

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)


Кафедра Медіасистем та технологій
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

Розробка графічного стилю мобільної гри
гіперказуального жанру «Fruit and Juice»
(тема)

Виконав:
студент 4 курсу, групи ВПВПС-18-2

 Зуєвський Д.Р.

Спеціальності _____ 186 Ви

Освітня програма
Видавничо-поліграфічна справа

Керівник  проф. Кулішова Н. Є.

Допускається до захисту
Зав. кафедри МСТ

(підпис) Дейнеко Ж.В.
(прізвище, ініціали)

2022 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
 Кафедра Медіасистем та технологій
 Рівень вищої освіти перший (бакалаврський)
 Спеціальність 186 Видавництво та поліграфія
 Тип програми Освітньо-професійна
 Освітня програма Видавничо-поліграфічна справа
 (шифр і назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри МСТ _____
(підпис)

« 23 » травня 2022 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Зуєвському Дмитру Романовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка графічного стилю мобільної гри
гіперказуального жанру «Fruit and Juice»

Затверджена наказом по університету від 21 травня 2022 р. № 558 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 16 червня 2022 р

3. Вихідні дані до роботи

Мови розробки: C#; середовище розробки – Unity, платформа розповсюдження — Play Market.


4. Перелік питань, що потрібно опрацювати в роботі

Вступ; Аналіз завдання на кваліфікаційну роботу; Аналітичний огляд літератури; Проектування технологічного процесу виготовлення електронного видання; Вибір інструментальних (програмних і технічних) засобів розробки; Проектування інформаційної структури електронного видання складових фірмового стилю; Розробка графічного дизайну електронного видання; Тестування і публікація електронного видання; Тестування і публікація; Економічна частина; Висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п. 5 включається до завдання за рішенням випускової кафедри)

Мета; Аналіз завдання; Особливості ігрового жанру; Структура виробничого процесу; Підготовка до розробки; Розробка та експорт ресурсів; Налаштування графічних матеріалів; Розробка супроводжувальних матеріалів; Архівування ресурсів; Вибір програмних та технічних засобів розробки; Розробка візуального стилю; Тестування; Економічна частина; Висновки.

6. Консультанти розділів роботи (п. 6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п. 1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Основна частина	проф. Кулішова Н. Є.		
Економічна частина	проф. Полозова Т.В.		04.06

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз завдання на кваліфікаційну роботу	23.05	виконано
2	Аналітичний огляд літератури		виконано
3	Проектування технологічного процесу виготовлення електронного видання		виконано
4	Вибір програмних засобів розробки		виконано
5	Вибір технічних засобів розробки		виконано
6	Проектування інформаційної структури електронного видання, складових фірмового стилю		виконано
7	Розробка графічного дизайну електронного видання		виконано
8	Тестування		виконано
9	Економічне обґрунтування роботи		виконано
10	Оформлення пояснювальної записки		виконано
11	Оформлення графічної частини		виконано
12	Попередній захист роботи		
13	Захист роботи	17.06	


Дата видачі завдання 23 травня 2022 р.

Студент


(підпис)

Зуєвський Д.Р.

Керівник роботи


(підпис)

проф. Кулішова Н. Є.
(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи містить: 70 стор., 9 ч., 5 табл., 32 рис., 10 джерел.

МОБІЛЬНА ГРА, РОЗРОБКА ГРИ, ОПТИМІЗАЦІЯ, ВІЗУАЛЬНИЙ СТИЛЬ, ТЕСТУВАННЯ.

Метою роботи є розробка мобільної гри гіперказуального жанру «Fruit and Juice», створення оригінального наповнення та візуального стилю. Описується процес розробки графічного наповнення, та його експорт та налаштування в середовищі розробки.

Під час виконання було проаналізовано особливості жанру, розроблено основні вимоги до графічного наповнення, описано структуру виробничого процесу та структуру проекту. Розроблені та налаштовані графічні ресурси, а також створено супутні матеріали для публікації.

Крім цього, було проведено економічне обґрунтування доцільності проведення даної кваліфікаційної роботи.

ABSTRACT

The explanatory note to the qualification work contains: 70 p., 9 pt., 8 tabl., 32 pic., 8 sources.

MOBILE GAME, GAME DEVELOPMENT, OPTIMIZATION, VISUAL STYLE, TESTING.

The work aims to develop a mobile game of the hyper-casual genre «Fruit and Juice», to create original content and visual style. Describes the process of developing graphic content, and its export and configuration in the development environment.

During the performance the peculiarities of the genre were analyzed, the basic requirements for the graphic content were developed, and the structure of the production process and the structure of the project were described. Graphic resources have been developed and configured, as well as related materials for publication.

In addition, an economic justification of the feasibility of this qualification work was conducted.

ЗМІСТ

	С.
ВСТУП.....	8
1 АНАЛІЗ ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ	9
2 АНАЛІТИЧНИЙ ОГЛЯД ЛІТЕРАТУРИ	12
2.1 Аудиторія гіперказуальних ігр	13
2.2 Аналіз аналогів	14
3 ПРОЕКТУВАННЯ ТЕХНОЛОГІЧНОГО ПРОЦЕСУ ВИГОТОВЛЕННЯ ЕЛЕКТРОННОГО ВИДАННЯ	17
4 ВИБІР ІНСТРУМЕНТАЛЬНИХ (ПРОГРАМНИХ І ТЕХНІЧНИХ) ЗАСОБІВ РОЗРОБКИ	26
4.1 Програмні засоби.....	26
4.2 Інструменти створення та редагування.....	29
4.3 Технічні засоби	31
5 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СТРУКТУРИ ЕЛЕКТРОННОГО ВИДАННЯ, СКЛАДОВИХ ФІРМОВОГО СТИЛЮ	33
5.1 Структура оригіналів.....	33
5.2 Структура супутніх матеріалів	36
5.3 Структура проекту Unity.....	36
5.4 Структура файлу оригіналу об'єктів для проекту Fruit and Juice.....	38
5.5 Структура проекту Fruit and Juice в Unity	40
6 РОЗРОБКА ГРАФІЧНОГО ДИЗАЙНУ ЕЛЕКТРОННОГО ВИДАННЯ	42
6.1 Стилiстичне рішення	42
6.2 Розробка моделей	44
6.3 Розгортка та використання колiрного атласу	46
6.4 Додаткові налаштування та експорт	49
6.5 Налаштування ресурсів в Unity	50
6.6 Промо-матеріали	53

7 ТЕСТУВАННЯ І ПУБЛІКАЦІЯ ЕЛЕКТРОННОГО ВИДАННЯ	56
7.1 Дослідницьке тестування	56
7.2 Ігрове тестування.....	58
7.3 Функціональне тестування	59
7.4 Технічне тестування	60
7.5 Публікація та установка	62
8 ЕКОНОМІЧНА ЧАСТИНА	65
ВИСНОВКИ	70
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	71

ВСТУП

Мобільна гра – це комплексний мультимедійний продукт, в якому поєднується велика кількість різних типів даних. Створення цих даних – складний багатоетапний процес. Графічне наповнення проекту є ключовим, і визначає першочергову привабливість проекту для користувача. Для швидкого та ефективного створення графічного наповнення потрібен чітко визначений виробничий процес, який дає змогу гнучко створювати наповнення, різне за типом, призначенням, графічним стилем. В процесі створення використовується широкий спектр програмного забезпечення, яке дозволяє роботу над різними типами даними, спрощує менеджмент задач та контенту.

Графічне наповнення має виконувати дві основні цілі: бути привабливим та звернути увагу користувача на себе, а також бути легкозрозумілим та легко зчитуваним для нього. Розроблений стиль має бути технічно оптимізованим для роботи на мобільних пристроях.

В основі розробленого стилю лежать колірні атласи, які використовуються як основна текстура розроблених моделей. Моделі стилізовані, не відображають фактуру та властивості поверхні, окрім кольору. Використання суцільного кольору робить модель занадто простою, тому для створення тіней використовуються градієнти там, де це можливо. Колірне рішення спрямоване на створення спокійної атмосфери, воно має звертати увагу гравця на об'єкти взаємодії.

1 АНАЛІЗ ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

В ході роботи розробляється графічне наповнення для мобільної гри гіперказуального жанру, виконується налаштування контенту в ігровому движку розробляються супроводжувальні матеріали для подальшої публікації на платформах розповсюдження.

Результатом роботи має бути створена та опублікована мобільна гра в гіперказуальному жанрі, яка може бути встановлена на Android пристроях з версією Android 5.0 та вище. Гра повинна правильно відображатися в будь-якій орієнтації на екранах будь-якого співвідношення сторін та роздільної здатності. Гра має бути також оптимізована для роботи на слабких та застарілих пристроях. Поширення гри виконуватиметься на платформі GooglePlay, і монетизуватиметься вона за рахунок показу реклами між ігровими сесіями або рівнями. Структура проекту має дозволяти легко оновлювати проект, а також доповняти новим контентом.

Напрямок гіперказуальних ігор спрямований на дуже широку аудиторію, це легко зрозуміла, невимоглива до вмінь гравця розвага, яка повинна дати змогу скоротити час будь де на будь-якому девайсі. Тематика таких ігор нейтральна, вона не заглиблюється у якусь специфічну діяльність. Ігрова механіка (правила гри, те що повинен робити гравець) також повинна бути простою і зрозумілою, часто це деяка повсякденна дія, яку перебільшують або доповнюють, перетворюючи на гру. Графічні рішення, насамперед, повинні бути легко зчитуваними, це несе дві цілі: першою є привернення уваги потенційної аудиторії при рекламуванні, другою – зрозумілість вже в грі, графіка не повинна відволікати або заважати грати. Вже додатково графічне рішення повинно бути індивідуальним та не нагадувати інші проекти. З технічної сторони від графічного наповнення залежать вимоги до пристрою, на якому гра буде запускатись та працювати стабільно. Це ставить

певні рамки до складності графічних рішень, які можуть використовуватись в грі.

Для такого роду проектів, окрім основного їх наповнення, створюється ще ряд графічних матеріалів, які використовуються на платформах публікації або при рекламі продукту. Мінімальним набором є іконка та постер для гри, інколи створюється індивідуальне оформлення знімків самої гри. Для всіх матеріалів також першою вимогою є легкозчитуваність та зрозумілість. Монетизація виконується за рахунок показу реклами між ігровими сесіями, або рівнями, тому важливо, скільки часу користувач проводить в грі. Через це графіка не повинна напружувати користувача та його зір, дуже контрастні та різкі кольори привертають увагу в рекламі, але в грі можуть заважати та зменшувати комфорт під час гри.

Аудиторія максимально широка і не може бути об'єднана в деяку конкретну категорію, це розвага для всіх, кого вона може зацікавити механікою гри. Це обмежує тематику гри, вона повинна бути зрозумілою всім, це обмежує і графічне наповнення – воно не може бути сильно стилізованим або абстрактним, так як це зменшує привабливість для середньостатистичного користувача.

Основні графічні рішення, використані в грі, спрямовані саме на спрощення зчитування та зрозумілість – проста стилізація об'єктів без значного перебільшення їх характеристик, з загальноприйнятими пропорціями, формою та кольорами. Більшість об'єктів позбавлена характерних текстур, вони змінені на більш прості в зчитуванні градієнти. Колірне рішення спокійне в пастельних тонах; акцент зроблений на об'єктах взаємодії, вони більш насичені, ніж оточення, що разом з анімацією привертає увагу гравця. У відповідь на дії гравця використовуються чітко зрозумілі ефекти, які мають інформувати про успішність дій. Окрім ефектів, прогрес в грі ілюструється інтерфейсом. Інтерфейс не нав'язливий і під час самої гри не має привертати увагу. Вже після завершення рівня він виходить на перший план, надає результати та показує нагороду.

З технічного боку графіка оптимізована для слабких девайсів. Засоби, що використовуються для цього – це і невелика кількість трикутників і оптимізація розміру текстур та кількість необхідних текстур матеріалів. Технологічне рішення включає в себе швидке редагування колірних оформлень всіх об'єктів, це дозволяє змінювати оформлення для запису реклами або оновлення версій. Також технічно графіка легко оновлюється та доповнюється, що також необхідно для подальшої підтримки проекту.

2 АНАЛІТИЧНИЙ ОГЛЯД ЛІТЕРАТУРИ

Гіперказуальні ігри – це мобільні ігри, з простою механікою (правилам, цілями гри) ігрова сесія коротка, але має затягувати гравця, при цьому розробка зазвичай значно коротша порівнянно з іншими ігровими проектами. Такі ігри мають популярність за свою простоту та захопливість.

На відміну від інших мобільних ігор, на створення, концептуалізацію та прототип яких можуть піти місяці, ці ігри можна випустити на ринок за короткий проміжок часу. Але випуск не означає повне завершення роботи над грою, часто вона змінюється для збільшення кчлькосты користувачів.

Гіперказуальні ігри концептуально схожі на найперші аркадні ігри, які з'явилися у 70х – 80х роках. Їх особливість – в простоті та циклічності механік. В основі таких ігор лежать одна чи дві механіки. Візуально вони мають бути простими і зрозумілими. Графіка яскрава та насичена без зайвих деталей і акцентує увагу на ігрових елементах, об'єктах. Інтерфейс також візуально простий, через простоту ігрових механік логіка навігації однотипна, і в різних проектах повторюється.

Як окремий жанр, гіперказуальні ігри не виділяють на платформах розповсюдження, об'єднуючи з більшими за складністю проектами. Найчастіше гіперказуальні ігри об'єднують з гонками та головоломками.

