулятор |
| Механіка | Автоматичний дохід, мінімальна участь | Вирощування, торгівля, взаємодія |
| Графіка | Мінімалістична, лаконічна | Деталізована, яскрава |
| Цільова аудиторія | Казуальні гравці | Любителі стратегій, креативні гравці |
| Монетизація | Реклама з винагородою | Внутрішньоігрові покупки, реклама |
| Соціальні функції | Відсутні | Торгівля між гравцями, листування |
| Складність | Низька | Середня/висока |
| Рівень залученості | Пасивний | Активний |

Згідно зі звітом Sensor Tower (2025), близько 66% прибутку мобільних ігор у 2024 році було отримано за рахунок реклами з винагородою (Rewarded Ads). Ще 24% забезпечили внутрішньоігрові покупки, 7% — підписки, і лише 3% — банерна реклама. Це ще раз підтверджує, що користувачі готові добровільно взаємодіяти з рекламою, якщо за це отримують реальні переваги у грі. Додатково варто згадати, що загальний прибуток від мобільних ігор у 2024 році склав 81 мільярд доларів, і ця цифра лише зростає.

Структура доходів мобільних ігор у 2024 році

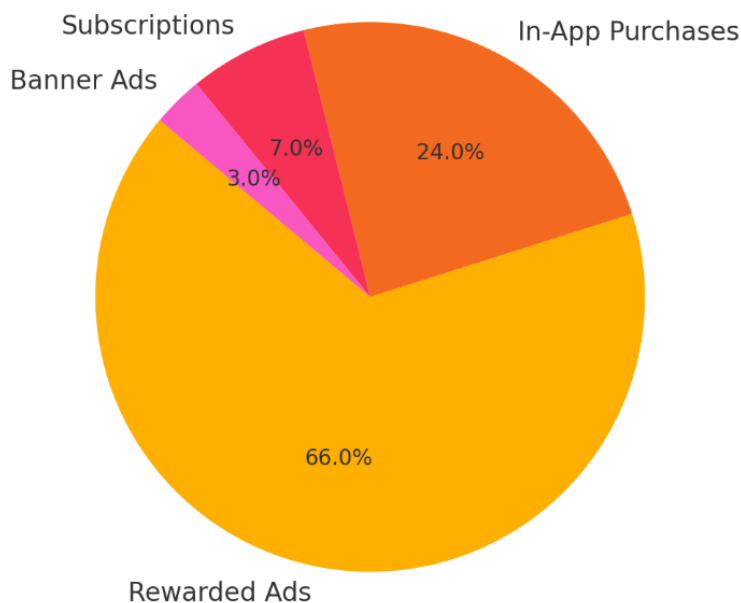


Рисунок 1.1. Графік доходів мобільних ігор

Усе це вказує на те, що ринок мобільних ігор, хоча й заповнений, залишається надзвичайно перспективним. Він вимагає не просто чергової копії існуючого проекту, а гри з унікальним баченням, зручним геймплеєм та чесною моделлю монетизації. Саме тому під час розробки нашої гри ми надихалися кращими прикладами, але водночас прагнули уникнути їхніх недоліків. Ми взяли легкість і доступність Idle Farming Empire, поєднали її з естетикою та глибиною Nau Day, і створили власний продукт — збалансований, сучасний та привабливий для гравця.

1.3 Аналіз цільової аудиторії гри

Створення мобільної гри — це не лише процес кодування чи розробки ігрових механік, а передусім ретельне дослідження поведінки користувачів, їх потреб, мотивацій та емоційного стану. Успішна гра — це та, яка відгукується в душі гравця, дає йому відчуття задоволення, поступу та причетності. Тому перш ніж почати розробку технічної частини, необхідно було чітко визначити, для кого створюється ця гра, які сценарії використання вона повинна охоплювати, і що саме зробить її привабливою та бажаною для запуску знову і знову.

У сучасному цифровому світі мобільна гра повинна бути зручною, короткою, візуально привабливою і водночас достатньо цікавою, щоб залишатися у пам'яті користувача. Найбільш активними гравцями в мобільні ігри є представники вікової групи 16–35 років. Це школярі, студенти, працівники офісів та фрилансери. Вони мають доступ до смартфонів, використовують додатки щодня, але при цьому не завжди мають багато вільного часу або бажання занурюватися у складний геймплей.

Саме тому idle-ігри ідеально підходять під стиль життя такої аудиторії. Основна ідея idle-жанру полягає в тому, що значна частина ігрового процесу відбувається автоматично. Гравець сіє насіння, запускає процес і повертається до гри пізніше, щоб зібрати результати. Це дозволяє створити ефект прогресу навіть без постійної участі, а також викликати відчуття задоволення від досягнення — без стресу чи перевантаження.

Idle-ігри задовольняють ще одну важливу потребу — відчуття контролю та спокою. У ритмі сучасного життя, де все змінюється швидко, ігри цього жанру дозволяють зупинитися, розслабитися та отримати позитивний досвід. Вони не вимагають від гравця реакції в реальному часі, не карають за помилки, і не створюють тиску. Натомість — дарують момент внутрішнього спокою, впевненості у поступі, що відбувається без напруження.

| | | | | | | |
|-----|------|----------|------|--------|--------------------------------|------|
| | | | | | <i>РП 08. 11 001. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Дата | Підпис | | 12 |

Цільова аудиторія нашої гри — це не геймери в класичному сенсі, а звичайні люди, які просто шукають кілька хвилин відпочинку: десь у транспорті, на перерві між парами або після важкого робочого дня. Вони цінують комфорт, простоту, швидкий старт гри без складного навчання та інтуїтивно зрозумілий інтерфейс. Тому всі елементи гри, включаючи меню, кнопки, візуальний стиль та розміщення елементів, були адаптовані під звичайного користувача.

Ще один важливий фактор — фінансова поведінка користувача. Досвід показує, що переважна більшість мобільних гравців не готові платити за ігри одразу. Проте вони охоче взаємодіють із рекламою, якщо це приносить їм вигоду. Саме тому у гри реалізовано механізм rewarded-реклами: гравець сам вирішує, чи переглядати ролик, але якщо погоджується — отримує цінний бонус. Це дозволяє поєднати безкоштовність гри з її монетизацією, не викликаючи роздратування в користувача.

У гри також передбачена логіка накопичення змін під час відсутності гравця. Це формує ще один емоційний тригер: гравець повертається і бачить, що щось відбулося без його участі — врожай виріс, з'явилися нові замовлення, збільшився прибуток. Така «гра на фоні» викликає ефект сюрпризу та позитивного підкріплення, що формує звичку повертатися.

Візуальна складова гри також була адаптована до вподобань цільової аудиторії. Ми обрали теплу кольорову гаму, м'які форми, мінімалістичний інтерфейс без перевантаження деталями. Це дозволяє грі бути приємною для очей, знижує візуальну втому і створює атмосферу затишку — наче гравець дійсно знаходиться в маленькому селі або на власній ділянці.

Гендерний баланс також враховано. За статистикою, idle-ігри мають майже рівний розподіл між чоловіками та жінками. Тому персонажі, елементи інтерфейсу та механіки були розроблені так, щоб бути привабливими для обох груп — без переваги в сторону «бойової» чи «дівочої» стилістики. Це дозволяє розширити потенційну аудиторію гри.

| | | | | | | |
|-----|------|----------|------|--------|--------------------------------|------|
| | | | | | <i>РП 08. 11 001. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Дата | Підпис | | 13 |

Особливу увагу приділено локалізації. Українська мова в інтерфейсі гри є важливим фактором емоційного зв'язку з користувачем, особливо в контексті сучасних подій. Крім того, у грі є унікальні NPC з локальними іменами, поведінкою та замовленнями. Це не лише додає глибини грі, а й створює емоційний контакт із гравцем — адже персонажі не абстрактні, а «живі» та впізнавані.

Окрема категорія — гравці, які взаємодіють з грою епізодично. Їх цікавить механіка, де щось відбувається навіть без їхньої присутності. Саме тому ми реалізували логіку пасивного прогресу: рослини ростуть, замовлення з'являються, баланс змінюється — все це триває навіть тоді, коли додаток не активний. Гравець повертається і бачить: щось змінилося, щось оновилося, він став на крок ближчим до мети. Це викликає позитивну реакцію, цікавість і бажання грати знову. Цей ефект, хоч і здається простим, є дуже сильним емоційним тригером, який формує звичку та прихильність до гри.

Підсумовуючи, основні риси нашої цільової аудиторії:

- від 16 до 35 років;
- цінують простоту, спокій і відсутність тиску;
- грають у транспорті, на перервах або ввечері перед сном;
- не готові витратити гроші, але позитивно ставляться до бонусів за перегляд реклами;
- мають досвід гри в мобільні додатки, але не прагнуть до складного геймплею;
- шукають гру, яка викликає приємну емоцію та створює звичку повертатися.

Ці характеристики вплинули на всі аспекти гри: від графіки та інтерфейсу до ігрової логіки, механік росту, поведінки NPC та впровадження системи реклами. Завдяки цьому ми створили продукт, який не лише працює технічно, а й емоційно «резонує» зі своїм користувачем — а отже, має всі шанси на тривале життя в пам'яті гравця.

| | | | | | | |
|-----|------|----------|------|--------|--------------------------------|------|
| | | | | | <i>РП 08. 11 001. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Дата | Підпис | | 14 |

1.4 Постановка задачі

Перш ніж розпочати розробку будь-якого програмного продукту, необхідно чітко визначити його цілі, призначення, функціональні можливості та вимоги до інтерфейсу й логіки роботи. У контексті мобільної гри у жанрі симулятора з елементами «idle-геймплею» ці аспекти мають особливу вагу — саме від правильного формулювання задач залежить зручність для гравця, стабільність роботи системи, відповідність очікуванням цільової аудиторії та потенціал для монетизації.

Метою проєкту стало створення мобільної гри, яка моделює спокійний, ненапружений процес фермерства. Основна ідея полягає в тому, щоб надати гравцю змогу вирощувати овочі, збирати врожай, продавати продукцію неігровим персонажам (NPC), поступово розвивати своє господарство та за бажанням переглядати рекламу, отримуючи бонуси. Усе це повинно відбуватись у розслабленому ритмі, без тиску на користувача — з можливістю грати епізодично та відчувати прогрес без постійної присутності.

Під час формування ігрової концепції було виділено основні функціональні задачі, які повинна реалізовувати гра:

- реалізувати цикл росту овочів із поділом на стадії, що змінюються з часом та мають візуальне відображення;
- створити логіку збирання врожаю з появою іконок готовності та інтерактивною взаємодією гравця;
- побудувати систему NPC, які рухаються по заданому маршруту, формують замовлення й очікують обслуговування;
- забезпечити механізм формування замовлень залежно від прогресу гравця та наявних розблокованих культур;
- розробити інвентар, що зберігає кількість зібраних овочів і дозволяє відображати їх у UI;

| | | | | | | |
|-----|------|----------|------|--------|--------------------------------|------|
| | | | | | <i>РП 08. 11 001. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Дата | Підпис | | 15 |

- впровадити систему монетизації через перегляд реклами з винагородою (наприклад, подвоєння прибутку);
- налаштувати логіку розблокування нових культур залежно від кількості накопичених коштів;
- створити адаптивний та інтуїтивно зрозумілий інтерфейс, що підходить для різних роздільних здатностей мобільних пристроїв.

Окремим завданням стало забезпечення ігрового досвіду, приємного для широкої аудиторії. Гра має викликати відчуття затишку, спокою та досягнення. З цією метою враховано не лише технічну сторону, а й емоційну складову: доброзичливий візуальний стиль, м'яка кольорова гама, мінімалістичний UI та приємна музика повинні формувати позитивне сприйняття гри.

Також одним із ключових викликів було поєднати механіку поступового прогресу з автономністю: користувач не повинен відчувати необхідності постійно перебувати в грі, однак при поверненні — бачити результати: вирости овочі, з'явилися нові замовлення, збільшився прибуток. Саме такий підхід дозволяє сформувати звичку та емоційний зв'язок із додатком без нав'язливості.

Таким чином, постановка задачі охопила не лише реалізацію ігрових механік, а й створення цілісного продукту, який відповідає сучасним очікуванням мобільної аудиторії — зручного, легкого для освоєння, технічно оптимізованого та водночас привабливого з точки зору візуального й емоційного оформлення.

1.5 Проєктування архітектури гри

Проєктування архітектури гри є основою, на якій будується вся подальша реалізація, тому цей етап відіграє надзвичайно важливу роль. Саме з нього починається життя цифрового продукту — від перших уявлень до конкретних реалізацій у середовищі розробки. Гра, створена в межах цього проєкту, поєднує функціональний мінімалізм, логічну структурованість і візуальну доступність, що відповідає як сучасним технічним вимогам, так і очікуванням аудиторії.

| | | | | | | |
|-----|------|----------|------|--------|--------------------------------|------|
| | | | | | <i>РП 08. 11 001. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Дата | Підпис | | 16 |

Головним завданням під час проєктування архітектури було організувати внутрішню логіку гри так, щоб вона була зручною для допрацювання, розширення, повторного використання компонентів та простого оновлення. У цьому контексті архітектура була спроектована за принципами модульності, ізоляції функцій і мінімізації залежностей між об'єктами.

Загальна структура побудована у середовищі Unity. У межах однієї сцени розміщено кілька основних зон: поле з грядками, стіл для взаємодії з NPC, зона переміщення персонажа та інтерфейсна панель. Такий поділ дозволяє розмежувати логіку гри на окремі блоки, кожен з яких керується власним менеджером і відповідає за певний функціонал. Прикладом є такі об'єкти, як Player, NPCManager, FieldZone, LevelManager тощо.

Основний ігровий персонаж (фермерка) переміщується по сцені за допомогою клавіш. Для реалізації плавного керування використовується векторне переміщення на основі Rigidbody2D. Це забезпечує гнучкий контроль за напрямком руху:

```
private void Update()
{
    float horizontal = Input.GetAxisRaw("Horizontal");
    float vertical = Input.GetAxisRaw("Vertical");
    Vector2 direction = new Vector2(horizontal, vertical).normalized;
    rb.velocity = direction * moveSpeed;
    AnimateMovement(direction);
}
```

Цей підхід дозволяє легко масштабувати керування, додавати анімації або обмеження руху. Метод AnimateMovement відповідає за зміну анімацій залежно від напрямку руху. Усі спрайти руху об'єднані у анімації за допомогою Unity Animator.

| | | | | | | |
|-----|------|----------|------|--------|--------------------------------|------|
| | | | | | РП 08. 11 001. 00 ДП ПЗ | Арк. |
| Зм. | Арк. | № докум. | Дата | Підпис | | 17 |

Грядки реалізовано як окремі об'єкти з прикріпленим скриптом росту. Для кожного овочу є індивідуальний скрипт, наприклад, CarrotGrowth.cs, який задає стадії росту, їх тривалість та іконку стиглості врожаю:

```
public class CarrotGrowth : MonoBehaviour
{
    public GameObject[] growthStages;
    public float growthTime = 15f;
    public GameObject readyIcon;
    private int currentStage = 0;

    void Start()
    {
        StartCoroutine(Grow());
    }

    IEnumerator Grow()
    {
        while (currentStage < growthStages.Length)
        {
            growthStages[currentStage].SetActive(true);
            yield return new WaitForSeconds(growthTime);
            currentStage++;
        }
        readyIcon.SetActive(true);
    }
}
```

Така реалізація дозволяє легко налаштовувати таймінги та етапи росту кожної культури. Кожна стадія росту активується послідовно, а наприкінці з'являється іконка, яка сигналізує, що овоч готовий до збору.