Найуспішніші гіперказуальні ігри стають вірусними, оскільки викликають звикання, ігрові сесії короткі, і їх легко зрозуміти. Це означає, що користувачі можуть грати в них, щоб зняти стрес і скоротити час.

У середньому час, який користувачі витрачають на ці ігри за день, на 45 відсотків менше [1], ніж час, який витрачається на інші мобільні ігри. Повсякденні ігри, включаючи настільні, карткові та дрібниці, мають тенденцію захоплювати мобільних геймерів на набагато більший проміжок часу – у середньому трохи більше 24 хвилин на день.

2.1 Аудиторія гіперказуальних ігр

Цільова демографічна група гіперказуальних ігр – це ті, хто хоче відволіктись, наприклад, під час поїздки або перегляду телевізора. Незважаючи на те, що вони популярні серед обох статей, жінки проводять більше часу, граючи в гіперказуальні ігри, ніж чоловіки, згідно з Індексом мобільних ігр, опублікованим Adjoe [1] наприкінці 2021 року.

За останні роки жанр набрав значної популярності, зросла і кількість розробників та видавців, і аудиторія, яка орієнтується на даний жанр.

Для розробників привабливість жанру полягає в короткому терміні розробки, який менше коштує та швидше дає змогу зрозуміти перспективність проекту. З точки зору аудиторії, жанр привабливий простотою та невибагливістю до вмінь гравця. Гру можна встановити та скоротати час або розважитись без необхідності розбиратись в правилах, концепціях гри, без необхідності вчитись грати. Модель розповсюдження зазвичай безкоштовна, що робить такі ігри ще більш доступними.

Так як розповсюдження гри безкоштовне, то дохід складається з двох джерел: покупки в доданку – це приблизно 1–5% від суми, та дохід з показу реклами в додатку – 95%. Реклама приносить найбільший прибуток і при проектуванні це потрібно враховувати [2].

Банерна реклама використовується в більшості гіперказуальних ігр, це – найпростіший спосіб монетизації. Подібно до невеликого віртуального рекламного щита, цей рекламний блок займає лише невелику частину екрана користувача і дозволяє гравцям продовжувати свій ігровий процес без будь-яких обмежень. Незважаючи на те, що такий банер є найпростішою стратегією монетизації гіперказуальних ігр, він має значний недолік, а саме, він має низьку ціну за тисячу показів і приносить невеликий дохід від реклами.

Реклама демонструється між ігровими сесіями, тобто рівнями або ігровими циклами. Динамічне оголошення автоматично запускається і його

можна пропустити після перегляду користувачем протягом кількох секунд (наприклад, десяти секунд). На відміну від банерної реклами, такі оголошення є більш помітними, оскільки вони переривають ігровий процес, а також займають весь екран. Це найпопулярніший варіант, який зустрічається майже у всіх іграх цього жанру, вони мають більшу потужність монетизації реклами в порівнянні з банерною рекламою. Їх можна легко інтегрувати в основний ігровий процес гіперказуальної гри.

Найголовнішим при проектуванні для розробника – це витримати баланс між ігровою сесією та часом на рекламу. Оптимальною кількістю є 2–3 оголошення в хвилину, тобто ігровий цикл має тривати приблизно 30 секунд.

Останній варіант реклами, який найчастіше використовується, – це відео з винагородою. Вони дозволяють користувачам увімкнути перегляд реклами в обмін на деякі бонуси або внутрішню ігрову валюту. Найефективніше це працює, коли користувачеві потрібно більше балів, життів або іншої валюти в додатку, щоб продовжити грати в гру. Однак, через характер ігрового процесу валюта гіперказуальної гри не завжди потрібна користувачам, щоб продовжувати грати, а це означає, що відео з винагородою можуть мати низький рівень залучення. Цей варіант, окрім основної мети монетизації, може збільшити і показники тривалості ігрового сеансу, та утримання гравців.

2.2 Аналіз аналогів

Гіперказуальний жанр ігор достатньо широко розповсюджений – в ньому об'єднані різні за механіками та стилем графіки проекти. Розглянемо дві гри, які мають велику кількість завантажень та позитивну оцінку на платформі публікації. Проекти можна вважати успішними, так як вони все ще підтримуються – отримують новий контент.

Гра Bottle Jump 3D. Графічне наповнення стилізоване. Моделі з різкими гранями, низька полігональність є частиною створеного стилю. Стилiзація не передає фактуру об'єктів, зберігає лише колір. Заднього плану як такого немає, всі об'єкти впливають на гру (рис. 2.1). Вся гра стилістично цілісна, жодний з об'єктів не вибивається з загального стилю. Гра Hopping Heads: Scream & Shout (рис. 2.2). Графічне наповнення стилізоване, персонажі дуже гіперболізовані.



Рисунок 2.1 – Графічний стиль Bottle Jump 3D

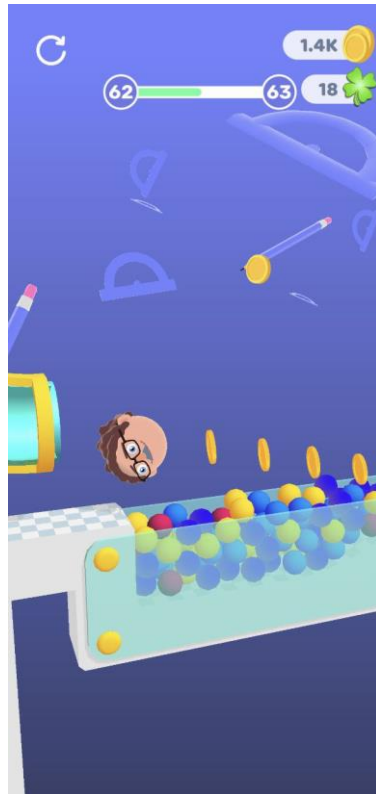


Рисунок 2.2 – Графічний стиль Hopping Heads

Моделі з гладкою поверхнею, без чітко виділених граней. Стилiзація не передає фактуру об'єктiв, зберiгає лише колiр. Заднiй план не вiдволiкає вiд гри i має метою заповнити пустий простiр, та надає бiльше контексту, iсторiї персонажу. Графіка бiльш навантажена, нiж у попередньому прикладі, але це робить вiзуал бiльш повним та насиченим, картинку можна розглядати, та вона не набридне дуже швидко.

3 ПРОЕКТУВАННЯ ТЕХНОЛОГІЧНОГО ПРОЦЕСУ ВИГОТОВЛЕННЯ ЕЛЕКТРОННОГО ВИДАННЯ

Мобільна гра – достатньо комплексний продукт, в якому комбінуються багато типів різних даних, і для ефективного створення та налаштування потрібно чітко розділити етапи проектування.

Глобально створення всього проекту можна поділити на чотири етапи: підготовка до розробки, розробка, створення промо матеріалів та ітерування або підтримка проекту.

Підготовка до розробки – це початковий етап, в якому визначаються цілі розробки, створюється технічне завдання, з якого виділяються та формуються конкретні задачі. Цей етап визначає всі виробничі процеси, які будуть виконуватись.

Підготовчий етап починається з розбору дизайн-документу, який надається видавцем. Дизайн-документ представляє собою дуже стислий опис продукту, в якому визначаються загальні концепції гри, приведено аналоги гри, концепти механік. Тут створюються перші розмиті рамки майбутньої гри, відштовхуючись від дизайн-документу вже створюється внутрішнє технічне завдання.

Внутрішностудійне технічне завдання (ТЗ) включає в собі більш розгорнуте пояснення механік, способи їх реалізації, приклади аналогічних або схожих механік в інших іграх, опис графіки, приклади графіки або конкретні концепти, опис технологій та технік які використовуються для реалізації графіки. Окрім інформації необхідної для створення основної гри, записуються ідеї ітерацій та оновлень, які можливі в майбутньому.

За створеним ТЗ визначають окремі задачі, які переносяться до Notion, де групуються за категоріями та пріоритетністю виконання. Під час створення проекту Notion являється платформою, за допомогою якої візуалізується як і загальна готовність проекту, так і окремих задач.

Стадія розробки – це створення наповнення проекту та його інтеграція до Unity, каталогізація створених матеріалів.

Створення контенту починається з визначення розміру створюваного контенту. Особливо це важливо, коли в грі будуть фізичні взаємодії, або коли предмети не мають реальних аналогів (вигадані). Визначений масштаб зберігається для всіх моделей, і це спрощує створення та налаштування моделей в Unity.

Розробка коду та наповнення ведуться паралельно тому для візуалізації базових механік, базового управління та налаштування камери використовуються “сірі коробки”, ще не створений контент замінюється примітивними об'єктами (куби, циліндри, трапеції). Це дає змогу концептуально візуалізувати майбутню гру. На цьому етапі вже можна зрозуміти, чи може гра розважати саме своїми механіками навіть без цікавої та привабливої графіки. Також цей етап важливий для побудови рівнів, визначення основних характеристик механік та об'єктів, з якими буде реалізована пряма та непряма взаємодія (висоти стрибка персонажа; розміру об'єкта, який потрібно розрізати, коли він падає; розмір паркану, щоб він не перекривав персонажа). Це дає змогу визначити розмір відповідних об'єктів, і в подальшому вносити менше змін для їх кращої роботи.

Паралельно з тим, як створюється код та концептуально вибудовується гра, з сірих коробок створюється оригінальне наповнення гри, відповідно спочатку створюється контент, який не може бути змінений під час тестування на сірих коробках. Це персонаж, та об'єкти оточення, з якими немає ніякої взаємодії. За необхідності створюються чорнові моделі об'єктів, з якими гравець може взаємодіяти, часто це об'єкти, які важко замінити примітивом.

В залежності від стилю та типу об'єкта етапи створення можуть відрізнятися. В основі лежить один виробничий процес, але він доповнюється різними етапами в залежності від того, що створюється. Можна виділити основні етапи, які проходить будь-яка модель; етапи, які

проходить персонаж, неорганічні та органічні об'єкти; і контекстні етапи, які залежать повністю від об'єкта та його призначення та стилю графіки.

Розробка моделі починається з чорнової моделі, за допомогою примітивів вибудовується загальна структура та пропорції моделі. Для блокауту використовуються вбудовані примітиви та елементарні інструменти переміщення, обертання, масштабування в просторі. На етапі моделювання вносити кардинальні зміни в пропорції може бути трудомістко, тому, оперуючи примітивами, визначається образ моделі, і тільки коли є впевненість у тому, що пропорції вірні, починається створення моделі, вже спираючись на чорнову модель.

Після створюється чистова модель об'єкта. В залежності від складності моделі та типу хід процесу може відрізнитись. Моделінг – це найпростіший підхід, в якому за допомогою простих інструментів та маніпуляцій формується об'єкт. Для складних за формою об'єктів просто моделювання може займати багато часу і бути невиправданим, тому додатково використовуються булеві операції, для чого створюються додаткові моделі, які вирізають або додають необхідний об'єм. Після булевих операцій виконується чистка сітки для уникнення артефактів затінення та прискорення роботи. Третім підходом є скульптинг, що використовується переважно для складних органічних моделей. Скульптинг схожий на класичну ліпку з глини, де форма нарощується шарами, а зайвий об'єм прибирається. У процесі скульптингу використовується високополігональна модель, яка не підходить для подальшої роботи. Тому після завершення додатково потрібно виконати ретопологію моделі, тобто створити нову, низько полігональну сітку поверх високо полігональної.

Після моделінгу створюється розгортка моделі для подальшого створення та накладання текстури. В залежності від стилю процес розгортки та подальші етапи відрізняються. Класична розгортка підрозуміває під собою призначення швів, за якими модель розділяється та проектується на площину [3]. Така розгортка дає можливість використовувати текстури, на яких

зберігається багато інформації про поверхню. Для створення текстур часто необхідне запікання технічних карт, які зберігають інформацію про деталі з високополігонального скульпту, товщину окремих частин моделі, контактні тіні тощо. Текстурування виконується на основі отриманих при запіканні карт, або малюється за допомогою різних пензлів. Такий підхід використовується для більш реалістичних моделей.

Другим методом розгорткування та текстурування є використання колірного атласу. Атлас представляє собою невелику текстуру, де кожен піксель може зберігати інформацію про колір частини моделі. Для розортки модель також ділиться на частини та проектується на площину, після чого кожна точка острову переміщаються на необхідний колір. Такий підхід надає можливість зберігати всю інформацію про колір не тільки однієї моделі але і всього асету. Це більш стилізований підхід, який зберігає менше деталей але більш художній.

Етап затінення включає в себе створення та налаштування матеріалів, необхідний для попередньої оцінки зовнішнього вигляду моделей та для рендерингу промо матеріалів в майбутньому. Створені матеріали також впливають на процес налаштування моделей в Unity.

Етап ригінгу та анімації виконується переважно для персонажів. Створюється та прив'язується до моделі необхідний скелет, який дає змогу розробити необхідні анімації.

Після того, як модель проходить попередні етапи, вона готова до експорту. Утворюються групи моделей, які експортуються до Unity, окремо експортуються персонажі. Повний виробничий цикл показаний на рис. 3.1.

Після експорту об'єкти налаштовуються в Unity (рис. 3.2), основна ціль – об'єднати створений контент та примусити його працювати в грі. Тут вже гра набуває фінального свого вигляду, тут немає кардинальних змін, лише невеликі правки для покращення відчуття від гри. При додаванні, налаштуванні, змінненні контенту використовується система контролю версій, вона дозволяє документувати правки, які вносяться, та працювати над

проектом одночасно всій команді. Кожна локальна зміна файлу може бути задокументована і збережена з можливістю відмінити зміни та створювати альтернативні варіанти налаштувань для визначення кращого рішення поставлених задач.

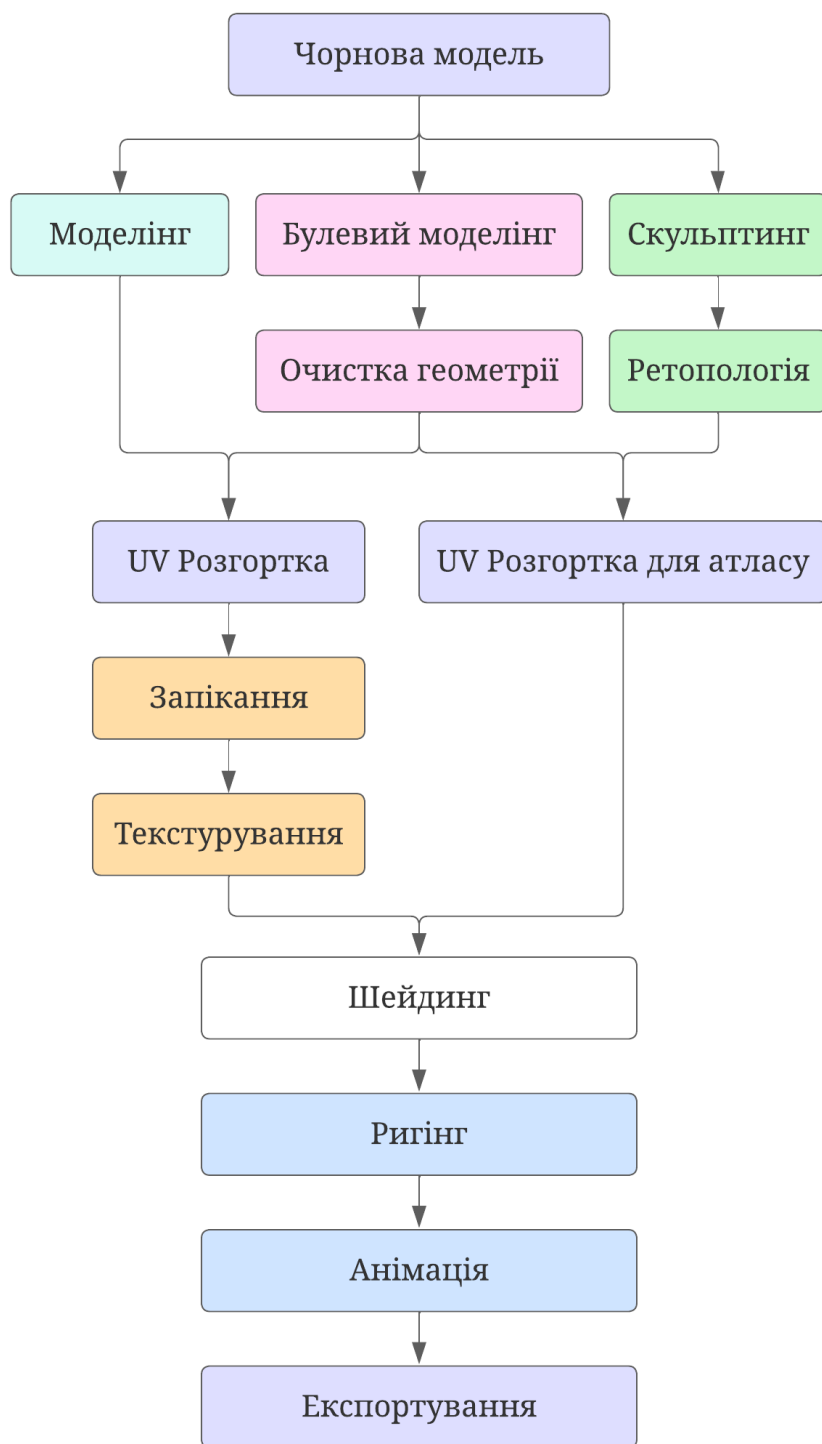


Рисунок 3.1 – Узагальнений виробничий процес створення наповнення гри



Рисунок 3.2 – Узагальнений виробничий процес налаштування контенту в Unity

Налаштування в Unity частково спирається на налаштування в файлах оригіналах, а структура проекту дозволяє прискорити налаштування.

Виробничий процес в Unity починається з налаштування текстур. Це опції збереження текстури, як вона стискається, як накладається на модель, які канали зберігає та як вони обробляються, чи генеруються зменшені версії текстури для оптимізації обчислень графіки і т.д. Правильне налаштування текстури критичне для правильного відображення об'єктів, забуті або неправильно встановлені налаштування та опції обробки створюють різноманітні артефакти. Частково процес налаштування виконується автоматично при імпортуванні, але від типу та призначення текстури потрібно змінити деякі налаштування.

Другим етапом є налаштування матеріалів. Матеріали створюються на основі матеріалів оригіналів. Використовується один чи два основних шейдерів з різними параметрами та можливостями.

Експортовані моделі налаштовуються для правильного відображення в Unity, налаштування включає і прості опції, такі, як модифікатори розміру або прив'язки матеріалу, так і складні, як налаштування імпорту нормалей, або обробки груп вершин для анімації. Спираючись на матеріали, створені в файлах-оригіналах, до моделей прив'язуються матеріали, створені в Unity. Це прискорює роботу вже при створенні префабів, або при внесенні правок до моделей та їх оновленні.

Для моделей персонажів виконується налаштування скелету. Створюється основний аватар, який прив'язується до всіх персонажів, виконується обробка анімацій для правильного відображення. Також виконується тестування роботи скелету та анімацій.

Створюються префаби – попередньо згруповані та налаштовані об'єкти, які будуть використовуватись для побудови сцен. Об'єкти комбінуються: до них підключаються скрипти для анімації моделі або анімації матеріалу. Вони зберігаються окремо і дають можливість швидко змінити об'єкти на всіх рівнях, та сценах, що пришвидшує внесення правок.

Налаштування сцени виконується на попередньо створеній сцені, на якій виконувався чорновий варіант рівня. Тут примітиви замінюються на вже створені моделі та префаби.

Створення промо-матеріалів – це останній етап (рис. 3.3) перед публікацією продукту, він включає в себе створення іконок, створення зображень оформлення web-сторінок для ресурсів, де публікуються гра, запис та редагування рекламних роликів гри. Промо-матеріали створюються на основі вже існуючих ресурсів.

Створюється композиція з розроблених об'єктів, вона має передавати якийсь аспект продукту, та має привернути увагу користувача. На сцені виставляється освітлення та виконується рендеринг. Відрендерена композиція за потреби обробляється в графічних редакторах і також експортується до проекту.

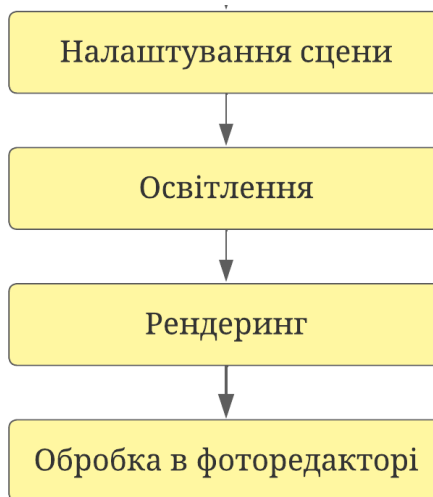


Рисунок 3.3 – Узагальнений виробничий процес створення промо матеріалів

Після завершення розробки супроводжувальних матеріалів виконується огляд створених ресурсів, їх групування в асети, та підготовка до використання в майбутніх проектах. Ресурси дублюються та зберігаються окремо, після чого об'єкти підготовлюються для асетування, видалення зайвих об'єктів, текстур, матеріалів, анімацій, створення асету та збереження. Етап спрямований на пришвидшення роботи над майбутніми проектами.

Вже у готового проекту є три варіанти подальшого існування. Перший і найчастіший – це ітерування гри. Вже на основі створення контенту змінюється конкретна частина проекту, ітерування часто змінює баланс гри, або приносить нові акценти в геймплейному плані, графічне наповнення при ітеруванні не змінюється або зміни незначні. Мета ітерування – змінити гру в кращу сторону і зробити її більш привабливою для користувача, також важливо зменшити ціну за скачування, збільшити час, який проводить користувач в грі.

Другим варіантом є підтримка гри новим контентом. Це нові рівні, нові персонажі, оточення і т.п. Найяскравішим прикладом є додавання нових “скінів” (персонажів), нових рівнів, тобто, здебільшого декоративних елементів, які не змінюють геймплей. При наповненні ядро гри, механіки, баланс, головна концепція не змінюються, а лише доповнюються для того,

щоб вже зацікавленого користувача зберегти якнайдовше. Процес створення цих додаткових ресурсів не відрізняється від створення основних.

Останнім варіантом є закриття проекту та припинення будь якої підтримки, в даному випадку вся робота припиняється і починається робота над наступним проектом.

Вибір між цими варіантами розвитку проекту залежить, в основному, від метрик, які показує вже випущена гра. Основними метриками є ціна за установку (CPI – cost per install) та тривалість життя користувача (в грі) (LT – lifetime) [4].

За метриками вже реалізованого проекту визначається його доля. Зазвичай це ітерування проекту з кардинальними змінами частини проекту і подальший моніторинг того, як змінюються метрики. Якщо метрики гарні, створюється додатковий контент для подовження тривалості життя користувача. Закриття проекту відбувається, коли метрики погані, а видавець вважає, що проект не є перспективним для ітерування.

4 ВИБІР ІНСТРУМЕНТАЛЬНИХ (ПРОГРАМНИХ І ТЕХНІЧНИХ) ЗАСОБІВ РОЗРОБКИ

4.1 Програмні засоби

Так як продукт достатньо комплексний, ряд програмного забезпечення широкий, і включає в себе не тільки інструменти створення і редагування, а і інструменти менеджменту контенту та задач.

Інструменти менеджменту:

- Notion (аналогі – evernote, jira, trello);
- GitHub (аналогі Perforce);
- PureRef;
- Eagle (аналогі – Adobe Bridge).

Інструменти створення та редагування:

- Adobe Illustrator (аналогі – Corel Draw, Affinity Designer);
- Adobe Photoshop (аналогі – Gimp, Affinity Photo);
- Blender (аналогі – Maya, 3dsMax,);
- Unity (аналогі – UE4, godot).

Для організації постановки завдань та менеджменту використовується Notion платформа з інструментарієм для постановки задач, де описується завдання, підзадачі, та її властивості, з яких найбільш важливими є пріоритетність, виконавець, термін задачі (рис. 4.1). Є можливість організації посилань, різних ресурсів доступ до яких може знадобитись. Аналогами є Trello. Функціонал Trello майже ідентичний до Notion. Обидва ресурси мають додатки на платформах Windows, MacOS, Android, iOS та Web-версії.

GitHub використовується для одночасної сумісної роботи та контролю версій. Він дає можливість паралельно працювати та вносити зміни до проекту всій команді. Також важливою є можливість працювати над кількома

версіями того самого проекту (важливим є для створення версій з дрібними змінами).

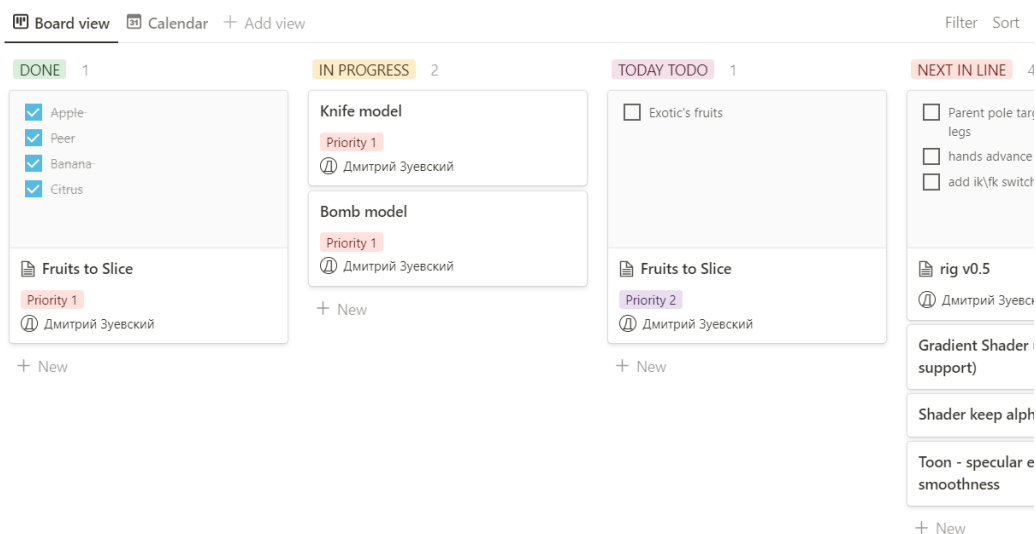


Рисунок 4.1 – Доска задач Notion

Недоліком є те, що git спрямований на збереження та обробку саме коду (текстових даних), він надає можливість швидко знаходити та об'єднувати зміни в текстових даних, але проект мультимедійний, і в собі зберігає тривимірні моделі, анімації цих моделей, аудіо файли, зображення. Зміни в них вже не можуть бути відслідковані та частково об'єднані – це зумовлено, в першу чергу, складнішим типом даних та їх різноманітністю. Це вирішується більш суворою документацією змін, що вносяться.