Для збору врожаю використовується окремий клас, який визначає відстань до гравця та дозволяє взаємодію при натисканні клавіші:

```
public class CarrotCrop : MonoBehaviour
{
    public GameObject harvestIcon;
    public float interactDistance = 0.5f;
    private Transform player;

    void Start()
    {
        player = GameObject.FindGameObjectWithTag("Player").transform;
    }

    void Update()
    {
        if (Vector2.Distance(transform.position, player.position) <=
interactDistance)
        {
            if (Input.GetKeyDown(KeyCode.E))
                Harvest();
        }
    }

    void Harvest()
    {
```

| | | | | | | |
|-----|------|----------|------|--------|--------------------------------|------|
| | | | | | <i>РП 08. 11 001. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Дата | Підпис | | 19 |

```

        InventoryManager.Instance.AddVegetable("Морква", 1);
        Destroy(gameObject);
    }
}

```

Цей скрипт забезпечує інтуїтивну логіку: при підході до стиглого овочу і натисканні кнопки, гравець збирає урожай. Всі зібрані овочі передаються до інвентаря.

Інвентар реалізований за допомогою серіалізованого класу CropData, що містить назву, іконку та кількість. Менеджер інвентаря зберігає всі типи овочів і дозволяє оновлювати кількість кожного:

```

[System.Serializable]
public class CropData
{
    public string cropName;
    public Sprite icon;
    public int quantity;
}

public class InventoryManager : MonoBehaviour
{
    public List<CropData> crops;

    public void AddVegetable(string name, int amount)
    {
        foreach (var crop in crops)
        {
            if (crop.cropName == name)

```

| | | | | | | |
|-----|------|----------|------|--------|--------------------------------|------|
| | | | | | РП 08. 11 001. 00 ДП ПЗ | Арк. |
| Зм. | Арк. | № докум. | Дата | Підпис | | 20 |

```

        {
            crop.quantity += amount;
            break;
        }
    }
}
}
}

```

Це забезпечує просту логіку оновлення інтерфейсу. Інвентар відкривається через взаємодію зі скринню у грі, де гравець бачить усі зібрані овочі у вигляді таблиці.

Клієнти (NPC) рухаються по сцені до столу, зупиняються та очікують замовлення. Логіка їх руху реалізована в класі NPC_Move:

```

public class NPC_Move : MonoBehaviour
{
    public Transform stopPoint1, stopPoint2, stopPoint3, stopPoint4;
    private int stage = 0;

    void Update()
    {
        Vector3 target = stage == 0 ? stopPoint1.position :
            stage == 1 ? stopPoint2.position :
            stage == 2 ? stopPoint3.position : stopPoint4.position;

        transform.position = Vector3.MoveTowards(transform.position, target,
speed * Time.deltaTime);

        if (Vector3.Distance(transform.position, target) < 0.1f)

```

| | | | | | | |
|-----|------|----------|------|--------|--------------------------------|------|
| | | | | | РП 08. 11 001. 00 ДП ПЗ | Арк. |
| Зм. | Арк. | № докум. | Дата | Підпис | | 21 |

```

    {
        stage++;
        if (stage == 2)
            ShowOrder();
    }
}

void ShowOrder()
{
    NPCOrder order = OrderGenerator.Instance.GenerateOrder();
    OrderBubbleSpawner.Spawn(order);
}
}

```

NPC зупиняються у заданій точці та запускають генератор замовлення. Це створює динаміку ігрового процесу — кожен клієнт робить унікальний запит.

Для генерації замовлення використовується окремий клас OrderGenerator, який обирає овочі з розблокованих:

```

public class OrderGenerator : MonoBehaviour
{
    public List<CropDefinition> unlockedCrops;

    public NPCOrder GenerateOrder()
    {
        var crop = unlockedCrops[Random.Range(0, unlockedCrops.Count)];
        return new NPCOrder(crop.name, Random.Range(1, 5), crop.price);
    }
}

```

Цей код дозволяє гнучко змінювати набір доступних овочів і адаптувати складність гри відповідно до прогресу гравця.

Всі описані компоненти працюють як окремі модулі, які взаємодіють один з одним через менеджери (наприклад, NPCManager, OrderManager, InventoryManager). Це дозволяє гнучко масштабувати проєкт, додавати нові об'єкти, овочі, механіки або типи NPC без необхідності глобальних змін.

На ілюстрації нижче зображено загальний вигляд сцени Unity, де видно розміщення основних об'єктів гри: грядки з різними культурами, персонаж-гравець, NPC, зона продажу та візуальні елементи інтерфейсу.



Рисунок 1.2. Загальний вид ігрової сцени

Крім цього, нижче представлено умовну схему архітектури логіки гри, що показує зв'язки між основними компонентами:

| | | | | | | |
|-----|------|----------|------|--------|--------------------------------|------|
| | | | | | <i>РП 08. 11 001. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Дата | Підпис | | 23 |

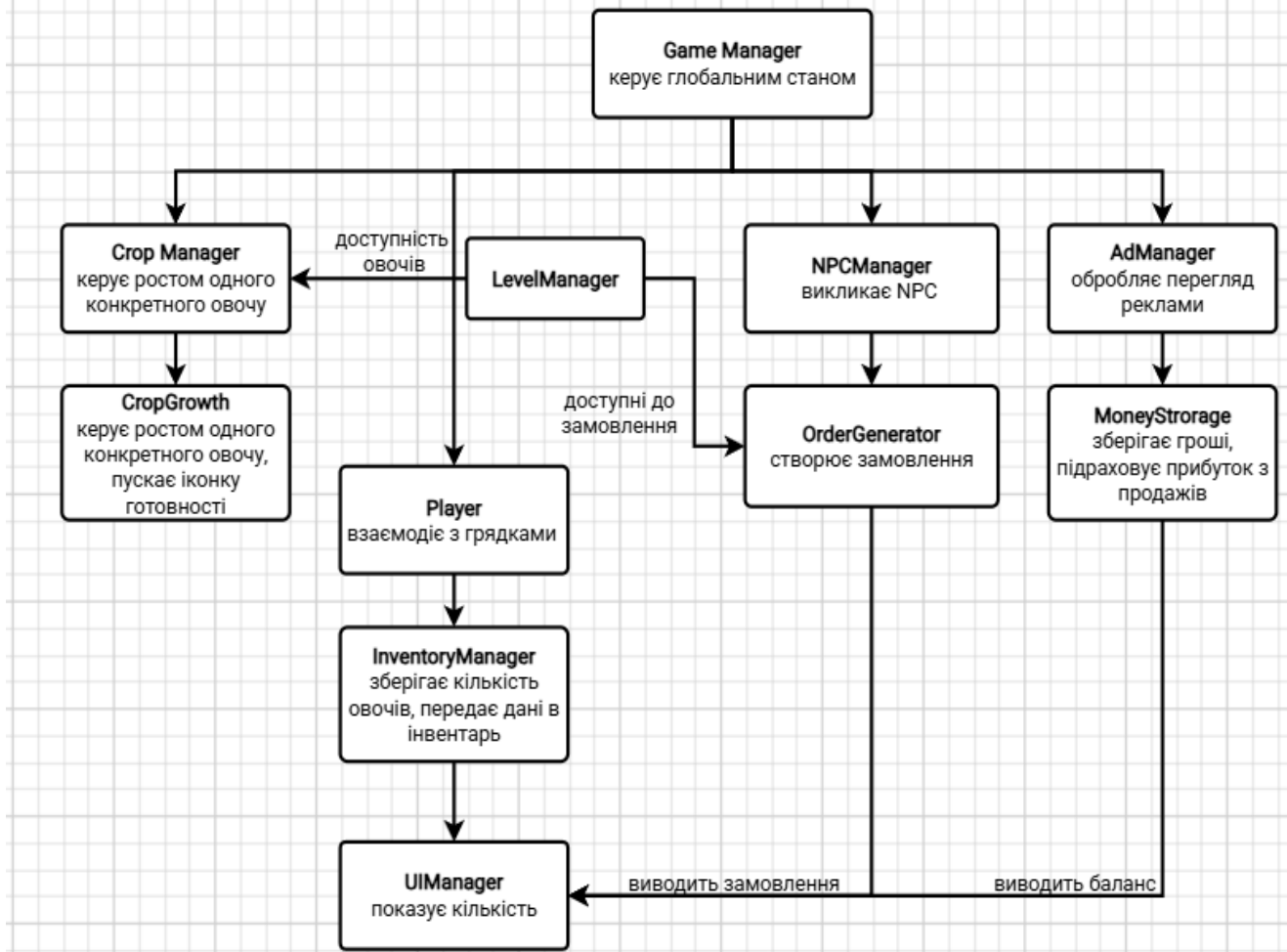


Рисунок 1.3. Схема логіки архітектури гри

На наведеній схемі показано ключові компоненти гри та їх взаємозв'язки. Центральним керуючим елементом виступає GameManager — він координує основні процеси гри. CropManager відповідає за контроль росту овочів, тоді як CropGrowth забезпечує поетапний розвиток конкретної культури. Player взаємодіє з грядками та передає зібраний врожай до InventoryManager, який зберігає кількість овочів та оновлює інтерфейс. NPCManager ініціює появу клієнтів, які формують замовлення через OrderGenerator. Після перегляду реклами AdManager активує бонус, що впливає на збереження коштів у системі MoneyStorage. Таким чином, кожен блок є частиною цілісної системи, де всі компоненти працюють синхронно та автономно, забезпечуючи стабільну логіку геймплею.

1.6 Реалізація ключових механік гри

Реалізація ключових механік гри стала найбільш об'ємним і визначальним етапом усього процесу розробки. Саме завдяки цим механікам формується основна ігрова петля: від вирощування овочів до їх продажу через взаємодію з неігровими персонажами (NPC), що забезпечує гравцю відчуття прогресу та взаємозв'язку між діями. Головною метою було побудувати повноцінний ігровий цикл, який не лише виконує функціональність, а й викликає інтерес та стимулює повторну гру. Весь процес було реалізовано поетапно, з урахуванням як технічних, так і ігрових аспектів.



Рисунок 1.4. Початковий вид сцени

Початковою точкою став механізм росту овочів. Для кожної культури використовується скрипт CropGrowth, який забезпечує контроль над стадіями росту, часом і готовністю врожаю. Масив об'єктів growthStages відповідає за візуальні етапи росту — від проростка до стиглого овочу. Реалізація росту здійснюється через корутину, що дозволяє зупиняти виконання між стадіями без заморожування гри. Нижче представлений приклад коду для буряка:

| | | | | | | |
|-----|------|----------|------|--------|--------------------------------|------|
| | | | | | <i>РП 08. 11 001. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Дата | Підпис | | 25 |

```

public class CropGrowth : MonoBehaviour
{
    public GameObject[] growthStages;
    public float growthTime = 10f;
    public GameObject readyIcon;
    private int currentStage = 0;

    void Start()
    {
        StartCoroutine(Grow());
    }

    IEnumerator Grow()
    {
        while (currentStage < growthStages.Length)
        {
            growthStages[currentStage].SetActive(true);
            yield return new WaitForSeconds(growthTime);
            currentStage++;
        }
        readyIcon.SetActive(true);
    }
}

```

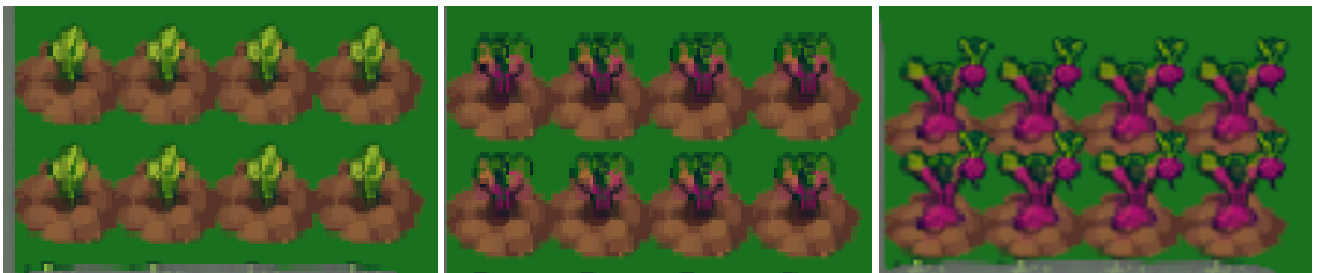


Рисунок 1.5. Візуалізація етапів росту овочу (буряк)

У цьому прикладі кожна стадія росту активується поступово з інтервалом у 10 секунд (для буряка). Такий підхід дозволяє гнучко налаштувати тривалість росту для кожної культури:

- Буряк — 10 секунд
- Морква — 15 секунд
- Картопля — 30 секунд
- Цибуля — 9 секунд
- Кабачок — 15 секунд
- Помідор — 27 секунд
- Баклажан — 30 секунд
- Перець чилі — 33 секунди
- Жовтий перець — 35 секунд
- Клубника — 37 секунд
- Ожина — 43 секунди
- Виноград — 47 секунд

Кількість стадій росту залежить від конкретної культури: більшість овочів мають по 4 стадії, проте деякі (наприклад, буряк, морква, картопля) — 3 стадії, що дозволяє створити різноманітність візуального розвитку.

| | | | | | | |
|-----|------|----------|------|--------|--------------------------------|------|
| | | | | | <i>РП 08. 11 001. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Дата | Підпис | | 27 |



Рисунок 1.6. Іконка готовності до збору над овочем

Після завершення росту гравець має підійти до овочу, щоб зібрати його. Для цього використовується перевірка відстані між гравцем і об'єктом. Збір реалізується через натискання клавіші «Е», що запускає метод `Harvest()`, в якому відбувається передача інформації до інвентаря та деактивація елементів сцени.

```
void Update()
{
    if (Vector2.Distance(transform.position, player.position) <= interactDistance)
    {
        if (Input.GetKeyDown(KeyCode.E))
            Harvest();
    }
}
```

Цей підхід забезпечує природну ігрову взаємодію. Використання `Vector2.Distance` дозволяє точно визначити момент, коли гравець наблизився

достатньо для збору. Завдяки цьому процес збору відчувається як навмисна дія, а не автоматична подія.