Аналогом є Perforce. Він надає схожий функціонал, дозволяючи паралельну роботу. В ньому частково вирішена проблема мультимедійних даних: для них можливий попередній перегляд, що полегшує оцінку внесених змін. Perforce комерційний, тому перевага надана GitHub.

PureRef – інструмент створення дошок референсів, мудбордів. Використовується як на початковому етапі створення, так і при полішингу, коли вносяться невеликі зміни в колір, композицію, положення камери тощо.

Інструмент дозволяє швидко оперувати великою кількістю зображень, дозволяє групувати і вирівнювати, що дозволяє одночасно бачити всі можливі шляхи графічних рішень (рис. 4.2). Використовується на всіх етапах

розробки, починаючи зі створення дизайн-документу, де він служить для групування аналогів та концептів, при розробці контенту для доступу до концептів, при завершальній стадії розробки для оцінки різних варіантів іконок промо-матеріалів. Аналогів йому немає, може бути замінений на вручну створені, великі за форматом графічні файли Photoshop, але такий підхід займає в рази більше часу, а файли занадто важкі та потребують ПЗ для їх відкриття та перегляду.

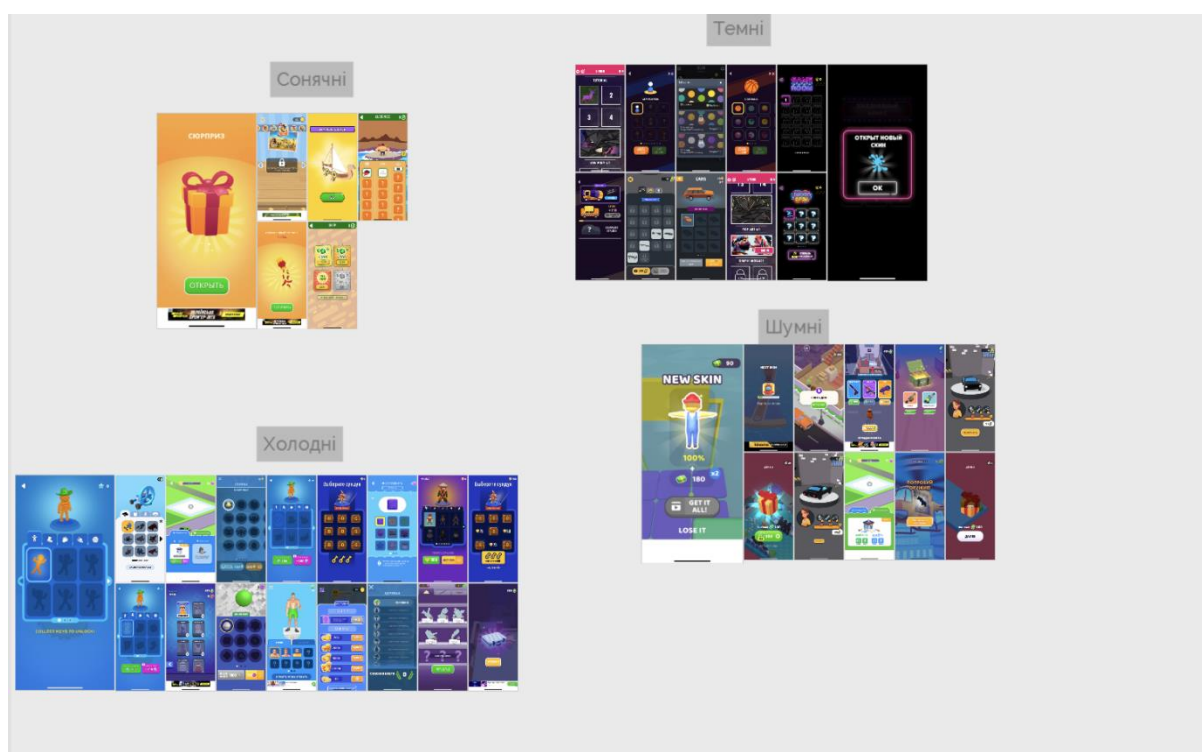


Рисунок 4.2 – Доска референсів інтерфейсу

Eagle (рис. 4.3) – інструмент створення мультимедійних баз даних, картинок референсів, дає можливість упорядкувати референси за тегами, категоріями, шукати за кольором. Це зручний інструмент для створення власної бібліотеки зображень, яка дає можливість швидко знаходити потрібні референси. Вона дозволяє зберігати, переглядати та впорядковувати більшість мультимедійних файлів, починаючи від текстових документів, до форматів тривимірних моделей. Структура створених бібліотек дозволяє працювати над ними сумісно за допомогою GitHub.

Використовується для створення бази даних концептів, шрифтів, іконок, збереження промо матеріалів.

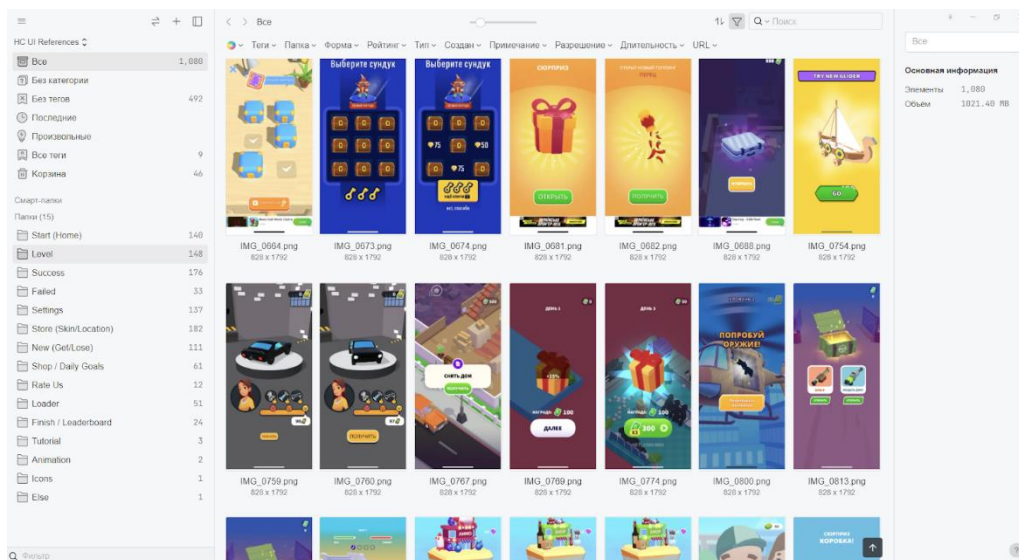


Рисунок 4.3 – Бібліотека Eagle

Аналогом є Adobe Bridge – він дозволяє створювати та організувати файли схожим образом. Є підтримка скриптів та інтеграція з іншими продуктами Adobe. Недоліком є більш складна структура, яку складніше передавати. Також Bridge, як і всі продукти Adobe, розповсюджуються за підпискою.

4.2 Інструменти створення та редагування

Adobe Illustrator – інструмент створення векторної графіки, який використовується для створення інтерфейсу користувача. Надає можливість створювати та редагувати векторну графіку, має широкий набір інструментів. Великою перевагою є синхронізація з Photoshop, завдяки чому можна переносити файли з програми в програму з можливістю редагування без втрати взаємозв'язку.

Аналогами є Affinity Designer та CorelDRAW. Функціонально вони майже ідентичні, і перевага надається за суб'єктивними параметрами такими, як дизайн інтерфейсу, звичність. Різниця є в ліцензуванні: Affinity Designer

розповсюджується за одноразову покупку, Illustrator та CorelDRAW – за підпискою.

Adobe Photoshop – інструмент для створення текстур, а також угруповання елементів інтерфейсу. Цей растровий редактор має широкий інструментарій обробки растрової графіки, підтримує роботу з шарами, надає змогу не деструктивно вносити значні зміни в графіку.

Після створення елементів інтерфейсу в Illustrator вони передаються до Photoshop, де групуються для утворення однієї таблиці спрайтів, після чого експортуються до движку. Також використовується для обробки промо-матеріалів, це обрізка до певних форматів, корекція кольору, вставка тексту та логотипів.

Аналогом є Affinity Photo, він надає ідентичний функціонал. Різниця є в ліцензуванні: Affinity Photo розповсюджується за одноразову покупку Photoshop за підпискою.

Blender – інструмент створення та редагування мультимедіа, акцентований на повному циклі створення тривимірних моделей. Є можливість автоматизації за допомогою скриптів, та вбудований менеджер асетів.

В роботі це – основний інструмент створення наповнення проекту, у ньому проектуються всі тривимірні моделі, створюються анімації, і в подальшому експортуються до движку. Функціонал покриває всі задачі, які можуть виникнути при розробці. Також в ньому, крім основного контенту, створюються промо-матеріали, виконується налаштування сцени та виконується рендер.

Ряд аналогів широкий – Autodesk Maya, Autodesk 3ds Max, Cinema4D, Houdini. Всі ПЗ надають функціонал для повного циклу створення тривимірної графіки, але вони акцентують увагу на певному аспекті виробничого процесу (табл. 4.1).

За рахунок усередненості функціоналу Blender краще підходить, як універсальне ПЗ створення графічного наповнення.

Таблиця 4.1 – Порівняння ПЗ для створення тривимірної графіки

	Blender	Maya	3ds Max	Cinema4D	Houdini
Можливість створення параметричних моделей	Є	Немає	Немає	Немає	Є
Комплексні інструменти ригінгу	Є	Є	Немає	Є	Є
Комплексні інструменти анімації	Немає	Є	Немає	Немає	Є
Інструменти менеджменту асетів	Є	Немає	Немає	Немає	Є
Ліцензія	Вільна	Підписка	Підписка	Підписка	Підписка

Unity – ігровий движок, інструмент, в якому весь створений контент комбінується, де створюється логіка роботи гри, та компілюється вже в готовий продукт. Аналогом є Unreal Engine; основна його відмінність – мова програмування, яка використовується: для Unity це C#, для Engine – C++. Функціонально вони дозволяють створювати проекти однакові за складністю, та графічними рішеннями. Обидва движки умовно безкоштовні (якщо проект за рік заробив більше певної суми, то виконується виплата). Перевага віддається за вибором мови програмування та суб'єктивними перевагами розробника.

4.3 Технічні засоби

Технічні засоби визначаються рекомендованими вимогами програмного забезпечення, але з поправкою на те, що робота буде

виконуватися не в одній програмі, а паралельно в декількох, що дає більше свободи та прискорює роботу, але потребує більше ресурсів (табл. 4.2).

Таблиця 4.2 – Рекомендовані системні вимоги для використаного ПЗ

	CPU	GPU	RAM
Unity	З підтримкою 64-розрядних розрахунків, 4 ядра, 2.4 GHz	8 GB, з підтримкою DirectX 10	8 GB
Blender	З підтримкою 64-розрядних розрахунків, 8 ядра, 2.7 GHz	8 GB, з підтримкою OpenGL 4.3	16 GB
Adobe Photoshop	З підтримкою 64-розрядних розрахунків, 4 ядра, 2.4 GHz	4 GB, з підтримкою DirectX 12	16 GB
Adobe Illustrator	З підтримкою 64-розрядних розрахунків, 4 ядра, 2.4 GHz	2 GB, з підтримкою DirectX 12	8 GB
Git	–	–	1 GB
Notion	–	–	2 GB
PureRef	–	–	2 GB
Eagle	–	–	4 GB

5 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СТРУКТУРИ ЕЛЕКТРОННОГО ВИДАННЯ, СКЛАДОВИХ ФІРМОВОГО СТИЛЮ

Весь проект можна поділити на декілька окремих структур, вони необхідні на різних етапах для різних файлів. Першочерговою є структура файлів-оригіналів всіх об'єктів, тут створюється та групується основний контент гри. Структура має надавати можливість легко орієнтуватись в уже створених моделях, в моделях, які зараз на етапі створення, в технічних об'єктах, які необхідні для роботи. Другою є структура проекту Unity, вона багато де спирається на структуру оригіналів, але деякі речі змінюються заради оптимізації або спрощення структури, так як контент тут вже не редагується, а лише комбінується.

Всі файли, що створюються та відносяться до проекту, можна поділити на дві категорії: це файли проекту Unity та файли-оригінали (моделей, супутніх файлів таких як промо-матеріали). Тому існує дві папки – одна з них включає проект Unity, друга – всі файли-оригінали. Файли-оригінали можна поділити на ті, в яких будуть в ході роботи створюватись моделі, текстури, які будуть безпосередньо експортуватись до Unity, та ті, які будуть супроводжувати продукт, іконки, оформлені скріншоти, постери тощо.

5.1 Структура оригіналів

Файл-оригінал моделей формату .blend – власний формат, розроблений Blender Foundation для програми Blender. Файл достатньо гнучкий і може зберігати багато різних типів даних, починаючи з 3d моделей растрових та векторних, закінчуючи більш специфічними, такими, як хмари точок.

В основному, для моделей створюються два окремих файли – це ch.blend та env.blend. В першому зберігаються персонажі та їх риг, анімації, другий служить для всіх моделей оточення, як геймплейних, з якими можна

взаємодіяти, так і декоративних. Для створення супутніх матеріалів таких, як іконки, рекламні постери, шапки для сайту, додатково створюються файли іконки або постеру (icon.blend, fillscreen.blend тощо), в яких комбінуються необхідні моделі. В таких додаткових файлах зазвичай вже нічого не редагується, і всі об'єкти додаються, як посилання на ch.blend та env.blend.

Структура .blend файлу відрізняється від тої, яка йде за замовчуванням і створена для більш швидкого початку роботи над створенням моделей та їх групуванням. Це одна сцена з кількома колекціями (рис. 5.1).