Успішно зібраний овоч передається в інвентар. Інвентар реалізований через список структур CropData — кожна структура зберігає назву овочу, його іконку та поточну кількість. Операція додавання нового овочу реалізується методом AddVegetable, який проходить по списку і знаходить потрібний овоч, збільшуючи його кількість.

```
public void AddVegetable(string name, int amount)
{
    foreach (var crop in crops)
    {
        if (crop.cropName == name)
        {
            crop.quantity += amount;
            break;
        }
    }
}
```

Вибір саме такого підходу пов'язаний із простотою масштабування: у майбутньому легко додати нові овочі без зміни логіки. Інтерфейс оновлюється автоматично, навіть якщо кількість дорівнює нулю — що створює у гравця розуміння повної картини врожаю.

| | | | | | | |
|-----|------|----------|------|--------|--------------------------------|------|
| | | | | | РП 08. 11 001. 00 ДП ПЗ | Арк. |
| Зм. | Арк. | № докум. | Дата | Підпис | | 29 |



Рисунок 1.7. Інтерфейс інвентаря з відображенням кількості зібраних овочів

Далі вступає в дію система NPC — неігрові персонажі, які виконують роль клієнтів. Вони з’являються по одному, проходять по маршруту до столу та роблять замовлення. Логіка маршруту включає в себе кілька точок, які NPC проходить з використанням `Vector3.MoveTowards`. Після досягнення точки біля столу викликається генератор замовлень.

```
public NPCOrder GenerateOrder()
{
    var crop = unlockedCrops[Random.Range(0, unlockedCrops.Count)];
    return new NPCOrder(crop.name, Random.Range(1, 5), crop.price);
}
```

Цей код вибирає випадковий овоч із переліку розблокованих культур і генерує випадкову кількість. Такий підхід забезпечує гнучкість: з розвитком гри перелік `unlockedCrops` розширюється, отже NPC замовляють складніші культури. Це сприяє прогресії складності та підтримує інтерес гравця.



Рисунок 1.8. NPC біля столу з іконкою замовлення над головою

Коли гравець підходить до NPC, відбувається перевірка наявності потрібного товару в інвентарі. Якщо кількість достатня — товар списується, а гравцеві нараховується прибуток. У цей момент перевіряється, чи було переглянуто рекламу: якщо так — сума подвоюється.

```
int totalEarned = order.totalPrice;
```

```
if (AdManager.Instance.WasAdWatched())
```

```
totalEarned *= 2;
```

```
MoneyStorage.Instance.AddMoney(totalEarned);
```

Це дозволяє реалізувати базову монетизацію: гравець не зобов'язаний дивитися рекламу, але отримує бонус у вигляді більшого прибутку. Система реклами працює через таймер: коли він доходить до 0:00, з'являється активна кнопка, яка запускає фіктивний перегляд ролика тривалістю 10 секунд. Після завершення активується прапорець `adWatched`, який діє для наступного замовлення.



Рисунок 1.9. Таймер реклами та кнопка активації бонусу

Таким чином, усі механіки гри об'єднані в логічний цикл: посадка або активація овочу → ріст через стадії → збір → облік в інвентарі → поява NPC → генерація замовлення → продаж → нарахування прибутку → використання реклами → відкриття нових культур. Цикл є замкненим і може розширюватися без порушення загальної логіки.

Саме тому були обрані ці підходи: корутини — для управління часом росту, списки структур — для зручної роботи з даними, окремі менеджери — для модульності та масштабованості. Це забезпечує зручність підтримки проєкту, можливість гнучкого оновлення та додавання нового контенту без серйозних змін у кодовій базі.

У підсумку, реалізація ключових механік створила завершену геймплейну систему, яка дає гравцеві чіткі цілі, візуальний прогрес, мотивацію до взаємодії та можливість розвитку. Усі дії логічно пов'язані, що робить ігровий досвід послідовним і приємним.

1.7 Внутрішня економіка гри

Щоб гра була цікавою та зрозумілою, всі її механіки об'єднані навколо внутрішньої економіки. Це означає, що кожен етап гри — вирощування овочів, збір урожаю, виконання замовлень, перегляд реклами, розблокування нових культур — впливає на прибуток гравця та його можливості для розвитку ферми.

У грі немає абстрактних балів чи випадкових бонусів. Усе засноване на простій логіці: саджаєш овочі, чекаєш поки вони виростуть, збираєш, продаєш — і отримуєш за це гроші. А вже за ці гроші можна відкрити нові грядки з новими культурами, які приносять більше прибутку. Це створює послідовність, яка дає гравцеві чітке відчуття прогресу.

Не всі овочі доступні з самого початку. Щоб розблокувати нову культуру, потрібно зібрати певну суму грошей. Наприклад, морква відкривається після того, як гравець заробив 20 грн, цибуля — після 45 грн, капуста — після 70 грн тощо.

| | | | | | | |
|-----|------|----------|------|--------|--------------------------------|------|
| | | | | | <i>РП 08. 11 001. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Дата | Підпис | | 32 |

Чим дорожча культура — тим пізніше вона відкривається. Завдяки цьому гравець поступово відкриває новий контент, і йому завжди є до чого прагнути. Така система поступового розблокування допомагає підтримувати інтерес та динаміку гри.

Щоб пришвидшити накопичення прибутку, гравець може переглянути рекламу. Після цього наступне замовлення NPC принесе вдвічі більше грошей. Наприклад, якщо замовлення приносить 12 грн, то після перегляду реклами — 24 грн. Це дозволяє гравцеві швидше досягати потрібної суми для відкриття нових культур, не змінюючи основну суть гри. Тобто, реклама стає не просто джерелом доходу для проекту, а ще й елементом, що вписується у внутрішню логіку ігрового процесу.

Ці умови прописані в Unity-компоненті LevelManager, де для кожної культури вказано "Unlock Amount" — скільки потрібно заробити, щоб вона стала доступною. Нижче наведено приклад таких умов:

Таблиця 1.2. При якому прибутку розблоковується овоч

| Овоч | Необхідний баланс для відкриття |
|----------------|---------------------------------|
| Буряк | 0 (доступний одразу) |
| Морква | 20 |
| Цибуля | 45 |
| Капуста | 70 |
| Картопля | 90 |
| Томат | 120 |
| Баклажан | 150 |
| Гострий перець | 180 |
| Жовтий перець | 200 |
| Полуниця | 230 |
| Ожина | 270 |
| Виноград | 300 |

Чим дорожча культура — тим більше вона приносить. Але й росте вона довше. Кожна культура має кілька стадій росту, і кожна стадія триває фіксований час. Наприклад, буряк має 3 стадії по 10 секунд — тобто загальний час росту становить 30 секунд. А виноград має 4 стадії по 47 секунд — це вже 188 секунд. Таким чином, прибутковість овочу пов'язана не тільки з його ціною, а й зі швидкістю росту.

Таблиця 1.3. Таблиця ціни та стадій росту

| Овоч | Ціна за 1 шт. | Кількість стадій | Час росту(с) | Можливий прибуток (1-6шт.) |
|----------------|---------------|------------------|--------------|----------------------------|
| Буряк | 1 грн | 3 | 30 | 1-6 грн |
| Морква | 1грн | 3 | 45 | 1-6грн |
| Цибуля | 1грн | 4 | 36 | 1-6грн |
| Капуста | 2грн | 4 | 60 | 2-12грн |
| Картопля | 2грн | 3 | 90 | 2-12грн |
| Томат | 3грн | 4 | 108 | 3-18грн |
| Баклажан | 3грн | 4 | 120 | 3-18грн |
| Гострий перець | 4грн | 4 | 132 | 4-24грн |
| Жовтий перець | 4грн | 4 | 140 | 4-24грн |
| Полуниця | 5грн | 4 | 148 | 5-30грн |
| Ожина | 5грн | 4 | 172 | 5-30грн |
| Виноград | 5грн | 4 | 188 | 5-30грн |

Гравець сам обирає, що вигідніше — швидко збирати буряк і потрохи просуватись, або чекати довше, але отримати більший прибуток. Це додає простий елемент стратегії в гру. Перегляд реклами допомагає ще більше підсилити прогрес і стати корисним інструментом у досягненні цілей.

if (AdManager.Instance.WasAdWatched())

*totalEarned *= 2;*

Після кожного перегляду реклама стає тимчасово недоступною — знову активується лише через певний час. Це робить баланс справедливим і не дозволяє "накручувати" прибуток без обмежень.

Уся економіка гри побудована просто, але розумно. Гравець бачить зв'язок між діями та результатом: вирощую — заробляю — відкриваю нове. Кожна культура має свою ціну та час росту, що допомагає самому вибирати стратегію: або часто продавати дешеві овочі, або чекати і отримувати більше. А реклама в цьому допомагає — не нав'язливо, а у той момент, коли гравець цього дійсно хоче. Все це створює захопливу економічну систему, яка мотивує повертатись у гру знову і знову.

1.8 Система монетизації через відеорекламу

У сучасному мобільному геймінгу монетизація відіграє надзвичайно важливу роль, адже саме вона дозволяє розробникам створювати безкоштовні ігри, отримуючи при цьому прибуток, необхідний для підтримки й розвитку проєкту. Однією з найефективніших і найменш нав'язливих моделей є система відеореклами з винагородою (rewarded ads), яка полягає в тому, що гравець самостійно вирішує, чи хоче переглядати рекламний ролик в обмін на певний бонус. Такий підхід є етичним і зручним, оскільки не примушує користувача до перегляду, як це часто трапляється з обов'язковими інтерстиціальними оголошеннями, а навпаки — мотивує добровільно включитися у процес, коли це йому вигідно.

У рамках розробки цієї гри було реалізовано власну спрощену систему монетизації через відеорекламу, яка дозволяє наочно продемонструвати логіку функціонування rewarded ads без необхідності інтеграції зовнішніх SDK, таких як Unity Ads або AdMob. Вибір на користь кастомного рішення був зроблений навмисно — для повного контролю над логікою, можливості тестування, відлагодження та демонстрації принципів монетизації на захисті дипломної

| | | | | | | |
|-----|------|----------|------|--------|--------------------------------|------|
| | | | | | <i>РП 08. 11 001. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Дата | Підпис | | 35 |

роботи. Це дало змогу максимально адаптувати систему під потреби саме цієї гри й уникнути зайвих залежностей від сторонніх сервісів.

Функціональність реалізована за наступним сценарієм: на екрані відображається таймер, який показує, коли наступна реклама стане доступною для перегляду. Він веде зворотний відлік у реальному часі, і щойно досягає нуля — змінюється на повідомлення «+2х», що символізує можливість отримати подвоєний прибуток. Гравець може натиснути кнопку перегляду реклами, після чого на весь екран з'являється відеоплеєр, де імітується перегляд ролика. Після завершення перегляду вмикається внутрішній прапорець, який сигналізує системі, що при наступному продажі NPC прибуток має бути збільшено у два рази. Після цього таймер перезапускається, і гравець знову очікує новий цикл. Такий підхід створює органічну інтеграцію реклами в геймплей, не відволікаючи, а навпаки — мотивуючи гравця на взаємодію.

Технічно реалізація складається з кількох основних компонентів. Насамперед — це таймер, який написано у вигляді окремого скрипта, що оновлює UI у реальному часі, перераховує хвилини й секунди та виводить їх у зручному форматі. Коли таймер досягає нуля, текст змінюється на «+2х!», що чітко дає гравцеві зрозуміти, що бонус готовий до використання. Приклад реалізації виглядає наступним чином:

```
public class AdTimer : MonoBehaviour
{
    public TextMeshProUGUI timerText;
    public float cooldown = 180f;
    private float timer;

    void Update()
    {
        if (timer > 0)
```

| | | | | | | |
|-----|------|----------|------|--------|--------------------------------|------|
| | | | | | <i>РП 08. 11 001. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Дата | Підпис | | 36 |

```

    {
        timer -= Time.deltaTime;
        int minutes = Mathf.FloorToInt(timer / 60);
        int seconds = Mathf.FloorToInt(timer % 60);
        timerText.text = $"{minutes:00}:{seconds:00}";
    }
    else
    {
        timerText.text = «+2x!»;
    }
}

public void ResetTimer()
{
    timer = cooldown;
}
}

```

Другий важливий елемент — це відеоплеєр. У грі він реалізований через компонент `VideoPlayer`, який запускається в повноекранному режимі після натискання кнопки. Його поява супроводжується відкриттям спеціальної панелі, що затемнює решту інтерфейсу та зосереджує увагу гравця на перегляді. Після завершення відео викликається подія, яка активує відповідний флаг `doubleRewardActive`, що зберігається в `GameManager` або іншому глобальному класі:

```

public class AdVideoPlayer : MonoBehaviour
{
    public GameObject videoPanel;
}

```

```

public VideoPlayer videoPlayer;

private void Start()
{
    videoPlayer.loopPointReached += OnVideoFinished;
    videoPanel.SetActive(false);
}

public void ShowAd()
{
    videoPanel.SetActive(true);
    videoPlayer.Play();
}

private void OnVideoFinished(VideoPlayer vp)
{
    GameManager.Instance.SetDoubleReward(true);
    videoPanel.SetActive(false);
}
}

```

Нарешті, логіка застосування бонусу реалізована в тій частині коду, яка відповідає за обробку продажу овочів. Саме тут перевіряється, чи активний бонус x2, і якщо так — фінальна сума, яку отримує гравець, подвоюється. Після цього флаг деактивується, щоб бонус не накопичувався:

```

public class MoneyStorage : MonoBehaviour
{
    public int currentMoney = 0;
}

```

| | | | | | | |
|-----|------|----------|------|--------|--------------------------------|------|
| | | | | | РП 08. 11 001. 00 ДП ПЗ | Арк. |
| Зм. | Арк. | № докум. | Дата | Підпис | | 38 |

```

public void SellVegetable(int price, int amount)
{
    int total = price * amount;
    if (GameManager.Instance.doubleRewardActive)
    {
        total *= 2;
        GameManager.Instance.SetDoubleReward(false);
    }
    currentMoney += total;
}
}

```

Переваги реалізованої системи очевидні. По-перше, вона не примушує гравця до дії — перегляд реклами є повністю добровільним. По-друге, бонус, який надається після перегляду, є цілком відчутним — гравець реально бачить, що отримує вдвічі більше грошей, і може це використати для розблокування нових культур або пришвидшення прогресу. По-третє, інтерфейс взаємодії з рекламою органічно вбудований у загальну логіку гри — таймер, кнопка перегляду, відео й бонус працюють як єдина система, яка не руйнує загальне враження від ігрового процесу.

Таким чином, реалізована механіка rewarded ads стала не лише ефективним засобом демонстрації монетизації, а й повноцінною частиною геймплею. Вона дозволяє гравцеві самостійно обирати спосіб прискорення прогресу, не викликаючи негативного досвіду чи відрази до гри. У майбутньому, за умови переходу до повноцінного релізу гри, до цієї системи легко можна буде інтегрувати справжні рекламні SDK, замінивши лише частину логіки, не змінюючи архітектуру всієї гри. Але навіть у поточному вигляді дана система чудово виконує

| | | | | | | |
|-----|------|----------|------|--------|--------------------------------|------|
| | | | | | РП 08. 11 001. 00 ДП ПЗ | Арк. |
| Зм. | Арк. | № докум. | Дата | Підпис | | 39 |

свої функції та підтверджує, що монетизація може бути не лише прибутковою, а й дружньою до користувача.

1.9 Технології, що використовуються у грі

Створення навіть нескладної на перший погляд мобільної гри передбачає використання цілого ряду технологій, інструментів та платформ, кожен з яких виконує свою, але водночас критично важливу функцію. Гра — це результат не лише творчого задуму, а й продуманого технічного втілення, в якому все — від розмітки сцени до обробки натискання кнопки — є частиною складного ланцюга взаємодій. У даному дипломному проєкті реалізовано повноцінний технологічний стек, адаптований до потреб студентського рівня, але побудований на професійних принципах розробки, що застосовуються в реальній індустрії мобільних застосунків.