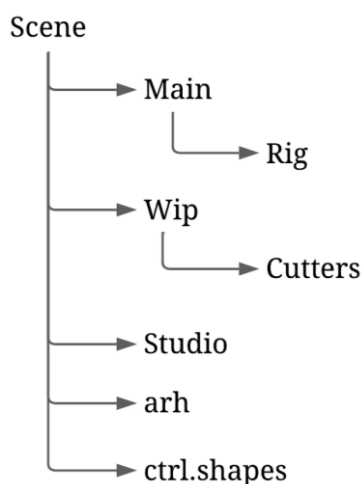


Рисунок 5.1 – Структура файлу оригіналу

Колекція Main включатиме в собі створені колекції об'єктів для експорту. Це основна колекція, в якій вже створені об'єкти перейменовуються, структуруються та групуються за внутрішніми колекціями. Внутрішні колекції формуються за призначенням об'єкту та версією, коли він додається до гри (рис. 5.2).

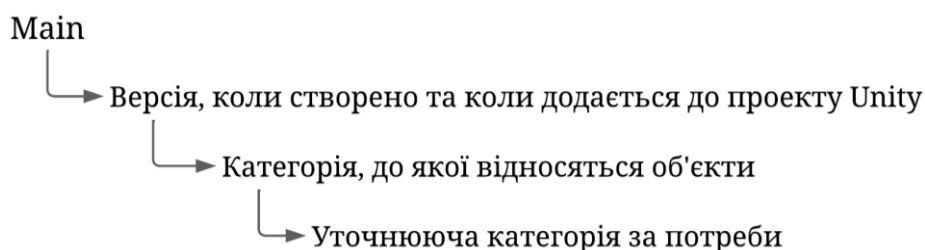


Рисунок 5.2 – Структура основної колекції

Подібна ієрархія переноситься до проекту Unity, де файли розподілені за версією та категорією.

Колекція Rig включає в собі основний риг та його контролери, його відокремлення від основних моделей спрощує орієнтацію в проекті. В основному використовується один риг, на який опираються всі персонажі.

Колекція Wip та її внутрішня колекція Cutters служать для створення моделі або групи моделей, які ще не завершені і в даний час моделюються, редагуються. Внутрішня колекція Cutters служить для збереження об'єктів, які використовуються при булевих операціях з моделями. Для цієї колекції структура та неймінг не є важливими, так як після завершення роботи над моделю вона переноситься до Main, а все, що використовувалось під час створення, видаляється або за потреби переноситься до колекції arh, де може бути знайдено за потреби в майбутньому.

Колекція Studio зберігає в собі базовий пресет освітлення, камеру, її контролери та декілька варіантів фону. Здебільшого колекція використовується для підготовки промо-матеріалів (іконки, постери, шапки, рекламні ролики).

Колекція arh – це технічна колекція, яка використовується для збереження об'єктів, які можуть знадобитися в майбутньому. Це можуть бути об'єкти до застосування модифікаторів, або до їх кардинальних змін. В основному використовується для кардинальних відмін редагування.

Колекція ctrl.shapes – ще одна технічна колекція, яка зберігає в собі об'єкти, які використовуються як контролери ригу, освітлення, камери.

Найменування об'єктів виконується без скорочень з додаванням номеру через крапку: найменування.001. Так декілька варіантів того самого об'єкту мають спільну назву з різними номерами. Якщо об'єкт складається з декількох частин, різних за призначенням, до найменування через крапку додається уточнення що це, а потім вже номер: найменування.уточнення.001. Така сама структура використовується для найменування матеріалів та текстур. Така структура назв дає змогу швидко знаходити потрібні об'єкти, та визначити, що це конкретно, і не допускати дублювання назв.

5.2 Структура супутніх матеріалів

Промо-матеріали мають окрему папку. Всі промо-матеріали мають окремі оригінали, які спираються на попередньо створений контент. Першочерговим є файл .blend, в якому комбінується необхідна композиція, додатковим є файл .psd, в якому створена композиція обробляється. В .blend структурно нічого не відрізняється, але в категорії Main об'єкти формуються в необхідну композицію для подальшого рендеру іконок, постеру або анімації. В подальшому рендер обробляється та експортується для публікації. Тут спрощена структура – всі файли знаходяться в одній папці і групуються лише за назвою, до одного промо-матеріалу зазвичай відносяться три файли .blend, .psd, та експортований матеріал може відрізнитись за типом.

5.3 Структура проекту Unity

Кореневою папкою проекту є Assets, в ній зберігаються всі файли, що відносяться до проекту, в першу чергу все розподілено за приналежністю ресурсів до певної категорії (рис. 5.3).

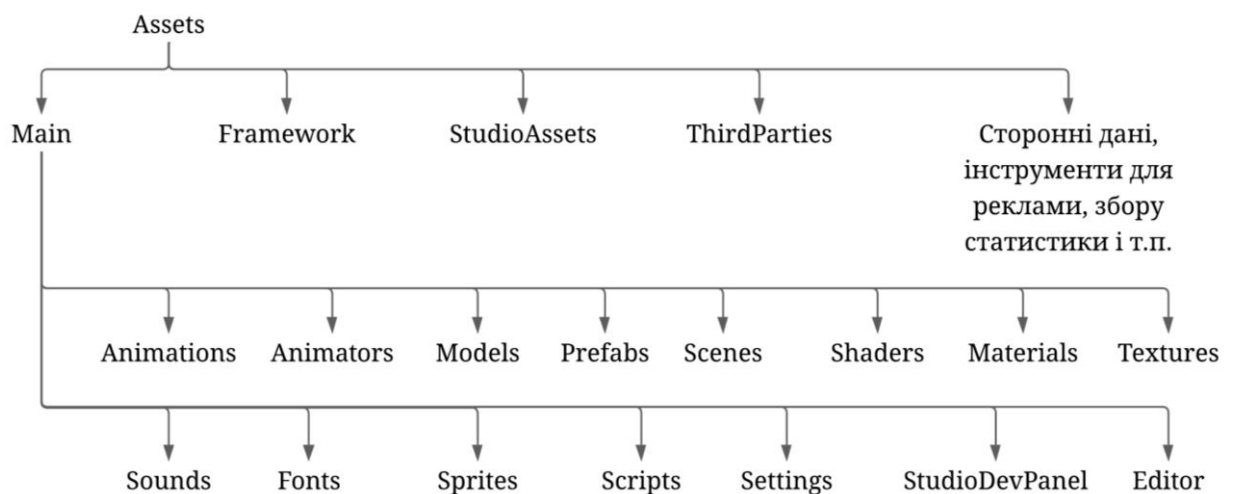


Рисунок 5.3 – Структура проекту Unity

Main – це папка, яка зберігає всі ресурси, створені для конкретно даного проекту (моделі, анімації, текстури та матеріали, скрипти, префаби, рівні налаштування). В середині всі файли розподілені за їх типом, а пізніше – вже за категорією, версією. В собі вона включає контент, розподілений за типом та призначенням.

Окрім наведеної загальної структури, можуть бути додаткові категорії специфічного контенту, який створений спеціально для даного проекту та загалом використовується рідко. Основними категоріями для візуального контенту є розділи Models, Materials, Textures, Sprites, Fonts, Prefabs.

В Models структура не відрізняється від файлів оригіналів, тобто, окремі файли групуються за версією, потім – за категорією контенту, це дає змогу швидко змінювати та оновлювати файли, не виникає путаниця при роботі.

В Materials структура вже відрізняється: загалом використовується невелика кількість матеріалів, два основних – це матеріал оточення та матеріал персонажів, які використовують різний набір текстур (атласів), всі властивості матеріалів зберігаються у наборі текстур. Це дає змогу одним матеріалом замінити декілька різних за властивостями матеріалів. За необхідністю створюються групи матеріалів, наприклад для різного неба, або специфічні анімовані матеріали.

Textures групуються, як матеріали – це дві основні групи оточення та персонажі, за потреби створюються додаткові категорії.

Fonts включає в себе шрифти, які використовуються в грі, в основному це один основний шрифт, тому складна структура не потрібна.

В Sprites зберігаються елементи інтерфейсу та іконки. Групуються вони за призначенням: основними категоріями є інтерфейс, іконка гри, іконки що відображаються безпосередньо в грі (локації, персонажі), ряд спрайтів, які використовуються для запису реклами і які не включаються до основної гри.

В Prefabs зберігаються вже налаштовані об'єкти, з підключеними матеріалами, скриптами. Основні внутрішні категорії – це, як і в оригіналах –

персонажі та оточення, а далі групування виконується за версією, коли додано префаб та за призначенням префабу.

Framework – це модуль, який включає в себе контент, який необхідний для кожного проекту, це напрацювання з інших проектів, які є універсальні і вирішують часті задачі. Це і набір скриптів, і власні шейдери, або власні інструменти для налаштування контенту, відладки, або для запису реклами. Особливістю модуля є те, що він оновлюється і переходить з одного проекту до іншого, і напрацювання з нових проектів можуть бути використані в старих, не “зламуючи” логіку роботи.

StudioAssets – папка, яка зберігає ресурси, які взяті з інших власних проектів, це контент вже створений та налаштований, але на відміну від модуля Framework, цей контент не універсальний і додається за конкретної потреби. Він включає в себе моделі, вже налаштованих персонажів, префаби. Часто використовується на ранніх етапах, коли основний контент ще не створений для наповнення рівнів моделями, які пізніше будуть замінені на оригінальні.

ThirdParties, як і StudioAssets – це вже налаштований контент, але він створений іншими авторами. Включає різноманітний контент, починаючи з моделей, закінчуючи шейдерами та ефектами. Сторонній контент зменшує необхідний час на розробку, так як використовується вже готове рішення.

Додатково проект включає в себе сторонні ресурси, які вбудовуються в проект і служать для збору статистики, показу реклами, реалізації покупок, та інших сервісів. Список сторонніх інструментів відрізняється від видавця, обраного рекламного сервісу, платформи, через яку монетизується продукт.

5.4 Структура файлу оригіналу об'єктів для проекту Fruit and Juice

В грі відсутні складні персонажі, які потребують окремого ригу та анімації, тому всі об'єкти розроблені в одному файлі, стандартному за структурою (рис. 5.4).

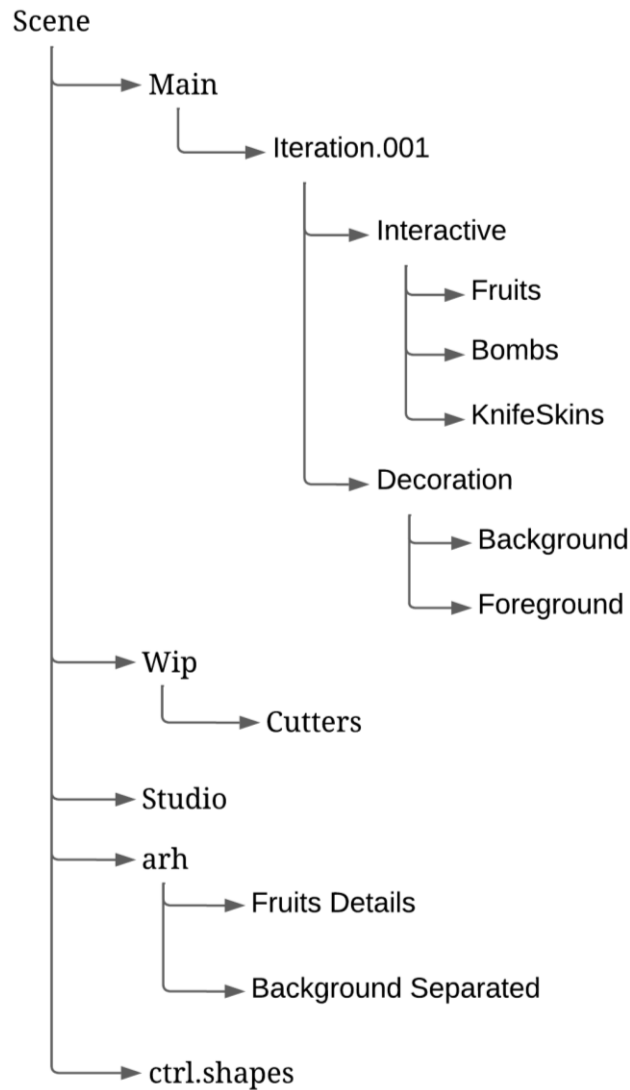


Рисунок 5.4 – Структура файлу оригіналу об'єктів

Об'єкти за призначенням утворюють дві різні категорії: об'єкти, з якими гравець може взаємодіяти, та об'єкти оточення, які служать декорацією і ніяк не впливають на гру. Інтерактивні об'єкти, в свою чергу, поділені за типом на ті, з якими потрібно взаємодіяти, ті, з якими не потрібно, та об'єкт взаємодії. Декоративні об'єкти поділені на ті, що знаходяться на передньому плані, та на задньому. Кожна з цих категорій експортується окремо, для спрощення редагування за потребою в ході роботи. Також є дві категорії, збережені для подальшого використання: в них зберігаються об'єкти деталізації фруктів (це різноманітні листки, набори тисячі геометрії, та деталі, які не увійшли до основної гри) та об'єднаний задній план, представлений рядом окремих об'єктів.

5.5 Структура проекту Fruit and Juice в Unity

Організація її не відрізняється від загально описаної структури, доповненої категоріями для більш специфічних ресурсів за призначенням. Це – згруповані ресурси для частинок, які з'являються під час розрізання фруктів, для них виділена окрема категорія в текстурах та матеріалах та префабах (рис. 5.5).

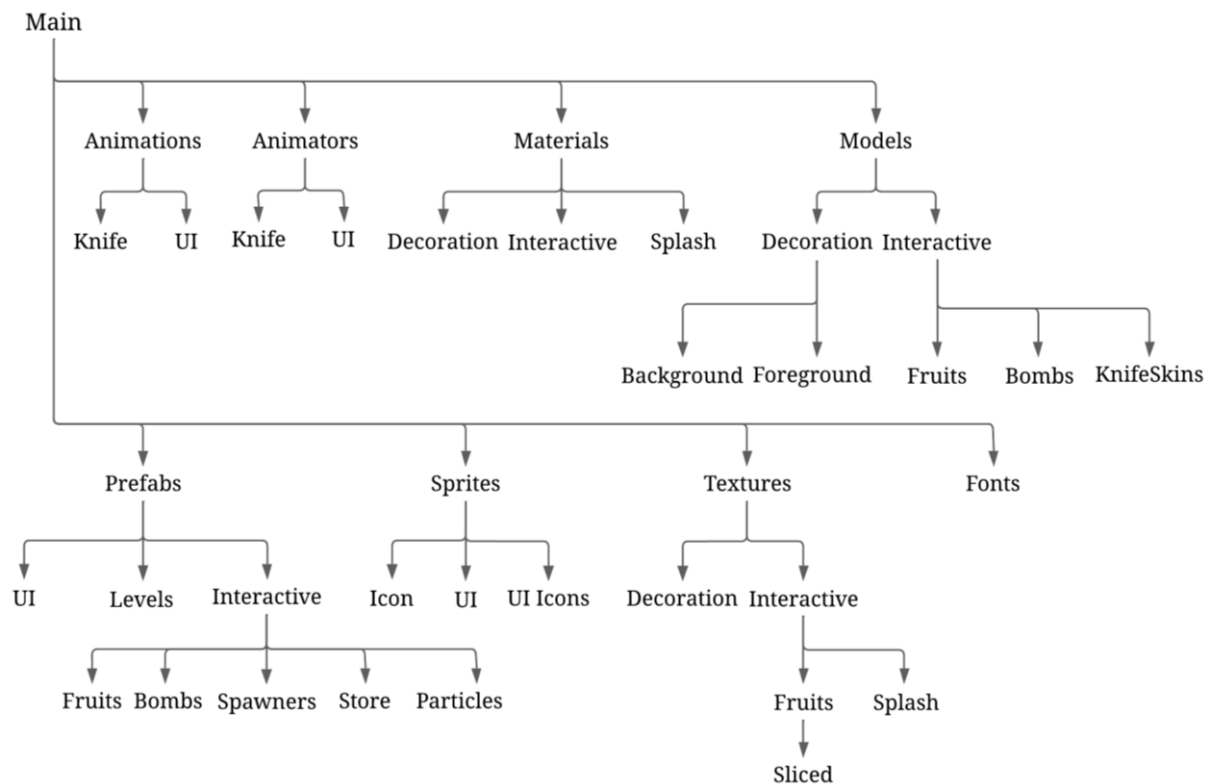


Рисунок 5.5 – Структура проекту Fruit and Juice в Unity

Точно визначена структура як самого проекту Unity, так і файлів-оригіналів, пришвидшує проектування на всіх етапах, спрощує експортування ресурсів їх налаштування в Unity. Структура зберігається, навіть якщо через кількість матеріалів в ній немає потреби. Це зберігає однотипність проектів, і способу навігації в них, тобто при переході від одного проекту до іншого немає потреби шукати необхідні ресурси. Точний

неймінг об'єктів прискорює орієнтацію в проекті, та прискорює налаштування матеріалів та префабів.

Визначення та опис необхідних правок, як і їх внесення, також пришвидшується. При роботі в команді виникає менше питань про розташування тих чи інших ресурсів в проекті. Також чітке дотримання структури підвищує швидкість асетування ресурсів для повторного використання в майбутніх проектах.

6 РОЗРОБКА ГРАФІЧНОГО ДИЗАЙНУ ЕЛЕКТРОННОГО ВИДАННЯ

6.1 Стилiстичне рiшення

Створення графіки починається з визначення стилю об'єктів, в залежності від цього зміниться виробничий процес. Також, на етапи, які проходить об'єкт, впливає призначення моделі.

В основі всього стилю лежить декілька правил, які спрямовані на простоту сприйняття та ідентифікації об'єктів, що важливо для жанру.

Першочерговим є створення ефективного силуету об'єкту, який буде легко зчитуватися користувачем (рис. 6.1). Якщо це зумовлено структурою об'єкту, силует поділяється на декілька основних частин. Вже до цього силуету додаються функціональні деталі, які мають надати певного реалізму об'єкту (рис. 6.2). Деталі мають не перевантажувати об'єкт, а лише додавати точки інтересу, на яких можна буде зосередити погляд.

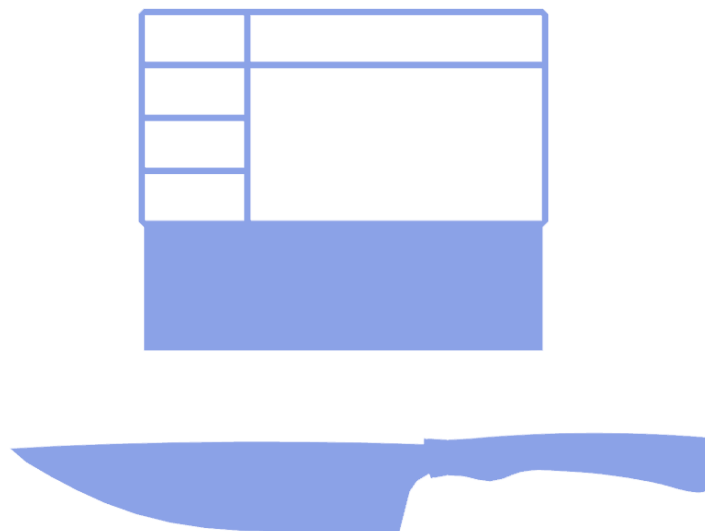


Рисунок 6.1 – Силует моделей



Рисунок 6.2 – Силует моделей з доданими деталями

Об'єкти не мають жодної фактури та не зберігають властивостей поверхні, зберігається лише колір. В свою чергу, колір формується градієнтами, вони урізноманітнюють поверхню (рис. 6.3). Використання однотонної заливки на всьому об'єкті чи його значній частині робить поверхню однотипною. Простота поверхні також спрощує сприйняття графіки, це має і технічну перевагу – менше ресурсів необхідно для обчислення графіки.

Затінення просте і має надати об'єму об'єктам, але без різких та глибоких тіней (рис. 6.3), які будуть забирати увагу на себе та дещо ускладнювати сприйняття силуету. Тінь доповнюється контурним затіненням.



Рисунок 6.3 – Накладена текстура та остаточне затінення моделі

Якщо всі об'єкти слідують правилам, то вони будуть формувати просту для розуміння картинку, що є першочерговим для гіперказуального жанру.

6.2 Розробка моделей

Розробка моделі починається з чорнової моделі (blockout) [5], за допомогою примітивів вибудовується загальна структура та пропорції моделі. Для чорнової моделі використовуються вбудовані примітиви та елементарні інструменти переміщення, обертання, масштабування в просторі. На етапі моделювання внесення кардинальних змін в пропорції може вимагати значних обсягів роботи, тому, оперуючи примітивами, визначається образ моделі (рис. 6.4), і тільки коли є впевненість у тому, що пропорції вірні, починається створення моделі, вже спираючись на чорнову модель.



Рисунок 6.4 – Чорнові пропорції моделі

В залежності від типу об'єкта та його складності, є декілька підходів до його створення. Перший – це просто моделювання, в цьому підході на основі одного або декількох примітивів, елементарних маніпуляцій, ряда простих інструментів створюється необхідна форма з сіткою, яка відповідає потребам та призначенню моделі. Коли об'єкт достатньо складний (має складні вирізи, переходи форм), просте моделювання може бути трудомістким, і тому оптимальним підходом для складних об'єктів є використання булевих операцій над моделлю, і виправлення артефактів сітки, які утворюються в процесі. Створюються додаткові моделі, які вирізають або доповнюють необхідний об'єм в основній моделі, при цьому виникають дефекти сітки, які

можуть спричинити артефакти при затіненні, а також ускладнювати подальшу роботу над моделлю. Для усунення цих дефектів виконується чистка сітки, автоматично за допомогою модифікаторів, частково вручну (об'єднанням вершин, створенням нових ребер, видаленням лишніх ребер, триангуляції).

Для складних органічних форм виправданим є процес скульптингу, та подальшої ретопології моделі. В основу скульптингу лягає чорнова модель. Скульптинг можна порівняти з класично ліпкою з глини або пластиліну, форма нарощується шарами, різними за дією пензлями. Під час скульптингу використовується високополігональна модель, яка дозволяє нарощувати складні форми та додавати велику кількість деталей, але така модель не підходить для подальшої роботи, так як потребує багато ресурсів для обробки, а текстурування в разі складніше, ніж при роботі з низькополігональними моделями. Тому після скульптингу виконується ретопологія, і сітка створюється повторно.

Моделі в Fruit and Juice виконані за допомогою моделінгу, в них немає комплексних об'єктів, які для створення потребують булевих операцій, а скульптинг для органічних об'єктів, представлених в грі, невиправданий їх простотою.

Для моделі ножа використано підхід простого моделінгу. На основі чорнової моделі створена модель ручки та леза, за допомогою фасок були реалізовані плавні переходи між гранями ручки та кривизна леза, додатково вони підправлені пропорційним редагуванням. На кінцях рукоятки створені гарда та тильник, додані заклепки. Модель розділена на декілька частин для можливості простого створення варіацій різних ножів, комбінуючи різні частини (рис. 6.5). Ідентичними методами створені всі інші моделі.

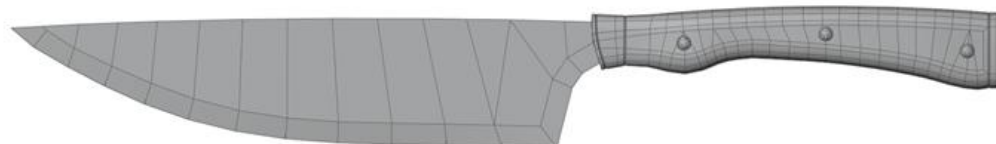


Рисунок 6.5 – Розроблена модель

6.3 Розгортка та використання колірної атласу

Після моделінгу створюється розгортка моделі для подальшого створення та накладання текстури. Поки немає розгортки немає інформації як накладати плоску текстуру на тривимірну модель, розгортка це представлення моделі на площині. В залежності від стилю процес розгортки та подальші етапи відрізняються. В даному проекті використовується метод розгортки для роботи з колірним атласом, що зумовлено стилем продукту.

Атлас (рис. 6.6) [6] представляє собою невелику текстуру, де певним чином розміщені набіри кольорових блоків або градієнтів, що надають інформацію про колір або градієнт частини об'єкта. На етапі створення такого атласу важливо приділити увагу угрупованню колірних блоків та їх назві. Також важливим є виділення основних кольорів або градієнтів, до яких будуть надсилатися відразу кілька об'єктів, важливо вибрати один напрямок для розміщення лінійних градієнтів, особливо якщо планується анімація текстури за допомогою її руху по якійсь осі.



Рисунок 6.6 – Приклад атласу

Для розгортки модель ділиться на частини та проектується на площину, після чого є декілька варіантів роботи з розгорткою, які дають різні результати. Перший – це переміщення кожної точки острова на необхідний колір. Так можна надати частині об'єкта суцільний колір (рис. 6.7). Для створення лінійного градієнту точки острова вирівнюються по одній осі (рис. 6.8). Можна використати просто проекцію частини об'єкта з певного ракурсу і розмістити отриманий острів на більш складному градієнті (рис. 6.9).

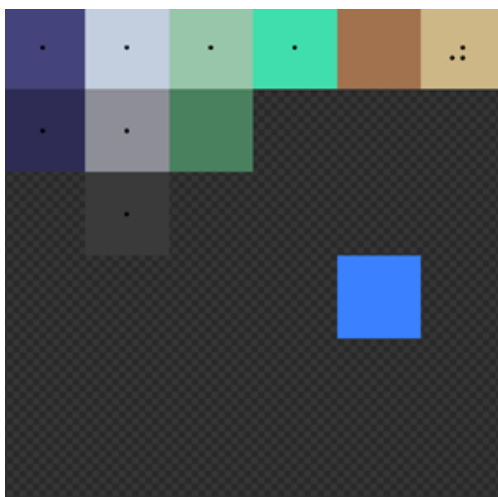


Рисунок 6.7 – Кожна точка це окремий острів

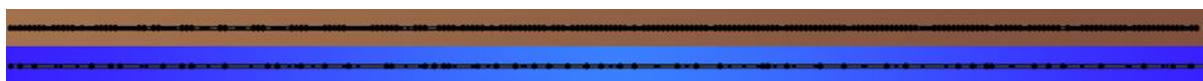


Рисунок 6.8 – Всі точки острова вирівняні по одній осі та розміщені на лінійному градієнті



Рисунок 6.9 – Розгортка створена проекцією моделі з одного ракурсу

Такий метод зменшує не тільки розмір текстури, яка використовується, але і кількість швів, та спрощує редагування текстур. Можливе і використання одного колірної атласу для всього асету, що дозволяє скоротити кількість звернень при завантаженні та вивантаженні рівнів, а також спростити налаштування самого асету, оскільки використовується один матеріал. Також плюсом є простота редагування такого атласу, що дозволяє швидко вносити правки в колір без необхідності мати спеціальні навички.