Unity — центральна платформа, навколо якої вибудовано весь процес розробки. Це один з найпопулярніших рушіїв у світі, який дозволяє створювати як двовимірні, так і тривимірні ігри. У нашій грі використано саме 2D-сцену, що дозволяє легше працювати зі спрайтами, зменшити навантаження на пристрій та зробити гру доступною навіть на слабких смартфонах. Unity дає змогу інтегрувати фізику, аудіо, анімації, інтерфейс, логіку взаємодії — усе в межах однієї сцени. Сцена в грі складається з кількох зон — поле з грядками, персонаж, NPC, точки зупинок, інвентар та панель монетизації.

Мова програмування C# використовується для написання усієї логіки гри. Ця мова є однією з найзручніших у використанні, особливо в контексті об'єктно-орієнтованого підходу, який чудово підходить для геймдеву. Усі класи, методи, змінні, події — реалізовані на C#. Ця мова дозволяє реалізовувати гнучку поведінку об'єктів, управляти подіями у грі, створювати модульну архітектуру з можливістю подальшого масштабування. Наприклад, рух гравця, перевірка стану

| | | | | | | |
|-----|------|----------|------|--------|--------------------------------|------|
| | | | | | <i>РП 08. 11 001. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Дата | Підпис | | 40 |

овочів, обробка замовлень NPC — усе це написано на C# та інтегровано безпосередньо в Unity-сцену.

Для написання коду застосовувалось середовище розробки Visual Studio, яке надає комфортну роботу з великими проєктами: тут зручно шукати помилки, використовувати автозаповнення, коментувати блоки коду, працювати з розділенням на файли та збірками. Середовище дозволяє уникнути багатьох дрібних помилок завдяки своїм підказкам та підсвічуванню синтаксису.

Для анімації всіх персонажів та об'єктів у грі застосовується Animator Controller в Unity. Завдяки йому задаються переходи між анімаціями в залежності від стану об'єкта. Наприклад, анімація для гравця перемикається між «стоячою» та «рухомою» залежно від натискання клавіш. Те саме стосується NPC — вони рухаються за заданими точками, повертаються в потрібний момент і стають біля столу, очікуючи замовлення.

Спрайти — візуальні зображення всіх об'єктів, таких як овочі, персонажі, іконки, кнопки — створювалися в Piskel (для піксельної графіки) або редагувалися в Photoshop (наприклад, тло чи великі елементи UI). Зображення були розділені на окремі кадри для створення анімацій, і за потреби оптимізовані за розміром, щоб уникнути перевантаження пам'яті пристрою.

Інтерфейс побудований на основі Canvas UI Unity — гнучкої системи, яка дозволяє створювати екранні елементи, що адаптуються під різні розміри екранів. Всі текстові елементи реалізовані за допомогою TextMeshPro — сучасного інструменту для виводу чіткого, масштабованого тексту. Елементи UI включають панель з грішми, таблицю інвентаря, таймер відеореклами, текстове замовлення NPC та інші інформаційні блоки.

Особливу увагу було приділено реалізації механізму монетизації, адже вона є центральною особливістю гри. У проєкті використано Unity Video Player — компонент, який дозволяє демонструвати відео (у нашому випадку — демонстраційне відео реклами). Після перегляду відео, гравець отримує бонус у

| | | | | | | |
|-----|------|----------|------|--------|--------------------------------|------|
| | | | | | <i>РП 08. 11 001. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Дата | Підпис | | 41 |

вигляді подвоєної винагороди при продажі овочів. Це стимулює користувача активно взаємодіяти з системою монетизації, не порушуючи його ігрового досвіду. Для організації даних було використано List<> та серіалізацію через [System.Serializable], що дозволяє зручно зберігати всі овочі, їх кількість та відповідні іконки. Збір інформації про продажі, замовлення NPC, рух грошей, відкриття інвентаря — усе це зав'язано на внутрішні менеджери, які реалізовані в окремих скриптах: InventoryManager, OrderGenerator, MoneyStorage тощо.

Щоб не втратити дані під час роботи, проєкт регулярно зберігався у хмарному сховищі Google Drive, а також створювались резервні копії. Це дозволяло уникнути втрати прогресу та швидко переміщати проєкт між різними комп'ютерами. За потреби можливе використання GitHub — платформи для збереження версій коду, але в межах цієї гри було достатньо Google Drive.

Також було протестовано використання Unity Profiler для аналізу продуктивності: перевірявся FPS, навантаження на CPU та GPU, перевірялись затримки при взаємодії гравця з об'єктами. Це дало змогу оптимізувати частину скриптів — наприклад, обмежити виклики Update() у неактивних NPC, зменшити використання об'єктів, які постійно шукаються через FindObjectOfType().

У підсумку, уся сукупність технологій дозволила реалізувати повноцінний цикл створення мобільної гри — від побудови сцени до фінального тестування геймплею. Кожна обрана технологія не просто виконувала свою функцію, а стала частиною цілісного середовища, в якому народилася, розвивалася та була доведена до фіналу унікальна ігрова механіка.

1.9 Тестування, результати та ігровий процес

Після завершення основних етапів розробки критично важливим кроком стало ретельне тестування гри — як з точки зору технічної коректності реалізованих механік, так і з позиції загального користувацького досвіду. Метою цього етапу було не лише переконатися в тому, що всі частини гри функціонують

| | | | | | | |
|-----|------|----------|------|--------|--------------------------------|------|
| | | | | | <i>РП 08. 11 001. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Дата | Підпис | | 42 |

належним чином, а й виявити слабкі місця, непередбачувані ситуації, незручності в інтерфейсі чи візуальному сприйнятті, а також перевірити баланс ігрової логіки та циклу.

Тестування проходило у кілька логічно послідовних фаз. Спочатку проводилось локальне внутрішнє тестування розробником, де перевірялися базові елементи:

- запуск сцени
- відображення гравця
- керування клавішами
- функціональність
- анімація росту овочів
- спрацьовування індикаторів
- взаємодія з колайдерами

Цей етап дозволив ще до глибокої перевірки впевнитися, що основні компоненти не містять критичних помилок.

Далі виконувалося функціональне сценарне тестування. Моделювались можливі ситуації, з якими може стикнутись реальний гравець:

- спроба зібрати овоч до появи іконки готовності
- NPC приходить, але замовлення не створено
- гравець не має потрібної кількості продукції в інвентарі
- інвентар заповнений, але дані не оновлюються
- реклама недоступна, але кнопка активна

Окремо перевірявся UI-інтерфейс. Тут тестувалась адаптація до різних розмірів екранів, зокрема на смартфонах зі співвідношенням сторін 18:9, 16:9 і 20:9. Зверталася увага на такі моменти:

- перекривання кнопок
- відсутність відступів
- некоректне відображення чисел у лічильнику овочів
- зсув іконки реклами на краю екрану

| | | | | | | |
|-----|------|----------|------|--------|--------------------------------|------|
| | | | | | <i>РП 08. 11 001. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Дата | Підпис | | 43 |

Завдяки цим перевіркам вдалося адаптувати інтерфейс до більшості пристроїв.

Функціональні перевірки охоплювали:

- стабільність руху гравця, точність зупинки біля грядок і NPC
- коректну поетапну появу NPC (без одночасного виклику кількох персонажів)
- генерацію замовлень згідно з розблокованими овочами
- перевірку механіки LevelManager — чи не пропускаються кроки розблокування
- спрацьовування логіки реклами — активація, перегляд, подвоєння
- зміни значень у InventoryManager після збору овочів
- накопичення валюти та запуск нової культури
- збереження стану гри між перезапусками (у рамках однієї сесії)
- динамічне відображення замовлень NPC через іконки над головою
- коректну анімацію збирання врожаю
- точне відображення цін продажу після реклами
- логіку перегляду реклами через поле "isAdWatched"
- зв'язок генератора замовлень і NPC через public NPCOrder GenerateOrder(GameObject npc)

Увага також приділялася граничним ситуаціям:

- гравець дивиться рекламу і виходить з гри — перевірка збереження бонусу
- NPC не отримує усе замовлення — чи зникає коректно
- клік по неактивному овочу — перевірка координат і колайдерів
- нова культура не запускає грядки — перевірка CropGrowth
- збір врожаю не оновлює інвентар — діагностика InventoryUIManager
- префаби NPC втрачають посилання на Main Camera — заміна на сценні об'єкти
- AddVegetable() не змінює UI — додано RefreshUI()

| | | | | | | |
|-----|------|----------|------|--------|--------------------------------|------|
| | | | | | <i>РП 08. 11 001. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Дата | Підпис | | 44 |

Таблиця 1.4. Приклади тестування

| № | Тестовий сценарій | Очікуваний результат | Код або причина / виправлення |
|----|--|---|---|
| 1 | Натиснути кнопку реклами до завершення таймера | Кнопка має бути неактивною | <code>interactable = (timer <= 0)</code> |
| 2 | NPC приходить без замовлення | Замовлення має генеруватись автоматично | Додано генерацію в <code>OnReachTable()</code> |
| 3 | Збір овочу до іконки | Збір не повинен відбуватись, UI не змінюється | Перевірка <code>if (isReady)</code> перед збором |
| 4 | Інвентар повний, збираємо ще овоч | Нові овочі не додаються, UI не ламається | Динамічне масштабування інвентаря через <code>ScrollRect</code> |
| 5 | Реклама недоступна, але кнопка активна | Кнопка сірого кольору, натискання заблоковано | <code>if (adAvailable) button.SetActive(true)</code> |
| 6 | Збирання не оновлює інвентар | Кількість оновлюється у UI | Додано <code>InventoryUIManager.RefreshUI()</code> |
| 7 | NPC не зникає, якщо не отримав замовлення | Має піти після таймера | Метод <code>LeaveScene()</code> викликається через таймер |
| 8 | Невірне оновлення грошей після продажу | Сума одразу оновлюється у верхньому куті | Метод <code>AddMoney()</code> викликає <code>UpdateMoneyUI()</code> |
| 9 | Префаб NPC втрачає посилання на Main Camera | Камера має бути підключена | Відмова від префабів, використання NPC на сцені |
| 10 | Клік по неактивному овочу | Не відбувається збір або помилка | Колайдери перевіряються перед дією |
| 11 | Зібраний овоч не додається в інвентар | Має одразу відобразитися | <code>AddVegetable() + RefreshUI()</code> |

| | | | |
|----|---|---------------------|---|
| 12 | Після перегляду реклами не подвоюється прибуток | Має бути множник ×2 | Поле isAdWatched, що впливає на розрахунок прибутку |
|----|---|---------------------|---|

Проведене тестування дало змогу не лише виявити та усунути критичні помилки, але й значно покращити якість взаємодії гравця з грою. Особливої уваги було приділено реалістичності логіки дій, плавності переходів між ігровими станами, а також стабільності роботи UI-компонентів та внутрішніх ігрових механік.

Додатково перевірявся сценарій, коли гравець після перегляду реклами одразу виходить із гри — було підтверджено, що бонус зберігається на наступну сесію. Це забезпечує гнучкість і надійність ігрового процесу при різних типах поведінки користувача.

Важливим моментом стала також відмова від використання NPC як префабів — під час розробки з'ясувалося, що префаби втрачають важливі посилання (зокрема на Main Camera), через що було прийнято рішення створювати NPC безпосередньо в сцені. Це дозволило уникнути нестабільностей і зменшити кількість помилок при оновленні або завантаженні сцени.

Загалом, тестування дало змогу:

- удосконалити взаємодію між модулями гри;
- перевірити відповідність усіх ключових функцій технічному завданню;
- зробити гру більш адаптивною до різних пристроїв;
- забезпечити стабільну логіку обліку врожаю, продажу і взаємодії з NPC.

Завдяки системному підходу до перевірки, вдалося створити стійкий до помилок ігровий продукт, орієнтований на якісний користувацький досвід та високий рівень задоволення гравців. Ретельно проведене тестування стало фінальним, але не менш значущим кроком у повному циклі розробки гри.

1.10 Майбутні доопрацювання та перспективи розвитку

| | | | | | | |
|-----|------|----------|------|--------|--------------------------------|------|
| | | | | | <i>РП 08. 11 001. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Дата | Підпис | | 46 |

Незважаючи на те, що поточна версія гри містить завершений цикл геймплею, її архітектура з самого початку закладена як масштабована та відкрита для розвитку. Усі ключові механіки реалізовані у вигляді незалежних модулів: логіка росту культур, взаємодія з NPC, збір та продаж овочів, система реклами й інвентар. Це дозволяє легко розширювати функціонал, додаючи нові елементи без порушення загальної структури проєкту. Таке гнучке проєктування не лише зменшує витрати часу на розробку нових функцій, а й відкриває великі можливості для покращення комерційної привабливості гри.

Одним із перших напрямків розвитку є додаткові культури, зокрема фрукти, ягоди, декоративні рослини, екзотичні види овочів тощо. Вони можуть мати різні характеристики: тривалість росту, кількість стадій, прибутковість, рідкість. Наприклад, нові культури можуть ставати доступними тільки після накопичення певної кількості прибутку або виконання особливих завдань. Це створює довгострокову мету для гравця та стимулює повертатися до гри щодня. Внаслідок цього зростає час, проведений у застосунку, що безпосередньо впливає на кількість переглядів реклами та підвищує доходи від монетизації.

Ще одним потужним інструментом залучення гравців стане система помічників — автономних персонажів (фермери, тварини, дрони), які виконують частину дій за гравця. Вони можуть відкриватися за внутрішню валюту, як винагорода за активність або за перегляд відеореклами. Це дозволить користувачам оптимізувати геймплей, а розробникам — збільшити кількість переглядів реклами, адже гравці з більшою охотою будуть взаємодіяти з нагородною системою, якщо за це отримуватимуть реальну допомогу в грі.

Планується впровадити розширення сцени ферми. Це можуть бути нові ігрові зони, теплиці, річкові ділянки, зимові або тропічні регіони. У кожній зоні — окремі умови для вирощування овочів та унікальні культури. Гравець зможе інвестувати час і ресурси в розвиток нових ділянок. Така багатозональність

| | | | | | | |
|-----|------|----------|------|--------|--------------------------------|------|
| | | | | | <i>РП 08. 11 001. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Дата | Підпис | | 47 |

створює глибший геймплей і дає додаткову мотивацію для тривалого використання гри, що знову ж таки підвищує ефективність реклами в довготривалій перспективі.

Особливо важливим кроком стане декоративне оформлення ферми. Гравець зможе змінювати стиль своєї локації, встановлювати прикраси (фонтанчики, лавки, дерева, скульптури), змінювати погодні теми. Частина з елементів буде доступна за внутрішню валюту, інші — тільки за перегляд реклами або перемогу в конкурсах. Це створює престижність деяких предметів, підштовхує гравців до регулярної взаємодії з проектом і стимулює соціальний інтерес, коли користувачі хочуть виділитися серед інших.

Наймасштабнішим напрямком розвитку стане впровадження онлайн-функціоналу:

- Глобальний рейтинг гравців — система, яка дозволяє користувачам змагатися за місце в топі за доходом, кількістю зібраного врожаю, активністю тощо.
- Відвідування ферм інших гравців — гравці зможуть побачити, як виглядає чужа ферма, надихнутися або порівняти прогрес.
- Події та турніри — регулярні тижневі чи місячні івенти, за підсумками яких найкращі гравці отримують ексклюзивні нагороди, наприклад, унікальні декоративні предмети, яких немає у звичайному магазині.