Мінусом є залежність деталізації від щільності сітки, так для створення деталей потрібно або ущільнювати сітку, формуючи з неї необхідний малюнок, або використовувати геометрію, що висить над основною (рис. 6.10).

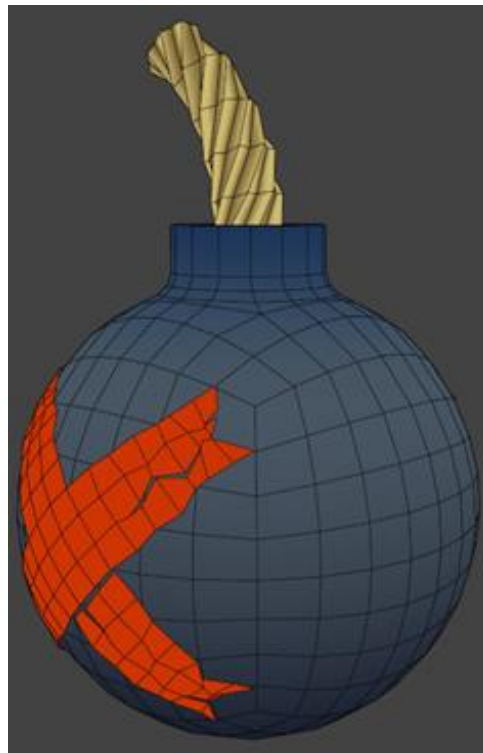


Рисунок 6.10 – Висяча геометрія надає деталізацію моделі, при цьому не порушується правильність топології самої моделі.

На моделі ножа було визначено шви (рис. 6.12), які розділять модель на розгортці, кожна частина моделі була спроектована на площину (рис. 6.11). Рукоятка має простий нелінійний градієнт, її розгортка отримана за допомогою простої проєкції з одного ракурсу. Лезо має два окремих острови, перший використовується для суцільного кольору, ця частина отримана проєкцією та переміщенням всіх вершин острова в одну точку на розгортці. Другий острів отриманий з допомогою проєкції, точки були розділені на дві групи та розміщені на кінцях градієнту, це створило необхідний градієнт на кромці леза. Гарда, тильник та заклепки були спроектовані на площину, де острови буди вирівняні по одній осі, після чого переміщені на окремий лінійний градієнт (рис. 6.11). Розгортка всіх інших моделей виконана за таким самим принципом.

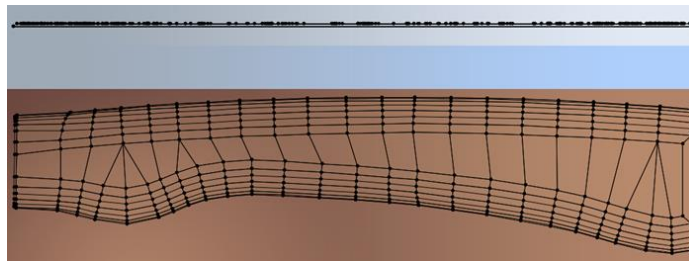


Рисунок 6.11 – Розгортка ножа

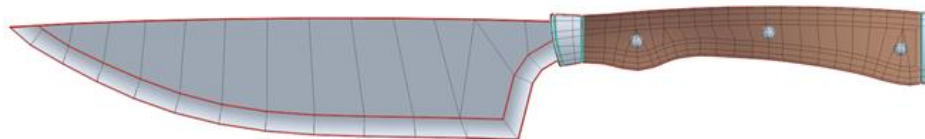


Рисунок 6.12 – Позначені шви та накладений атлас на модель

6.4 Додаткові налаштування та експорт

Для моделей створюється власний матеріал, проводиться перейменування та групування моделей; видаляються зайві дані, які були створені під час створення, матеріали, групи вершин, зайві розгортки.

Матеріали, які будуть використовуватись, створюються за атласом. На весь проект є три атласи: окремі для ножів, об'єктів, що розрізаються, та оточення. Три матеріали з відповідними назвами та підключеними текстурами, не етап експорту налаштування матеріалів не важливе, ключовою є тільки назва, та до яких об'єктів вони підключені.

Перевіряються параметри моделі такі, як розмір та орієнтація в просторі, орієнтація нормалей моделі. Це зменшує кількість помилок при використанні об'єктів в Unity, особливо, якщо вони анімуються. Файл-оригінал очищується від зайвої інформації, це полегшує наступну роботу при внесенні правок, асетуванні моделей та при створенні промо-матеріалів.

В залежності від типу об'єкту, він може проходити додаткові етапи створення, налаштування. Це характерно для персонажів, які використовують скелет та скелетні анімації. Для ножа та всіх інших моделей додаткові етапи не є необхідними, тому вони готові до експорту. На цьому етапі створюються групи моделей, які експортуються до Unity.

6.5 Налаштування ресурсів в Unity

Експортовані моделі налаштовують для правильного відображення в Unity, налаштування включає і прості опції такі, як модифікатори розміру або прив'язки матеріалу, так і складні, як налаштування імпорту нормалей або обробки груп вершин для анімації. Текстури мають додаткові налаштування, які відрізняються від стандартних. Так як використовуються колірні атласи, важливим є алгоритм стиснення зображень. Сильне стиснення викликає артефакти (рис. 6.13), а на текстурі малого розміру це критично, так як артефакти на моделі будуть помітними. Також включаються опції, які служать для оптимізації великих текстур, вони також викликають різного роду артефакти на різній відстані об'єкта від камери.

На основі атласів створюються ряд матеріалів для окремих груп об'єктів. Матеріал стилізований, параметри не опираються на реальні фізичні

параметри поверхонь (рис. 6.14). Налаштовується колір затінення, наскільки воно сильне та його згладжування, також налаштовується контурне затінення (rim) його сила та ступінь згладжування. На матеріал ніяк не впливають колір світла та оточення, враховується лише напрям світла, такий матеріал легше налаштовувати і він не залежить від оточення.



Рисунок 6.13 – Зліва артефакти стиснення атласу, справа – вірні налаштування стиснення

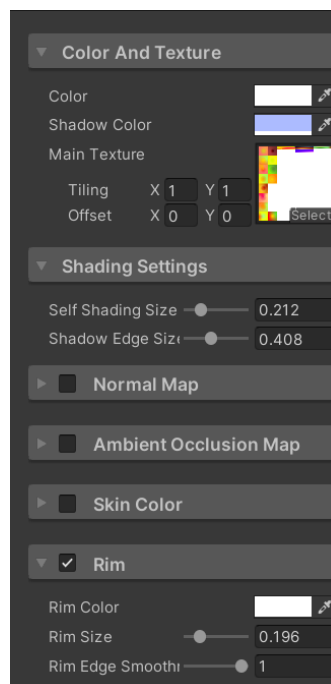


Рисунок 6.14 – Налаштування матеріалу

Експортовані моделі налаштовують для правильного відображення в Unity. Налаштування включає і прості опції такі, як модифікатори розміру

або прив'язки матеріалу, так і складні, як налаштування імпорту нормалей або обробки груп вершин для анімації. Спираючись на матеріали, створені в файлах-оригіналах, до моделей прив'язуються матеріали, створені в Unity. Це прискорює роботу вже під час створення префабів або під час внесення правок до моделей та їх оновлення.

На наступному етапі створюють префаби – попередньо згруповані та налаштовані об'єкти, які будуть використовуватись для побудови сцен. Об'єкти комбінуються до них, підключаються скрипти для анімації моделі або анімації матеріалу. Вони зберігаються окремо і дають можливість швидко змінити об'єкти на всіх рівнях, та сценах, що пришвидшує внесення правок.

Для ножу префаб включає в себе, по-перше, декілька колайдерів для взаємодії з об'єктами, що розрізаються, а також сам ніж, структура префабу дозволяє швидко змінювати об'єкти ножу (рис. 6.15).

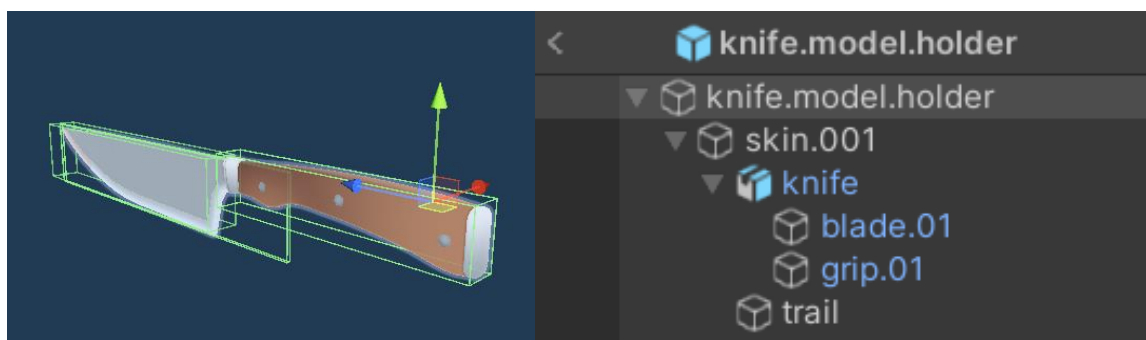


Рисунок 6.15 – Вигляд префабу, структура префабу

Налаштування сцени виконується на попередньо створеній, на якій виконувався чорновий варіант рівня. Тут примітиви замінюють на створені моделі та префаби, налаштовують освітлення та ефекти; додають моделі оточення, виставляють фінальне положення камери (рис. 6.16).



Рисунок 6.16 – Налаштована сцена

6.6 Промо-матеріали

Для промо-матеріалів створюється новий файл, до нього додаються необхідні об'єкти для створення композиції, як правило, нові об'єкти не створюються. Для промо матеріалів додатково налаштовуються матеріали, вони не мають сильно відрізнитись від представлених в грі, але для деяких поверхонь додається характеристика гладкості або прозорості для більшої природності.

З об'єктів створюється композиція, налаштовується камера (рис. 6.17). Виконується налаштування світла, воно залежить від вмісту та цілей. Найчастіше використовується зображення з широким діапазоном яскравості для загального освітлення, для створення художніх тіней або бліків, вже додатково використовуються вбудовані джерела світла. Після налаштування освітлення виконується налаштування рендеру: формат та розмір зображення, якість обчислення освітлення, які постпроцеси виконуються. Рендер також ділиться на декілька частин за необхідності для спрощення обробки в графічних редакторах.

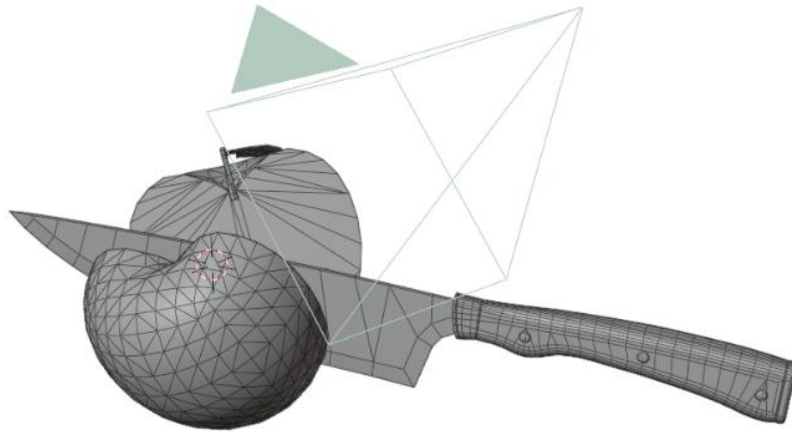


Рисунок 6.17 – Налаштована сцена

Після виконується рендер. Отримане зображення експортується до графічного редактору, де воно додатково обробляється, виконується корекція кольору, додаються текст, логотипи, обводка, свічення та інші ефекти. В графічному редактору також створюються інші за форматом промо матеріали.

Є два основні промо-матеріали, які створюються – це іконка (рис. 6.18) та постер (рис. 6.19). Для платформи розповсюдження вони відрізняються за форматом, але зазвичай створюються на основі одного зображення. Для постеру додається назва та логотип, якщо це необхідно.



Рисунок 6.18 – Іконка на різних етапах створення



Рисунок 6.19 – Постер для платформи розповсюдження

В промо-матеріалах Fruit and Juice використовуються метакулі для імітування рідини [7]. Так як симуляція трудоемка та займає багато часу, для створення рідини використано метакулі. Першочерговим було створено модель-каркас, кожна вершина якої програмними інструментами замінюється на метакулю. Таким чином створюється основний об'єм рідини та її форма. Після того, як основна форма створеного всплеску соку готова, вручну підправляється радіус метакуль та додаються окремі краплі. Такий метод імітування рідини ефективний для статичних кадрів, а для анімацій можна використовувати модифікацію методу з фізичною симуляцією (частинки, або тканина), що буде швидше ніж симуляція рідин.

7 ТЕСТУВАННЯ І ПУБЛІКАЦІЯ ЕЛЕКТРОННОГО ВИДАННЯ

Під час розробки виконується ряд тестувань, які мають покращити різноманітні аспекти проекту.