Цей підхід стимулює соціальну взаємодію, конкуренцію та залучення, а також дозволяє проводити внутрішньоігрові розіграші та акції. Наприклад, предмети типу "золотий фонтан" або "новорічна ялинка" можна буде отримати лише перемігши у рейтингу. Такі обмежені предмети створюють штучний дефіцит, що збільшує інтерес до участі у подіях, сприяє залученню нових користувачів, адже гравці діляться результатами в соцмережах, розповідають друзям і рекомендують гру.

У свою чергу, висока активність користувачів в онлайні прямо підвищує кількість щоденних сесій, переглядів відеореклами, і в результаті — доходи

| | | | | | | |
|-----|------|----------|------|--------|--------------------------------|------|
| | | | | | <i>РП 08. 11 001. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Дата | Підпис | | 48 |

розробника. Онлайн-функціонал дозволяє проводити спонсорвані івенти, які можуть включати партнерську рекламу або інтеграцію з брендами.

Ще одна важлива перевага — зростання кількості завантажень. Наявність рейтингів, соціальних функцій, сезонних івентів, унікальних нагород та регулярного оновлення контенту значно підвищує ймовірність рекомендацій у маркетплейсах, отримання позитивних відгуків, вірусного поширення через соцмережі. Користувачі охочіше завантажують гру, де є активна спільнота, престиж, новинки та можливість виграти щось ексклюзивне.

Таким чином, усі запропоновані доопрацювання мають не лише геймплейну або естетичну цінність, а й чітку економічну стратегію. Вони спрямовані на збільшення тривалості сесій, зростання лояльності гравців, розширення аудиторії, підвищення доходу від реклами та мікротранзакцій. Усі ці елементи разом можуть трансформувати дипломну гру з навчального проєкту у прибутковий мобільний продукт, конкурентоспроможний на сучасному ігровому ринку.

| | | | | | | |
|-----|------|----------|------|--------|--------------------------------|------|
| | | | | | <i>РП 08. 11 001. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Дата | Підпис | | 49 |

2. ЕКОНОМІЧНИЙ РОЗДІЛ

2.1 Резюме

У даному дипломному проєкті розроблено програмний продукт (ПП) — мобільну гру-симулятор ферми з елементами монетизації, реалізовану в середовищі Unity з використанням мови програмування C#. Гравець має можливість вирощувати різні види овочів, збирати врожай, виконувати замовлення неігрових персонажів (NPC) та отримувати віртуальну винагороду. Крім того, реалізовано механізм збільшення прибутку за перегляд реклами, що є прикладом вбудованої монетизації мобільного додатку.

Якість програмного продукту визначається з погляду користувача (зручність інтерфейсу, простота навігації), ефективності використання ресурсів мобільного пристрою та виконанням вимог технічного завдання. У цьому розділі буде проведено оцінку трудомісткості розробки програмного забезпечення та здійснено розрахунок його орієнтовної вартості нормативним методом.

2.2 Визначення трудомісткості розробки програмного забезпечення

Для оцінки трудомісткості використано нормативний метод на основі аналогії з існуючими рішеннями. Оскільки мобільна гра містить елементи автоматизації, симуляції та взаємодії з користувачем, її аналогом обрано програму автоматизованих розрахунків.

Таблиця 2.1. Каталог аналогів програмного забезпечення

| Найменування ПП | Обсяг функції ПП – $V_{o, \text{усл. машинних команд}}$. |
|--------------------------------|---|
| ПП автоматизованих розрахунків | 1300 – 8600 |

| | | | | | | |
|-----|------|----------|------|--------|--------------------------------|------|
| | | | | | <i>РП 08. 11 002. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Дата | Підпис | | 50 |

На основі аналізу вихідного коду (2321 рядок) визначено, що умовний обсяг становить 2.0 тис. умовних машинних команд.

Таблиця 2.2. Визначення норми часу

| Обсяг ПП, тис. ум. машин. команд | Норма часу, люд./год |
|----------------------------------|----------------------|
| 2.00 | 244 |

На основі таблиці 2.2 визначено укрупнену норму часу, яка скоригована з урахуванням умов розробки ПП на комп'ютері..

$$K_k=0,75$$

$$T_{ra}=244 \times 0,75=183 \text{ люд./год}$$

Трудомісткість розраховується для кожного етапу окремо з урахуванням аналогів, складності, новизни та використання стандартних модулів за відповідними формулами:

- Розробка технічного завдання

$$T_{TЗ} = T^a p \times L_1 \times K_H$$

$$T_{TЗ} = 183 \times 0,12 \times 0,7 = 15,4$$

- Розробка технічного проекту

$$T_{ТП} = T^a p \times L_2 \times K_H$$

$$T_{ТП} = 183 \times 0,11 \times 0,7 = 14,1$$

- Розробка робочого проекту

$$T_{РП} = T^a p \times L_3 \times K_H \times K_T$$

$$T_{РП} = 183 \times 0,61 \times 0,7 \times 0,7 = 54,6$$

Таблиця 2.5. Значення коефіцієнта ступеня використання в розробці типових програм

| Ступінь охоплення реалізованих функцій розроблювального ПП типовими програмами, % | Значення K_T |
|---|----------------|
| 40-60 | 0,7 |

Таблиця 2.3. Значення питомих коефіцієнтів трудомісткості стадії в загальній трудомісткості розробки ПП.

| Код стадії | Ступінь новизни |
|------------|-----------------|
| | В |

| | |
|----------------------|------|
| ТЗ (L ₁) | 0,12 |
| ТП (L ₂) | 0,11 |
| РП (L ₃) | 0,61 |

Таблиця 2.4. Значення поправочного коефіцієнта, що враховує ступінь новизни

| Код ступеня новизни | Ступінь новизни | Значення K _н |
|---------------------|------------------|-------------------------|
| В | ПП маючий аналог | 0,7 |

Тому що розробка системи є ПЗ, що має аналоги програмних продуктів, то код ступеня новизни для мого ПЗ — В, а значення коефіцієнта K_н=0,7

По таблиці 2.4, знаючи код ступеня новизни, тепер можна визначити значення питомих коефіцієнтів трудомісткості (по таблиці 2.3):

$$L_1=0,12; L_2=0,11; L_3=0,61$$

Таблиця 2.5. Значення коефіцієнта ступеня використання в розробці типових програм

| Ступінь охоплення реалізованих функцій розроблювального ПП типовими програмами, % | Значення K _т |
|---|-------------------------|
| 40-60 | 0,7 |

У розробленому програмному продукті використовується від 40 до 60% відсотків існуючих функцій, це значить, що K_т=0,7.

Тепер потрібно розрахувати трудомісткість по кожному етапу окремо:

Трудомісткість технічного завдання

$$T_{ТЗ} = T^a p * L_1 * K_n = 183 \times 0,12 \times 0,7 = 15,4 \text{ (люд./год)}$$

Трудомісткість розробки технічного проекту

$$T_{ТП} = T^a p * L_2 * K_n = 183 \times 0,11 \times 0,7 = 14,1 \text{ (люд./год)}$$

Трудомісткість розробки робочого проекту

$$T_{РП} = T^a p * L_3 * K_n * K_t = 183 \times 0,61 \times 0,7 \times 0,7 = 54,6 \text{ (люд./год)}$$

Для подальших розрахунків необхідно визначити кількість паперу, витраченого на кожен етап:

$$N_{ТЗ}=13 \text{ (стор.)}; N_{ТП}=17 \text{ (стор.)}; N_{РП}=12 \text{ (стор.)}$$

Таблиця 2.6. Розрахунок трудомісткості ПП

| Найменування етапів | Розрахунок, години. | | |
|----------------------------------|--|---------------------------------------|--|
| 1 | Розробка ПП | Контроль керівника | Нормоконтроль |
| 1.ТЗ | $T_{РТЗ}=15,4$ | $T_{КК}=0,7*N_{ТЗ}=0.7\times 13=9.1$ | $T_{НК}=0,15*N_{ТЗ}=0.15\times 13=2.0$ |
| 2.Розробка ТП | $T_{РТП}=14,1$ | $T_{КК}=0,7*N_{ТП}=0.7\times 17=11.9$ | $T_{НК}=0,15*N_{ТП}=0.15\times 17=2.6$ |
| 3.Розробка РП | $T_{РРП}=54,6$ | $T_{КК}=0,7*N_{РП}=0.7\times 12=8.4$ | $T_{НК}=0,15*N_{РП}=0.15\times 12=1.8$ |
| 4.Розробка пояснювальної записки | $T_{РПЗ}=1,5*N_{ПЗ}=1.5\times 42=63.0$ | $T_{КК}=0,7*N_{ТЗ}=0.7\times 42=29.4$ | $T_{НК}=0,15*N_{ПЗ}=0.15\times 42=6.3$ |
| Усього, в т.ч.: | $T_{ПП}=147,1$ | | |
| - на розробку | $T_p=147,1$ | | |
| -контролькерівника | | $T_{КК}=58,8$ | |
| - нормоконтроль | | | $T_{НК}=12,7$ |

На основі таблиці 2.6 розрахуємо тривалість розробки в роках:

$$T_{ПП} = T / (8,0 * 0,73 * 360) = 218,6 / (8,0 * 0,73 * 360) = 0,104(р.) = 1,25 \text{ місяці}$$

2.3 Розрахунок ціни програмного продукту

У цьому розділі розраховується вартість розробки ПЗ: основна заробітна плата виконавців, матеріальні витрати, вартість машино-години та інші витрати. Розрахунок заробітної плати наведено в таблиці 6.7. Згідно зі статтею 8 «Закону про Державний бюджет України на 2024 рік», з 1 січня 2025 року мінімальна місячна зарплата становить 8000 грн, а погодинна ставка — 46 грн.

Таблиця 2.7 Розрахунок основної заробітної плати виконавців.

| Найменування робіт | Трудомісткість робіт, роб.години | Годинна тарифна ставка, грн.. | Розрахунок, грн. |
|--------------------------|----------------------------------|-------------------------------|------------------|
| 1.Розробка ПП | $T_p=147,1$ | 46 | 6766,6 |
| 2.Контроль керівника | $T_{КК}=58,8$ | 46 | 2704,8 |
| 3.Нормоконт-роль | $T_{КК}=12,7$ | 46 | 584,2 |
| Усього (З _о) | - | - | $Z_o=10055,6$ |

3. РОЗДІЛ ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

3.1 Потенційні шкідливі фактори при роботі над дипломним проєктом

Під час реалізації дипломного проєкту основним робочим середовищем стало персональне робоче місце з комп'ютером. Незважаючи на зовнішню безпеку такого формату роботи, він все ж супроводжується низкою шкідливих та потенційно небезпечних факторів, що можуть негативно впливати на фізичне та психоемоційне здоров'я.

До основних фізичних факторів належать:

- електромагнітне випромінювання від монітора та периферійних пристроїв;
- підвищене зорове навантаження внаслідок довготривалої концентрації на екрані;
- статична поза під час роботи, що призводить до перенапруження м'язів шиї, спини та рук;
- недостатній мікроклімат, наприклад, сухе повітря, висока або знижена температура, недостатня вентиляція;
- незбалансоване освітлення, відсутність прямого світла або наявність відблисків на екрані.

До психофізіологічних факторів відносяться:

- високе розумове навантаження, особливо при вирішенні технічних завдань;
- монотонність рухів і відсутність фізичної активності протягом тривалого часу;
- емоційне напруження, пов'язане з дедлайнами або помилками в роботі;
- можливе порушення режиму сну та відпочинку, зумовлене роботою у вечірній або нічний час.

| | | | | | | |
|-----|------|----------|------|--------|--------------------------------|------|
| | | | | | <i>РП 08. 11 003. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Дата | Підпис | | 55 |

Також слід враховувати електротехнічну небезпеку — ризик короткого замикання, перегріву пристроїв або ураження струмом у разі несправного обладнання.

Виявлення та розуміння цих факторів є першим кроком до їх нейтралізації за допомогою відповідних технічних і організаційних рішень, що буде детально розглянуто в наступних підрозділах.

3.2 Технічні та організаційні заходи безпеки

Для зниження впливу шкідливих факторів, які супроводжують роботу за комп'ютером, необхідно впровадити комплекс технічних і організаційних заходів. Вони спрямовані на створення безпечних і комфортних умов праці відповідно до чинних нормативів з охорони праці.

До технічних заходів належать:

- Оптимізація мікроклімату приміщення. Температура повітря в межах 22–24 °С, вологість — 40–60%. За потреби використовуються кондиціонери, зволожувачі або вентиляція. Це дозволяє уникнути пересушування слизових оболонок і підтримувати комфортне дихання.
- Освітлення. Робоче місце повинно бути добре освітленим. Перевагу надають комбінованому освітленню — природному (з вікна) та штучному (настільна лампа з м'яким світлом). Джерела світла мають бути розміщені так, щоб не створювати відблисків на екрані.
- Електробезпека. Всі електропристрої мають бути справними і заземленими. Використання мережевих фільтрів із захистом від перенапруг є обов'язковим. Дроти не повинні створювати загрози спотикання або випадкового пошкодження.
- Шумозахист. Рекомендується використовувати тихі моделі комп'ютерів або розміщувати системні блоки на достатній відстані від користувача.

До організаційних заходів належать:

| | | | | | | |
|-----|------|----------|------|--------|--------------------------------|------|
| | | | | | <i>РП 08. 11 003. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Дата | Підпис | | 56 |

- Раціональне планування робочого дня. Щоб уникнути перевтоми, слід робити перерви кожні 45–60 хвилин. Протягом перерви — встати, змінити положення тіла, зробити нескладну гімнастику або вправи для очей.
- Чергування видів діяльності. Якщо є можливість — слід чергувати розумову та фізичну активність або перемикатися між різними завданнями.
- Самоконтроль і самодисципліна. Працівник має слідкувати за своїм самопочуттям, підтримувати правильну поставу та дотримуватися режиму праці та відпочинку.
- Навчання основам безпеки праці. Навіть при роботі з ПК важливо знати правила першої допомоги, дій при короткому замиканні чи загорянні, а також ознайомитися з протипожежними інструкціями.

Реалізація зазначених заходів дозволяє значно знизити шкідливий вплив робочого середовища, підвищити комфорт, зберегти здоров'я і продуктивність під час довготривалої роботи з комп'ютером.

3.3 Облаштування безпечного та зручного робочого місця користувача ПК

Під час роботи над дипломним проектом основним робочим інструментом був персональний комп'ютер, тому особливу увагу варто приділити організації комфортного і безпечного робочого місця. Зручне розташування меблів і техніки допомагає зменшити навантаження на зір, опорно-руховий апарат і нервову систему.

Стіл має бути достатньо просторим і розташованим на висоті близько 70–75 см. Крісло бажано використовувати регульоване, зі спинкою, що підтримує поперек. Монітор рекомендується встановити на відстані 60–70 см від очей, на рівні зору, щоб уникнути зорового напруження. Клавіатура та миша повинні розміщуватись так, щоб руки залишалися у природному положенні, а зап'ястя — не згиналися.

| | | | | | | |
|-----|------|----------|------|--------|--------------------------------|------|
| | | | | | <i>РП 08. 11 003. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Дата | Підпис | | 57 |

Освітлення має бути м'яким і рівномірним. Удень бажано працювати при природному світлі, а ввечері — використовувати настільну лампу з розсіяним світлом, яка не створює відблисків на екрані. Важливо уникати прямого яскравого світла в очі чи на монітор.

Мікроклімат приміщення теж відіграє важливу роль: оптимальна температура становить 22–24 °С, а вологість — не нижче 40%. За необхідності слід користуватись зволожувачем або кондиціонером. Кабелі мають бути заховані або зафіксовані, щоб уникнути плутанини та небезпеки спотикання.

Крім того, для профілактики втоми рекомендується кожні 45–60 хвилин робити невеликі перерви — встати, пройтися, зробити кілька вправ для очей або змінити положення тіла. Це покращує самопочуття і сприяє ефективній роботі.

3.4 Основи пожежної безпеки при роботі з комп'ютерною технікою

Робота з комп'ютерною технікою хоч і не належить до особливо небезпечної, однак супроводжується певними ризиками, пов'язаними з пожежною безпекою. Причинами загорянь можуть бути коротке замикання, перегрів обладнання, несправні кабелі чи перевантаження електромережі.

Згідно з "Правилами пожежної безпеки в Україні" (НАПБ А.01.001-2014), у приміщеннях із електрообладнанням мають бути передбачені заходи захисту. Всі пристрої повинні бути справними, підключеними через заземлені розетки або сертифіковані подовжувачі з захистом від перенапруг. Не можна використовувати пошкоджені кабелі, саморобні зарядні пристрої або підключати кілька подовжувачів один до одного.

Системні блоки, монітори та інші пристрої мають бути розташовані так, щоб їхні вентиляційні отвори не були перекриті. Не можна накривати техніку тканиною, зберігати біля неї легкозаймисті матеріали або ставити предмети на

| | | | | | | |
|-----|------|----------|------|--------|--------------------------------|------|
| | | | | | <i>РП 08. 11 003. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Дата | Підпис | | 58 |

системні блоки. Після завершення роботи рекомендується повністю вимикати комп'ютер та інші пристрої з мережі.

У приміщенні має бути вогнегасник відповідного типу (порошковий або вуглекислотний) — наприклад, ВП-5 або ВВК-2. Його потрібно розмістити у доступному місці та регулярно перевіряти термін придатності. Також доцільно підготувати план евакуації із позначенням виходів, вогнегасника та дій у разі пожежі. Всі користувачі мають бути ознайомлені з цим планом і базовими діями при загорянні.

Не можна залишати ввімкнену техніку без нагляду на ніч або на тривалий час. У разі появи запаху гару, диму або іскор — обладнання слід негайно вимкнути з мережі та повідомити відповідальну особу. За наявності великої кількості техніки доцільно встановити автоматичну пожежну сигналізацію.

Таким чином, дотримання простих правил — справне обладнання, вимикання техніки, наявність вогнегасника та обережне поводження з електрикою — дозволяє значно знизити ризики та забезпечити пожежну безпеку навіть у звичайному комп'ютерному середовищі.

| | | | | | | |
|-----|------|----------|------|--------|--------------------------------|------|
| | | | | | <i>РП 08. 11 003. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Дата | Підпис | | 59 |

ВИСНОВКИ

Під час виконання дипломного проєкту було реалізовано ігрову систему, яка поєднує просту ідею вирощування овочів із поступовим розвитком механік, інтерфейсу та взаємодії з гравцем. Усе починалося зі створення базових функцій: посадка, ріст та збір урожаю. Але в процесі роботи над грою перетворилася на справжній багатосторонній проєкт, де довелося продумувати не лише технічну реалізацію, а й зручність, логіку, монетизацію і навіть візуальну атмосферу.

У грі були реалізовані: цикл росту овочів з анімаціями, система інвентаря, черга NPC-клієнтів, генерація замовлень, продаж товарів, накопичення грошей, поступове розблокування нових овочів і бонуси за перегляд реклами. Гравець бачить прогрес і має мотивацію розвиватися далі — що і було основною метою проєкту.

Особливу увагу було приділено тому, щоб зробити гру простою у сприйнятті, але цікавою у динаміці. Кожен новий елемент додавав відчуття руху вперед: спочатку нові грядки, потім покупці, замовлення, і, зрештою, заробіток. Завдяки цьому вдалося досягти ефекту “затягуючого” геймплею, який хочеться розвивати далі.

Робота над грою дозволила не тільки застосувати знання з програмування та дизайну, а й глибше зрозуміти, як працює ігрова логіка, UX-деталі, менеджмент сцен, збереження даних і синхронізація об’єктів. Це був корисний досвід повного циклу розробки — від ідеї до готової системи.

Отриманий результат може бути використаний як основа для подальшого розвитку гри: можна додати більше культур, нових покупців, покращити анімації або ввести декорації ферми. Гра має потенціал для мобільного запуску, а сам проєкт — як приклад реального застосування знань, отриманих під час навчання. Цей досвід став не просто навчальним, а й творчим процесом, який надихнув на ідеї для майбутніх проєктів.

| | | | | | | |
|-----|------|----------|------|--------|--------------------------------|------|
| | | | | | <i>РП 08. 11 003. 00 ДП ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Дата | Підпис | | 60 |

ВИКОРИСТАНІ ДЖЕРЕЛА

1. Unity Technologies. *Unity User Manual (2023)* [Електронний ресурс]. – Режим доступу: <https://docs.unity3d.com/Manual/index.html>
2. Microsoft. *C# Programming Guide* [Електронний ресурс]. – Режим доступу: <https://learn.microsoft.com/en-us/dotnet/csharp/>
3. GameDev Academy. *How to Make an Idle Clicker Game in Unity* [Електронний ресурс]. – Режим доступу: <https://gamedevacademy.org>
4. Іванов О.О. Розробка мобільних ігор у середовищі Unity: навчальний посібник. – Київ: Видавництво Ліра-К, 2021. – 176 с.
5. W3Schools. *HTML, CSS & JavaScript Tutorial* [Електронний ресурс]. – Режим доступу: <https://www.w3schools.com>
6. Nielsen, J. *Usability Engineering*. – Academic Press, 1994.
7. DMarket Blog. *Game Monetization Models Explained* [Електронний ресурс]. – Режим доступу: <https://dmarket.com/blog/game-monetization-models/>
8. Закон України «Про охорону праці» від 14.10.1992 № 2694-ХІІ.
9. ДСанПіН 3.3.2.007–98. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами ЕОМ.
10. НАПБ А.01.001-2014. *Правила пожежної безпеки в Україні*. – МВС України.
11. ДБН В.2.5-28:2018. *Природне і штучне освітлення*.
12. Український інститут охорони праці. *Методичні рекомендації до організації безпечного робочого місця користувача ПК*. – Київ, 2020.
13. Gamasutra. *Game Design: Principles, Practice, and Monetization Strategies* [Електронний ресурс]. – Режим доступу: <https://www.gamedeveloper.com>

| | | | | | | |
|-----|------|----------|------|--------|--------------------------------|------|
| | | | | | РП 08. 11 003. 00 ДП ПЗ | Арк. |
| Зм. | Арк. | № докум. | Дата | Підпис | | 61 |

ДОДАТОК А

Поведінка NPC при отриманні замовлення та русі сценою

```
using UnityEngine;
using System;

public class NPC_Move : MonoBehaviour
{
    public event Action onFinished;
    public Transform stopPoint1;
    public Transform stopPoint2;
    public Transform stopPoint3;
    public Transform stopPoint4;
    public float speed = 1.5f;
    [SerializeField] private OrderBubbleUI orderBubbleUI;
    [SerializeField] private OrderGenerator orderGenerator;
    [SerializeField] private MoneyStorage moneyStorage;
    private NPCOrder currentOrder;
    private InventoryManager inventoryManager;
    private int stage = 0;
    private Animator animator;
    private bool isWaiting = false;
    private bool orderAnnounced = false;
    private Transform player;

    void Start()
    {
        animator = GetComponent<Animator>();
        if (orderBubbleUI != null) orderBubbleUI.Hide();
        inventoryManager = InventoryManager.Instance;
        player = GameObject.FindGameObjectWithTag("Player").transform;
    }
}
```

```

void Update()
{
    if (isWaiting)
    {
        TrySellOrder();
        return;
    }

    Vector3 target = Vector3.zero;
    switch (stage)
    {
        case 0: target = new Vector3(stopPoint1.position.x, transform.position.y,
transform.position.z); break;
        case 1: target = new Vector3(transform.position.x, stopPoint2.position.y,
transform.position.z); break;
        case 2: target = new Vector3(transform.position.x, stopPoint3.position.y,
transform.position.z); break;
        case 3: target = new Vector3(stopPoint4.position.x, transform.position.y,
transform.position.z); break;
        default:
            animator.SetBool("ToIdle", true);
            animator.SetFloat("MoveX", 0);
            animator.SetFloat("MoveY", 0);
            onFinish?.Invoke();
            gameObject.SetActive(false);
            return;
    }

    Vector3 direction = target - transform.position;
    Vector3 moveDir = direction.normalized;

```

```

if (stage == 0 || stage == 3) moveDir.y = 0; else moveDir.x = 0;
animator.SetFloat("MoveX", moveDir.x);
animator.SetFloat("MoveY", moveDir.y);
animator.SetBool("ToIdle", false);
transform.position += moveDir * speed * Time.deltaTime;
if (direction.magnitude < 0.05f)
{
    animator.SetFloat("MoveX", 0);
    animator.SetFloat("MoveY", 0);
    animator.SetBool("ToIdle", true);
    if (stage == 1 && !orderAnnounced)
    {
        isWaiting = true;
        orderAnnounced = true;
        currentOrder = orderGenerator.GenerateOrder();
        if (currentOrder != null && orderBubbleUI != null)
            orderBubbleUI.ShowOrder(currentOrder, transform);
    }
    else stage++;
}
}
private void TrySellOrder()
{
    if (currentOrder == null || player == null || inventoryManager == null ||
moneyStorage == null) return;

    float distance = Vector2.Distance(player.position, transform.position);
    if (distance > 1.5f) return;

```

```

        if (inventoryManager.HasEnough(currentOrder.cropName,
currentOrder.quantity))
        {
            inventoryManager.RemoveVegetable(currentOrder.cropName,
currentOrder.quantity);
            moneyStorage.AddMoney(currentOrder.totalPrice);
            if (orderBubbleUI != null) orderBubbleUI.Hide();
            currentOrder = null;
            isWaiting = false;
            stage++;
        }
    }
    {
        public void ResetNPC()
        {
            stage = 0;
            orderAnnounced = false;
            isWaiting = false;
            currentOrder = null;
            if (orderBubbleUI != null) orderBubbleUI.Hide();
        }
    }
}

```

Клас CropDatabase — керування даними культур у грі

```

using System.Collections.Generic;
using UnityEngine;
public class CropDatabase : MonoBehaviour
{
    public static List<CropDefinition> crops;
    [Header("Список культур (редагується в інспекторі)")]
    public List<CropDefinition> cropList;
}

```

```

void Awake()
{
    crops = cropList;
}

public static CropDefinition GetByName(string name)
{
    foreach (var crop in crops)
    {
        if (crop.name == name)
            return crop;
    }
    return null;
}

public static Sprite GetIconByCropName(string name)
{
    CropDefinition crop = GetByName(name);
    return crop != null ? crop.icon : null;
}

public static int GetPriceByCropName(string name)
{
    CropDefinition crop = GetByName(name);
    return crop != null ? crop.price : 1;
}

public static List<CropDefinition> GetUnlockedCrops()
{
    List<CropDefinition> unlocked = new List<CropDefinition>();
    foreach (var crop in crops)
    {
        bool anyUnlocked = false;

```

```

foreach (var obj in crop.cropObjects)
{
    if (obj != null && obj.activeInHierarchy)
    {
        MonoBehaviour growth =
            obj.GetComponent<CropGrowth>() as MonoBehaviour ??
            obj.GetComponent<CarrotGrowth>() as MonoBehaviour ??
            obj.GetComponent<OnionGrowth>() as MonoBehaviour ??
            obj.GetComponent<TomatoGrowth>() as MonoBehaviour ??
            obj.GetComponent<CabbageGrowth>() as MonoBehaviour ??
            obj.GetComponent<PotatoGrowth>() as MonoBehaviour ??
            obj.GetComponent<StrawberryGrowth>() as MonoBehaviour ??
            obj.GetComponent<EggplantGrowth>() as MonoBehaviour ??
            obj.GetComponent<ChiliPepperGrowth>() as MonoBehaviour ??
            obj.GetComponent<YellowPepperGrowth>() as MonoBehaviour ??
            obj.GetComponent<BlackberryGrowth>() as MonoBehaviour ??
            obj.GetComponent<GrapeGrowth>() as MonoBehaviour;
        if (growth != null)
        {
            var type = growth.GetType();
            var isUnlockedProp = type.GetField("isUnlocked");
            if (isUnlockedProp != null)
            {
                bool isUnlocked = (bool)isUnlockedProp.GetValue(growth);
                if (isUnlocked)
                {
                    Debug.Log("Овоч '" + crop.name + "' разблокировано через
об'єкт: " + obj.name);
                    anyUnlocked = true;
                }
            }
        }
    }
}

```

```

        break;
    }
}
}
}
}
if (anyUnlocked)
{
    unlocked.Add(crop);
}
}
return unlocked;
}
}

```

Клас OrderGenerator — створення замовлення для NPC

```

using UnityEngine;
using System.Collections.Generic;
public class OrderGenerator : MonoBehaviour
{
    [Header("База даних культур")]
    public CropDatabase cropDatabase;
    [Header("Кількість одиниць в одному замовленні (мін/макс)")]
    public int minQuantity = 1;
    public int maxQuantity = 6;
    public NPCOrder GenerateOrder()
    {
        List<CropDefinition> availableCrops = cropDatabase.GetUnlockedCrops();
        if (availableCrops == null || availableCrops.Count == 0)

```

```

{
    Debug.LogWarning("Немає розблокованих овочів для замовлення!");
    return null;
}
foreach (var crop in availableCrops)
{
    Debug.Log("Доступний для замовлення: " + crop.name);
}
CropDefinition selectedCrop = availableCrops[Random.Range(0,
availableCrops.Count)];
int quantity = Random.Range(minQuantity, maxQuantity + 1);
return new NPCOrder(
    selectedCrop.name,
    quantity,
    selectedCrop.price,
    selectedCrop.icon
);
}
}

```

Дипломний проєкт

Розробка мобільної гри з монетизацією через інтеграцію реклами

Дипломний проєкт зі спеціальності 121 «Інженерія програмного забезпечення»

Виконала:

студентка групи РП-08

Жмайло Анна

Керівник ДП: Кунуп Т.В.

Мета та основні завдання дипломного проєкту

Мета проєкту:

Розробити мобільну гру у жанрі фермерського симулятора, де гравець вирощує овочі, продає їх клієнтам і отримує бонуси за перегляд реклами. Головна ціль — створити просту, але захоплюючу гру з можливістю розвитку та монетизації.

Основні завдання:

- Створити логіку росту овочів, збору врожаю та взаємодії з грядками.
- Реалізувати NPC, які роблять замовлення та реагують на дії гравця.
- Побудувати зручний інтерфейс інвентаря та відображення ресурсів.
- Налаштувати механіку перегляду реклами з ігровими бонусами.
- Забезпечити приємний візуальний стиль та стабільну роботу гри.



Слайд 3

Актуальність теми

Сьогодні мобільні ігри залишаються одним з найпопулярніших напрямів у сфері розробки програмного забезпечення. Особливо успішними є проєкти з простим геймплеєм, які легко монетизуються за допомогою вбудованої реклами.

Жанр фермерських симуляторів залишається затребуваним завдяки простоті, наочності та постійному відчуттю прогресу у гравця. Такі ігри дозволяють розслабитися, але водночас втягують у процес розвитку — що ідеально підходить для мобільної платформи.

Об'єднання зрозумілої гри та механіки монетизації через рекламу робить тему проєкту не лише актуальною, а й практично корисною — як приклад реального застосування знань у сучасному IT-середовищі.

Слайд 4

Аналіз ринку та аналогів

Перед розробкою було проаналізовано популярні мобільні ігри у жанрі фермерських симуляторів. Найбільш відомими прикладами є:

Hay Day — гра з детальною графікою, великою кількістю функцій та мікротранзакціями.

Idle Farming Empire — проста, мінімалістична гра з автофермою та акцентом на пасивний дохід.

Green Farm 3, FarmVille 2 — класичні приклади з упором на естетику, спілкування з іншими гравцями.

Наша гра поєднує елементи, властиві цим проєктам: збирання врожаю, взаємодія з клієнтами, відчуття прогресу. Водночас вона залишається простою та швидкою у геймплеї, без зайвих складних систем.

Також було враховано сучасні підходи до монетизації, де основний дохід отримується через добровільний перегляд реклами, а не через тиск на гравця.



Слайд 5

Концепція гри

Проект — це мобільна 2D-гра у жанрі фермерського симулятора з монетизацією через рекламу. Гравець отримує базову ділянку з грядками, вирощує овочі, збирає врожай, обслуговує клієнтів і поступово розвиває свою ферму.

Ключові ідеї гри:

- Вирощування різних видів овочів із візуальними стадіями росту.
- Збір врожаю одним дотиком після появи іконки готовності.
- З'явлення NPC-клієнтів, які роблять замовлення на певні овочі.
- Продаж продукції — основне джерело прибутку гравця.
- Можливість перегляду реклами для отримання бонусів (x2 дохід).
- Прогрес — відкриття нових грядок, культур, збільшення кількості клієнтів.

Гра побудована на простому, але затягуючому геймплеї з елементами автоматизації, візуального задоволення і відчуття зростання.



Слайд 6

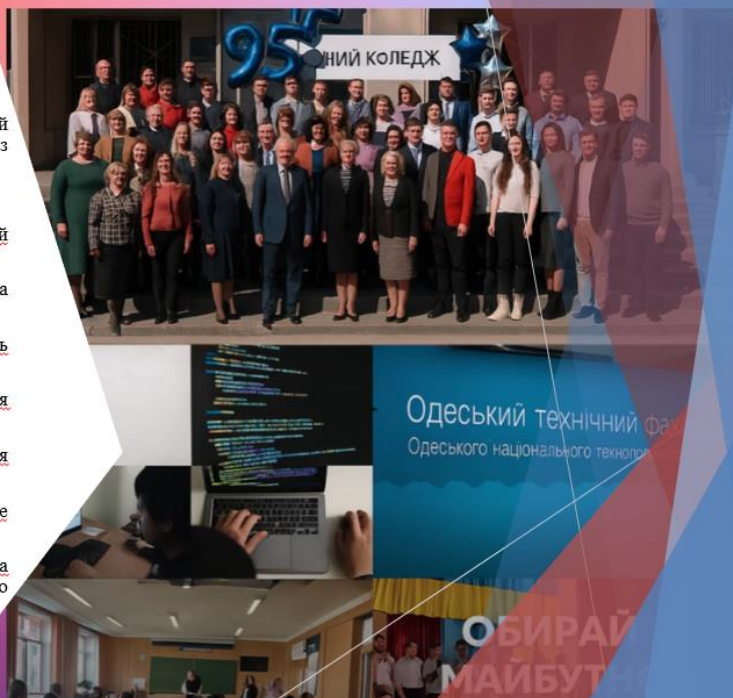
Основні механіки гри

Гра реалізована як цикл дій, що повторюється й поступово ускладнюється. Кожна механіка пов'язана з розвитком і прибутком гравця.

Впроваджені механіки:

- **Ріст овочів** — кожна культура проходить кілька стадій росту з таймером.
- **Збір урожаю** — автоматичний збір при натисканні на іконку, що з'являється після дозрівання.
- **NPC-клієнти** — з'являються по черзі, роблять замовлення на конкретні овочі.
- **Продаж овочів** — перевірка замовлення, списання продуктів з інвентаря, додавання грошей.
- **Перегляд реклами** — за бажанням гравця активується x2 дохід від наступного продажу.
- **Розблокування нових культур** — поступове відкриття грядок за зароблені гроші.

Ці механіки працюють разом як єдина ігрова система, забезпечуючи динаміку, прогрес і мотивацію до подальшої гри.



Слайд 7

```
foreach (var crop in crops)
{
    bool anyUnlocked = false;
    foreach (var obj in crop.plantObjects)
    {
        if (obj != null && obj.activeInHierarchy)
        {
            bool isUnlocked = (bool)isUnlockedProp.GetValue(growth);
            if (isUnlocked)
            {
                // ...
            }
        }
    }
}
```

Структура проєкту в Unity

Гра побудована у середовищі Unity з використанням мови програмування C#. Весь проєкт розділений на окремі функціональні компоненти, що взаємодіють між собою.

Ключові елементи структури:

- **CropManager** — керує ростом овочів на кожній грядці.
- **InventoryManager** — зберігає дані про зібраний врожай.
- **NPCManager** — відповідає за появу та черговість NPC.
- **OrderGenerator** — створює замовлення на овочі з урахуванням доступних культур.
- **MoneyStorage** — обробляє фінансову частину гри.
- **AdManager** — активує бонуси після перегляду реклами.
- **UIManager** — показує інвентар, замовлення, кількість грошей тощо.

Кожен скрипт відповідає лише за свою частину логіки, що полегшує підтримку й розширення гри.

```
bool isUnlocked = (bool)isUnlockedProp.GetValue(growth);
if (isUnlocked)
{
    // ...
}
```

Слайд 8

Інтерфейс гри (UI)

Інтерфейс розроблений таким чином, щоб бути простим, зрозумілим та візуально привабливим для гравця. Усі елементи розташовані логічно та не заважають ігровому процесу, дозволяючи зосередитись на основних діях.

Гравець бачить:

- **Інвентар**, який відкривається через скриню на сцені. Він показує всі типи овочів, навіть ті, що ще не зібрані, з кількістю біля кожного. Це допомагає легко орієнтуватись у запасах.
- **Лічильник грошей** — завжди розміщений у верхній частині екрана. Оновлюється миттєво після кожного продажу або перегляду реклами.
- **Замовлення NPC** — виводиться прямо над головою персонажа у вигляді іконки овоча з кількістю. Це дозволяє одразу зрозуміти, що потрібно клієнту.
- **Іконка дозрілого врожаю** — з'являється над грядкою, як тільки овоч готовий до збору. Гравець може підійти і зібрати його одним дотиком.
- **Кнопка перегляду реклами** — з'являється після таймера. Натискання активує бонус на наступний продаж.

Колірна палітра інтерфейсу витримана в теплих фермерських тонах, а всі елементи оформлені у стилі 2D-ілюстрацій, що відповідають загальній атмосфері гри.

Слайд 9



Монетизація через рекламу

Однією з ключових особливостей гри є система добровільної монетизації через інтеграцію реклами. Вона побудована так, щоб не заважати гравцеві, а навпаки — бути для нього вигідною.

Як це працює:

- На екрані з'являється кнопка перегляду реклами, коли таймер активує можливість бонусу.
- Гравець сам вирішує, чи дивитися відео. Примусу немає.
- Після перегляду короткого рекламного ролика гравець отримує **x2 дохід** від наступного продажу.
- Бонус працює лише один раз — це мотивує гравця використовувати рекламу стратегічно.

Такий підхід є зручним як для гравця, так і для розробника. З одного боку, це **чесна форма монетизації**, яка не впливає негативно на геймплей. З іншого — вона дозволяє отримувати дохід від активної аудиторії, не створюючи бар'єрів у грі.

Слайд 10

Охорона праці при роботі над проєктом

Розробка гри проходила в умовах тривалої роботи за комп'ютером, тому важливо було врахувати основні вимоги охорони праці.

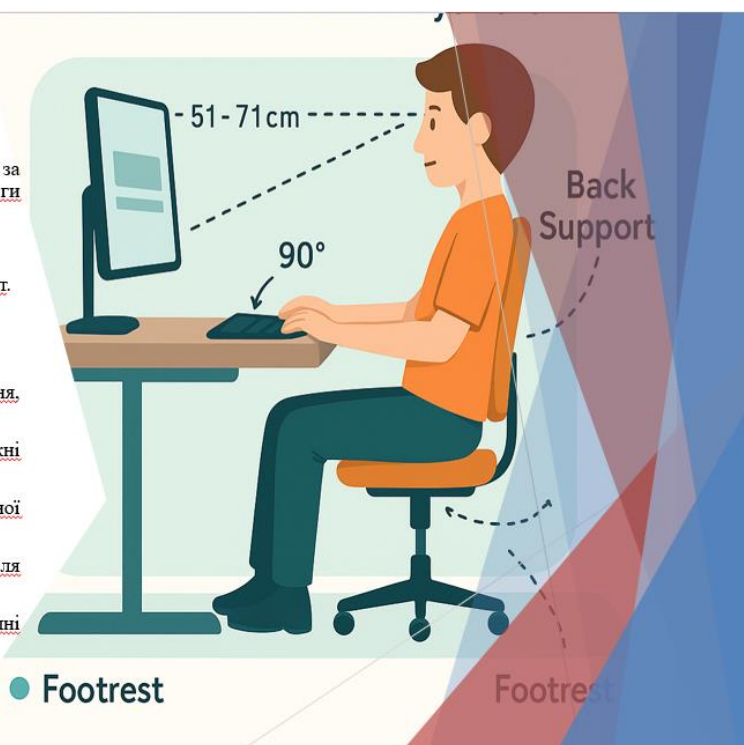
Аналіз умов праці показав:

- Можливі навантаження на зір та опорно-руховий апарат.
- Ризики перегріву обладнання та короткого замикання.

Реалізовані заходи безпеки:

- Робоче місце організоване з урахуванням освітлення, мікроклімату та ергономіки.
- Дотримання режиму праці та відпочинку: перерви кожні 45–60 хвилин.
- Застосування фільтрів напруги та справної електромережі.
- Пожежна безпека: відсутність займистих матеріалів біля обладнання, вимкнення техніки після роботи.

Такі дії дозволили створити комфортні й безпечні умови праці під час реалізації дипломного проєкту.



Слайд 11

Результати проекту

У межах дипломної роботи було створено повноцінну мобільну гру, яка поєднує просту ідею фермерства з поступовим розвитком та монетизацією через рекламу.

Досягнуті результати:

- Реалізовано механіку росту, збору та продажу 12 видів овочів.
- Побудовано систему NPC-клієнтів, які роблять замовлення і реагують на дії гравця.
- Створено інтерфейс інвентаря, замовлень, грошей та кнопки реклами.
- Налаштовано логіку монетизації через перегляд відео з бонусами.
- Організовано послідовне відкриття нових культур через накопичення грошей.
- Гра стабільно працює, підтримує прогрес і забезпечує гнучкий розвиток.

Проект повністю відповідає поставленій меті та демонструє практичне застосування навичок з програмування, дизайну, логіки геймплею й аналізу користувачького досвіду.

Слайд 12

Перспективи розвитку



Проект заклав міцну основу для гри, яка може рости далі разом із гравцем. У вже реалізовану механіку легко додати нові функції, не порушуючи цілісності геймплею.

Подальші ідеї:

- Додати нові культури, які відкриватимуться на вищих рівнях.
- Запровадити покращення ферми — декоративні об'єкти, бонуси, апгрейди.
- Додати більше типів NPC з різною поведінкою та замовленнями.
- Удосконалити візуальну частину: анімації, ефекти, саундтрек.
- Зробити внутрішній магазин з покупками за ігрову або реальну валюту.
- Підготувати гру до публікації в Google Play або інших платформах.

Гра вже має всі базові компоненти, працює стабільно, і головне — її приємно тестувати. Це означає, що вона готова до розвитку, оновлень і навіть комерційного використання.

РЕЦЕНЗІЯ

на дипломний проект (роботу) здобувача (здобувачки) освіти
відділення комп'ютерних систем

Жмайло Анни Анатоліївни

(прізвище, ім'я та по батькові)

Спеціальність 121 Інженерія програмного забезпечення

Освітньо-професійна програма Розробка програмного забезпечення

Керівник дипломного проекту (роботи) Кунун Тетяна Василівна

(прізвище, ім'я та по батькові)

Тема дипломного проекту (роботи) Розробка мобільної гри з монетизацією через інтеграцію реклами

Обсяг розрахунково-пояснювальної записки 75 сторінок

Обсяг графічної (презентаційної) частини 13 аркушів (слайдів)

ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЕКТУ (РОБОТИ)

а) заключення про ступінь відповідності виконаного дипломного проекту (роботи) завданню

Дипломний проект повністю відповідає поставленому завданню. У роботі реалізовано всі основні функції: геймплей, система замовлень, збір і продаж врожаю, а також монетизація через рекламу. Проект виконано згідно з вимогами спеціальності та у встановлені терміни.

б) характеристика виконання кожного розділу дипломного проекту (роботи)

При виконанні дипломного проекту здобувач продемонстрував уміння застосовувати сучасні підходи до розробки, зокрема об'єктно-орієнтоване програмування, модульну архітектуру та інтеграцію реклами як засобу монетизації. Була ефективно використана ігрова платформа Unity, мова програмування C#, а також редактор анімацій, системи подій і серіалізовані дані для збереження стану гри.

в) оцінка якості виконання пояснювальної записки та графічної частини дипломного проекту

(роботи) Пояснювальна записка виконана, з дотриманням структури та вимог до оформлення. Графічна частина чітко відображає реалізовану функціональність гри та підтримує розкриття змісту проекту.

г) перелік позитивних якостей дипломного проекту (роботи) _____
Тема дипломного проекту є актуальною, виконана якісно та відповідно до поставленого завдання. Розроблено мобільну гру на Unity з внутрішньою економікою, системою замовлень та монетизацією через рекламу, що забезпечує гравцеві інтерес і прогрес у грі.

д) основні недоліки дипломного проекту (роботи) _____
Обмежена кількість рівнів складності, відсутність адаптації під різні платформи та невеликий обсяг графічних елементів власної розробки. Присутні помилки оформлення пояснювальної записки. Слабкий графічний матеріал проекту

| | |
|------------------------------|-------|
| Оцінка розрахункової частини | Добре |
| Оцінка графічної частини | Добре |
| Загальна оцінка | Добре |

Прізвище, ім'я, по батькові рецензента _____ *к.т.н. Шibaєва Наталя Олегівна*

Місце роботи і посада рецензента _____ *Національний університет «Одеська політехніка», доцент кафедри інформаційних технологій*

Підпис: _____

« 23 » _____ 2025 р.



ВІДГУК

керівника на дипломний проєкт здобувача (здобувачки) освіти
відділення комп'ютерних систем

Жмайло Анни Анатоліївни

(прізвище, ім'я та по батькові)

Спеціальність: 121 "Інженерія програмного забезпечення"

Освітньо-професійна програма: «Розробка програмного забезпечення»

Тема дипломного проєкту: Розробка мобільної гри з монетизацією через
інтеграцію реклами

ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЄКТУ

а) обсяг і якість виконання проєкту (графічного матеріалу і розрахунково-пояснювальної записки) Дипломний проєкт виконано відповідно технічному завданню. Пояснювальна записка містить 75 сторінки. У пояснювальній записці виконано опис етапів, відповідно до індивідуального завдання, розділи пояснювальної записки відповідають етапам рішення завдання, поставленого у дипломному проєкті. Графічна частина складається з 13 слайдів мультимедійної презентації. Якість виконання пояснювальної записки та графічної частини добра, розробку виконано в повному обсязі.

б) самостійність роботи над проєктом: Протягом всього строку дипломного проєктування та переддипломної практики здобувачка освіти Жмайло Анна Анатоліївна, поступово та послідовно виконувала всі етапи розробки. Всі роботи здобувачка освіти виконувала самостійно, з оглядом на рекомендації керівника

в) теоретична підготовка випускника (випускниці): Здобувачка освіти Жмайло Анна Анатоліївна під час роботи над дипломним проєктом вивчила достатню кількість літературних джерел та матеріалів за даною тематикою.

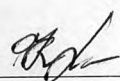
Вважаю, що теоретична підготовка дипломника добра і він готовий до захисту дипломного проєкту

г) вміння розв'язувати виробничі та конструкторські питання _____
Під час дипломного проектування здобувачка освіти Жмайло А.А. розробила програмний продукт, а саме idle-ферму з монетизацією через рекламу. Дипломниця володіє навичками написання програмного продукту за допомогою таких технологій як: C#, Unity.

Оцінка розрахункової частини _____ *Відмінно*
Оцінка графічної частини _____ *Відмінно*
Загальна оцінка _____ *Відмінно*

Прізвище, ім'я, по батькові керівника дипломного проекту _____
Кунун Тетяна Василівна

Місце роботи і посада керівника дипломного проекту _____
НУ "Одеська політехніка" к.т.н. ст. викладач кафедри "Інформаційні технології".

Підпис _____ 

« 16 » 06 2025 р.

**ДОЗВІЛ
НА РОЗМІЩЕННЯ
ВИПУСКНОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ
(ДИПЛОМНОГО ПРОЄКТУ)
В ЕЛЕКТРОННОМУ РЕПОЗИТАРІЇ ВСП «ОТФК ОНТУ»**

Ми, що нижче підписалися,

Жмайло Анна Анатоліївна
здобувачки освіти гр. 4РП-08, та

Кунуп Тетяна Василівна,
керівник дипломного проекту,

не заперечуємо щодо розміщення електронного варіанту пояснювальної записки до дипломного проекту фахового молодшого бакалавра на тему:

**«Розробка мобільної гри з монетизацією через інтеграцію реклами»
(автор роботи – Жмайло А.А., керівник роботи – Кунуп Т.В.)**

виконаного у ВСП «Одеський технічний фаховий коледж Одеського національного технологічного університету» в 2025 році, у повному обсязі в електронному репозитарії ВСП «ОТФК ОНТУ» для вільного доступу через мережу Інтернет.

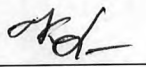
Несемо відповідальність за ідентичність електронного та друкованого варіантів випускної кваліфікаційної роботи і даємо згоду на обробку персональних даних.

Виконавець



/ Жмайло А.А. /

Керівник



/ Кунуп Т.В. /

«16» червня 2025 р.

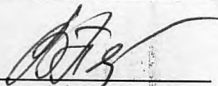
ДОВІДКА

циклової комісії КТ та ПІ
про допуск до захисту дипломного проєкту
здобувача (здобувачки) освіти ІV курсу
відділення комп'ютерних систем групи 4РП-08

Жмайло Анни Анатоліївни

на тему Розробка мобільної гри
з монетизацією через інтеграцію реклами

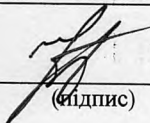
Висновок відповідальної особи за проведення нормоконтролю:
пояснювальна записка до дипломного проєкту виконана з деякими
порушеннями ДСТУ та оформлена відповідно до вимог Положення про
дипломне проєктування


(підпис)

18.06.2025
(дата)

Петрашова В.І.
(П.І.Б.)

Висновок відповідальної особи за перевірку роботи на наявність академічного
плагіату згідно звіту про перевірку від 16.06.2025 р. значення коефіцієнту
подібності в роботі становить 6,59%, коефіцієнт цитування – 0,70%.


(підпис)

18.06.2025
(дата)

Краснокутська К.Г.
(П.І.Б.)

Попередня експертиза (малий захист) дипломного проєкту

здобувача (здобувачки) освіти

Жмайло А.А.
(П.І.Б.)

проведена « 18 » червня 2025 р.

Висновки Пояснювальна записка до дипломного проєкту виконана у повному
обсязі. Випускна кваліфікаційна робота (дипломний проєкт) відповідає
вимогам Положення про дипломне проєктування та рекомендована до
захисту.

Голова ЦК КТ та ПІ


(підпис)

Кривченко Ю.В.
(П.І.Б.)

Звіт подібності

метадані

Назва організації

Odesa Technical Professional College of Odesa National University of Technology

Заголовок

Розробка мобільної гри з монетизацією через інтеграцію реклами

Автор

Науковий керівник / Експерт

Жмайло Анна Анатоліївна Кунуп Тетяна Василівна

підрозділ

Відокремлений структурний підрозділ "Одеський технічний фаховий коледж Одеського національного технологічного університету"

Обсяг знайдених подібностей

Коефіцієнт подібності визначає, який відсоток тексту по відношенню до загального обсягу тексту було знайдено в різних джерелах. Зверніть увагу, що високі значення коефіцієнта не автоматично означають плагіат. Звіт має аналізувати компетентна / уповноважена особа.



25

Довжина фраз для коефіцієнта подібності 2



11794

Кількість слів

93171

Кількість символів

Тривога

У цьому розділі ви знайдете інформацію щодо текстових спотворень. Ці спотворення в тексті можуть говорити про МОЖЛИВІ маніпуляції в тексті. Спотворення в тексті можуть мати навмисний характер, але частіше характер технічних помилок при конвертації документа та його збереженні, тому ми рекомендуємо вам підходити до аналізу цього модуля відповідально. У разі виникнення запитань, просимо звертатися до нашої служби підтримки.

| | | |
|------------------------|--|----|
| Заміна букв | | 20 |
| Інтервали | | 0 |
| Мікропробіли | | 21 |
| Білі знаки | | 0 |
| Парафрази (SmartMarks) | | 42 |

Подібності за списком джерел

Нижче наведений список джерел. В цьому списку є джерела із різних баз даних. Колір тексту означає в якому джерелі він був знайдений. Ці джерела і значення коефіцієнту подібності не відображають прямого плагіату. Найкраще відкрити кожне джерело і проаналзувати зміст і порівняти з отриманим джерелом.

10 найдовших фраз

| ПОРЯДКОВИЙ НОМЕР | НАЗВА ТА АДРЕСА ДЖЕРЕЛА URL (НАЗВА БАЗИ) | Кількість такти |
|---------------------|---|---|
| | | КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ) |
| 1 | https://card-file.ontu.edu.ua/server/api/core/bitstreams/a141b858-51a7-4f90-b0bd-7f0ccaed21e5/content | 51 0.43 % |
| 2 | https://card-file.ontu.edu.ua/bitstreams/82a5d375-2b69-4233-b80f-bf1a149b7747/download | 37 0.31 % |
| 3 | https://card-file.ontu.edu.ua/bitstreams/035f6436-20b4-4ee6-8e99-bcde670e308b/download | 30 0.25 % |
| 4 | https://card-file.ontu.edu.ua/server/api/core/bitstreams/a141b858-51a7-4f90-b0bd-7f0ccaed21e5/content | 29 0.25 % |
| 5 | https://card-file.ontu.edu.ua/bitstreams/035f6436-20b4-4ee6-8e99-bcde670e308b/download | 28 0.24 % |

| | | |
|----|---|-----------|
| 6 | https://card-file.ontu.edu.ua/server/api/core/bitstreams/a141b658-5fa7-4f90-b0bd-7f0ccaed21e5/content | 26 0.22 % |
| 7 | https://card-file.ontu.edu.ua/bitstreams/bbed74c8-2ea7-44c5-8d00-0fe3fd9790ee/download | 24 0.20 % |
| 8 | https://card-file.ontu.edu.ua/server/api/core/bitstreams/a141b658-5fa7-4f90-b0bd-7f0ccaed21e5/content | 23 0.20 % |
| 9 | https://card-file.ontu.edu.ua/bitstreams/bbaf3f38-16e8-4070-bead-5562769b7c71/download | 23 0.20 % |
| 10 | https://card-file.ontu.edu.ua/bitstreams/82a6d375-2b69-4233-b80f-fbfd149b7747/download | 23 0.20 % |

з домашньої бази даних (0.54 %)

| порядковий номер | заголовок | кількість ідентичних слів (фрагментів) |
|------------------|--|--|
| 1 | Розробка 3D-гри у жанрі survival-horror з налаштуваннями рівнів складності 8/12/2025 Odessa Technical Professional College of Odessa National University of Technology (Відокремлений структурний підрозділ "Одеський технічний фаховий коледж Одеського національного технологічного університету") | 36 (4) 0.31 % |
| 2 | Розробка системи авторизації користувача на web-сервері за допомогою pdf-модулю 8/15/2025 Odessa Technical Professional College of Odessa National University of Technology (Відокремлений структурний підрозділ "Одеський технічний фаховий коледж Одеського національного технологічного університету") | 16 (2) 0.14 % |
| 3 | Розробка web-застосунку для генерації повідомлень із використанням технологій штучного інтелекту 8/14/2025 Odessa Technical Professional College of Odessa National University of Technology (Відокремлений структурний підрозділ "Одеський технічний фаховий коледж Одеського національного технологічного університету") | 12 (2) 0.10 % |

з програми обміну базами даних (0.11 %)

| порядковий номер | заголовок | кількість ідентичних слів (фрагментів) |
|------------------|--|--|
| 1 | Вплив температури розпаду переохолодженого аустеніту на структуру і властивості низьковуглецевих сталей 8/21/2024 Dniprovsk State Technical University (Фізика конденсованого стану) | 13 (1) 0.11 % |

з Інтернету (5.94 %)

| порядковий номер | джерело URL | кількість ідентичних слів (фрагментів) |
|------------------|---|--|
| 1 | https://card-file.ontu.edu.ua/server/api/core/bitstreams/a141b658-5fa7-4f90-b0bd-7f0ccaed21e5/content | 356 (21) 3.02 % |
| 2 | https://card-file.ontu.edu.ua/bitstreams/82a6d375-2b69-4233-b80f-fbfd149b7747/download | 96 (5) 0.81 % |
| 3 | https://card-file.ontu.edu.ua/bitstreams/035f6436-20b4-4ee6-8e90-bede670e308b/download | 83 (4) 0.70 % |
| 4 | https://card-file.ontu.edu.ua/bitstreams/34a6756b-592f-4b77-a805-183aa03a6a26/download | 26 (2) 0.22 % |
| 5 | https://card-file.ontu.edu.ua/bitstreams/bbed74c8-2ea7-44c5-8d00-0fe3fd9790ee/download | 24 (1) 0.20 % |
| 6 | https://card-file.ontu.edu.ua/bitstreams/bbaf3f38-16e8-4070-bead-5562769b7c71/download | 23 (1) 0.20 % |
| 7 | https://card-file.ontu.edu.ua/bitstreams/6cf43324-8f08-4031-ba42-f90b15afbbc8/download | 21 (1) 0.18 % |
| 8 | https://card-file.ontu.edu.ua/server/api/core/bitstreams/8da72c29-65f1-4ee1-9b22-716dedf53f5/content | 19 (2) 0.16 % |

| | | |
|----|---|---------------|
| 9 | https://card-file.ontu.edu.ua/server/api/core/bitstreams/44c16132-5f53-4fe2-b6c0-61e9a2f0f375/content | 18 (2) 0.15 % |
| 10 | https://card-file.ontu.edu.ua/bitstreams/5240e379-7721-49f0-8ee8-27140b0a473a/download | 14 (2) 0.12 % |
| 11 | https://card-file.ontu.edu.ua/bitstreams/1df552d-7200-49b8-ae1d-ba76a1335685/download | 10 (1) 0.08 % |
| 12 | https://card-file.ontu.edu.ua/server/api/core/bitstreams/d5a3d14f-d5cb-420f-9c49-cba3f9d50554/content | 5 (1) 0.04 % |
| 13 | https://card-file.ontu.edu.ua/server/api/core/bitstreams/995bdcec-4e4d-4321-8070-4d6badcb8e49/content | 5 (1) 0.04 % |

Список прийнятих фрагментів (немає прийнятих фрагментів)

| ПОРЯДКОВИЙ НОМЕР | ЗМІСТ | КІЛЬКІСТЬ ОДНІЙКОВИХ ОБ'ЄКТІВ (ФРАГМЕНТІВ) |
|------------------|-------|--|
|------------------|-------|--|

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ВСП **ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАКУЛЬТЕТ КОЛЕДЖ ОНТУ**

Спеціальність: 121 **Інженер програмного забезпечення**

Освітньо-професійна програма: **Розробка програмних застосунків**

Група: 4РП-08

Дипломний проект

здобувача освіти денної форми навчання

РП.08.11.000.ДП

ЖМАЙЛО

АННИ АНАТОЛІВНИ

м. Одеса

2025 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ВСП **ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАКУЛЬТЕТ КОЛЕДЖ ОНТУ**

Спеціальність: 121 **Інженер програмного забезпечення**

Освітньо-професійна програма: **Розробка програмних застосунків**

Група: 4РП-08

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проекту на тему:

Проектний матеріал складається з пояснювальної записки на _____ сторінках та графічного (презентаційного) матеріалу на _____ аркушах (слайдах)

Дипломник _____ (Жмайло А. А.)

Керівник _____ (Кунуп Т.В.)

Консультанти:

з економічного розділу _____ (Канський М.Ю)

з розділу охорони праці та техніки безпеки _____ (Чорновіл Н.І.) д. нормоконтролю

_____ (Петрашова В.І.) старший консультант _____ (Кривченко Ю. В.) До

захисту допущений Голова циклової комісії _____ (Кривченко Ю. В.) Звідувач відділенням

_____ (Краснокутська К.Г.)

Захист «_____» _____ 2025 р. Протокол ЕК № _____

Оцінка ЕК _____

Секретар ЕК _____

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ВСП **ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАКУЛЬТЕТ КОЛЕДЖ ОНТУ**

Відділення _____ Комісія _____