7.1 Дослідницьке тестування

Створення та налаштування контенту на грі виконується на великих моніторах, і це в деякій мірі спотворює сприйняття візуалу, ще в більшій мірі це відображається на управлінні (керування мишею та сенсорне мають значну різницю), тому під час розробки прийняте дослідницьке тестування, коли проект компілюється та тестується. Такі білди створюються на всіх етапах проектування, і вони служать ілюстрацією механік, управління, графіки, інтерфейсу безпосередньо на пристроях, для яких вони створюються. Тестування включає в себе запуск та перевірку того, що тестується, з наступним аналізом того, що виглядає або працює дивно. Аналіз цілком спирається на минулий досвід (інші ігри). Це достатньо суб'єктивне тестування, але під час розробки воно виправляє критичні недоліки.

Після тестування можна визначити, що потрібно змінити. Тестування саме графічного наповнення найчастіше виконується після створення базових механік, коли створена чорнова версія (блокаут), визначаються пропорції об'єктів, оптимальний розмір об'єктів взаємодії, як ці розміри сприймаються на малих екранах чи зручне управління. Також тестування виконується наприкінці розробки, тут аналізується вже цілісна картина зовнішнього вигляду. Аналізуються колірні схеми, деталі моделей, зрозумілість анімацій.

Для Fruit and Juice виконувалось кілька дослідницьких тестувань. На етапі чорнового варіанту були визначені розміри та пропорції ігрових об'єктів. В першу чергу – це визначення розміру об'єктів, які розрізаються:

визначення балансу між занадто маленькими, на які складно навестись, та занадто великими, які зменшують залучення гравця. Також визначається розмір переднього плану (блендер, скляна чаша), це зона, яка активну роль набуває лише по завершенню рівня, а під час гри вона несе лише інформативну роль, яка доповнює звичайний інтерфейс. Разом з пасивним переднім планом знизу розміщується рекламний банер, який також забирає площу екрану, тому потрібно чітко визначити розмір цих елементів, щоб зберегти якомога більше місця саме під активну зону.

Тестувався варіант з повним прибиранням переднього плану, таке рішення значно впливає на першочергове сприйняття користувачем гри. Прибираючи передній план, користувач отримує менше контексту, і не має або не розуміє цілі своїх дій. Таке позбавлення контексту також впливає на рекламу, де потрібно чітко показати суть та цілі потенційному користувачу.

На завершальній стадії дослідницькі тестування стосувалися колірних схем, ефектів бризг, анімацій.

Тестовані колірні схеми сконцентровані саме на темній темі, більш контрастному відображенню (рис. 7.1-7.2). Після було прийняте рішення залишитись на світлому варіанті. Це рішення обґрунтоване тим, що більш світлі кольори приваблюють більше, чисте наповнення не так сильно набридає з часом. Темний варіант збережений, і був використаний в одному з варіантів реклами. (При показі реклами рішення було підкріплене статистикою, світла версія реклами збирала більше переходів до гри, ніж темна.)

Предметом тестування були ефекти при розрізанні, потрібно визначити розмір бризг, їх щільність та швидкість, з якою вони зникають. Як і з розміром ключових об'єктів, ефекти мають бути оптимального розміру, щоб не заважати сприйняттю гри, одночасно з цим, ефекти також вносять віддачу гравцю, тому вони повинні бути достатньо помітними.

Тестування анімацій об'єктів, які вилітають, різні варіанти швидкості, обертання. Після тестувань була обрана комфортна швидкість, для бомби

було прибране обертання – це рішення полегшує її ідентифікацію, створило більший контраст між об'єктами, що можна розрізати, та тими, що не потрібно.



Рисунок 7.1 –Перший варіант теми

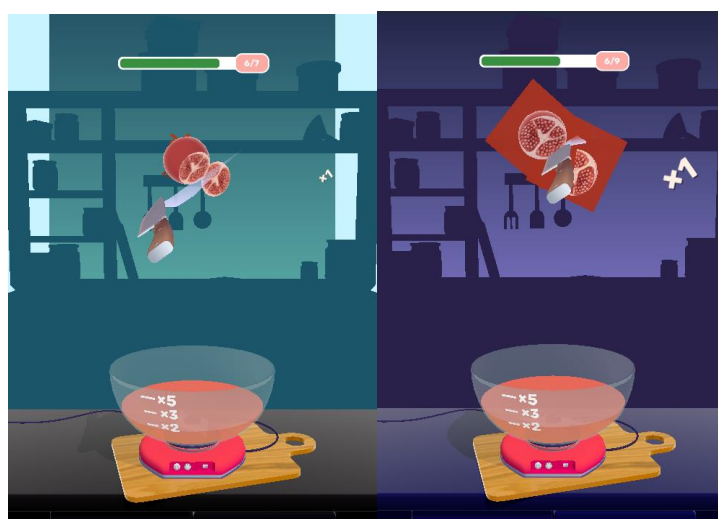


Рисунок 7.2 – Варіанти тем, які тестувались

7.2 Ігрове тестування

Дослідницькі тестування виконуються під час роботи, правки на їх основі визначаються розробником і частково суб'єктивні, для зменшення суб'єктивності виконується ігрове тестування (play testing) [8]. Тестування виконують не зв'язані за розробкою користувачі, відгуки також суб'єктивні, але їх більше. Якщо вибірка буде достатньо широкою, можна узагальнити

думки та скласти загальне враження. Тестування виконується на пізніх етапах розробки, білд повинен мати закінчений вигляд. Цілі тестування – визначити недоліки та проблеми, які можуть виникнути у користувача. Аналізуючи відгуки, можна визначити, які елементи сприймаються по-іншому, отримати враження від графіки, управління. Також формується список ідей з розвитку або ітерування. Окремо можна виділити коментарі про складність гри.

Для Fruit and Juice основний відгук стосується невеликої складності та однотипності рівнів. По завершенню тестування було вирішено додати криву складності для рівнів з невеликим збільшенням кількості об'єктів та бомб. Сильне ускладнення може відштовхнути зацікавлених на перших рівнях, тому тут витримано баланс, щоб відчувалась незначна прогресія і не було зовсім нудно та легко.

7.3 Функціональне тестування

Функціональне тестування виконується протягом всього циклу розробки, а також перед публікацією. Тестування зосереджене на правильному функціонуванні всіх елементів. Тестуються правильність роботи, яку було закладено при проектуванні, а також реакція на нестандартні, нелогічні дії зі сторони користувача. Отримані реакції можна поділити на коректні та некоректні, документується реакція та спосіб її досягнення. Реакціям надається статус критичності і вони групуються та виправляються за цим статусом. Критичність визначається впливом на процес гри, самим критичним є неможливість продовження гри (критичний збій, закриття гри, застрявання в інших об'єктах, відсутність реакції інтерфейсу тощо). Перед публікацією всі критичні функціональні помилки мають бути виправлені.

Під час розробки функціональні помилки виникали при генерації нової частини об'єкта, який був розрізаний, найефективнішим рішенням є

збільшення полігонажу моделей. Це частково впливає на оптимізацію, але повністю вирішує проблему неправильно згенерованих частин об'єктів.

Критичних помилок у Fruit and Juice на стадії релізу не було виявлено. Ряд виявлених функціональних проблем стосується специфічного положення ножа, при якому об'єкт, що має бути розрізаним, зависає в просторі і починає розрізатись до нескінченності, це було виправлено до публікації. (Ряд не суттєвих, складнодосяжних проблем був залишений на вирішення під час ітерування або оновлення, таке рішення дозволяє не витратити зайвий час на виправлення помилки, яку побачить менше процента користувачів, та швидше випустити продукт.)

7.4 Технічне тестування

Виконується під час розробки та надає інформацію про необхідні технічні ресурси для запуску та комфортної роботи гри на мобільних пристроях. Це тестування дає розуміння, наскільки гарно буде працювати гра на слабких пристроях. Можна виявити місця, які потребують оптимізації (моделі, шейдери, текстури, прорахунок фізики і т.п.). Тестування виконується вбудованими інструментами движку Unity і надає покадрову інформацію про внутрішнє функціонування, необхідні ресурси, час розрахунку кадру. (Таке тестування імітує ідеальні умови, яких в реальності не буде, потрібно враховувати, що крім гри на пристрої будуть запущені ще декілька програм, буде виконуватись ряд фонових процесів. Частина вільних ресурсів буде займати рекламний сервіс, який буде завантажувати рекламу.) Це дає змогу приблизно зрозуміти ступінь оптимізації роботи проекту, виявити слабкі місця, але швидкість функціонування буде завжди відрізнятись на різних пристроях при різних умовах.

Основні параметри технічного тестування, які стосуються графічного наповнення, є кількість викликів малювання (drawcalls) [9], кількість трикутників на сцені, кількість матеріалів та об'єм пам'яті, необхідний для

збереження текстур. У Fruit and Juice через використаний підхід до текстурування кількість матеріалів та текстур мінімальні. Полігонаж в рамках допустимого (більше того, на сцені немає персонажів із скелетною анімацією, що розширює стандартні рамки полігонажу). Для мобільних проектів ідеальною кількістю відмалювки вважається значення drawcalls між 70 та 100, на практиці гарно працюють значення до 150. У Fruit and Juice можна виділити два стани сцени, для яких значення значно різняться. Це стан спокою, коли вимальовуються вже створені об'єкти, виконуються нескладні анімації (рис. 7.3). Активний стан, коли при розрізанні об'єктів потрібно виконати більше операцій з відмалювки (рис. 7.4) (виконується генерація розрізаної частини, створюються окремі моделі розрізаних частин, накладається текстура зрізу, з'являється ефект бриз, для якого обчислюється прозорість). Тому аналізуються обидва стани (табл. 7.1).

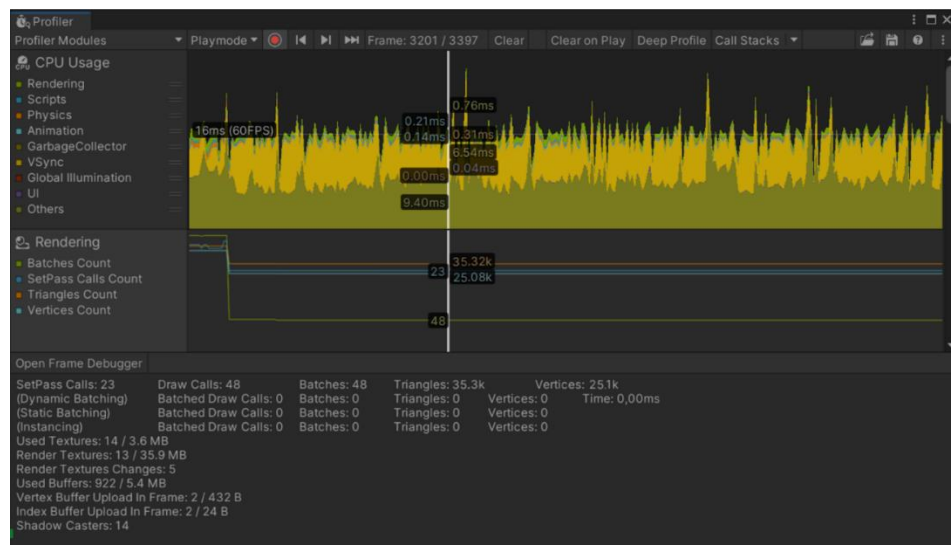


Рисунок 7.3 – Кількість drawcalls під час спокою

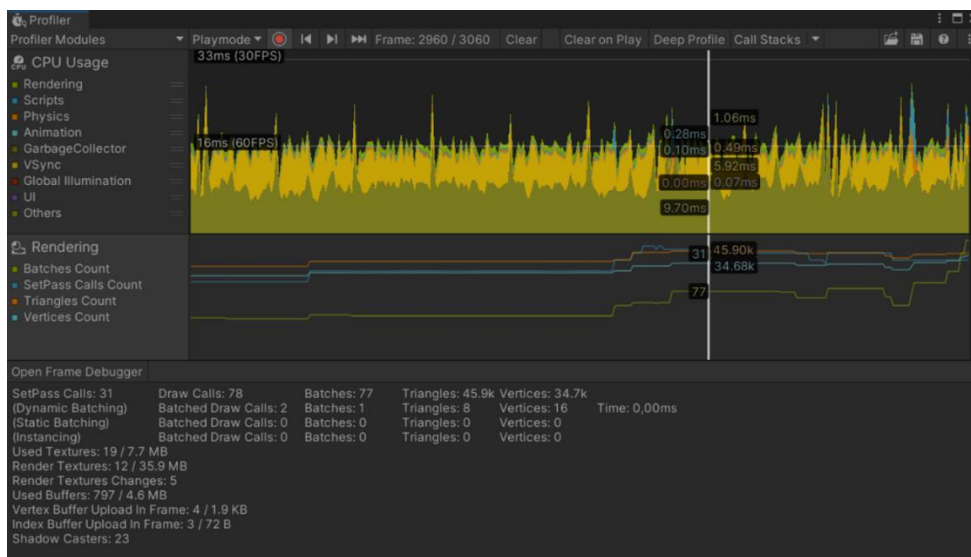


Рисунок 7.4 – Кількість drawcalls під час генерації об’єктів

Таблиця 7.1 – Показники кількості викликів малювання для різних режимів гри

	В простій	Під час розрізання	Середні значення в аналогах
Кількість відмальовок	48 ± 10	77 ± 10	80 – 120
Кількість трикутників	35k ± 5k	45k ± 5k	< 50k

7.5 Публікація та установка

Проект розроблений для платформи Android версії 5.0 та вище, графіка оптимізована для застарілих та слабких пристроїв, та має правильно відображатись на екранах різної роздільної здатності та співвідношення. Гра успішно опублікована на Google Play (рис. 7.5). Успішно завантажити та встановити можна на пристрої з відповідною версією операційної системи. Також встановити та успішно запустити можна на емуляторах платформи Android.

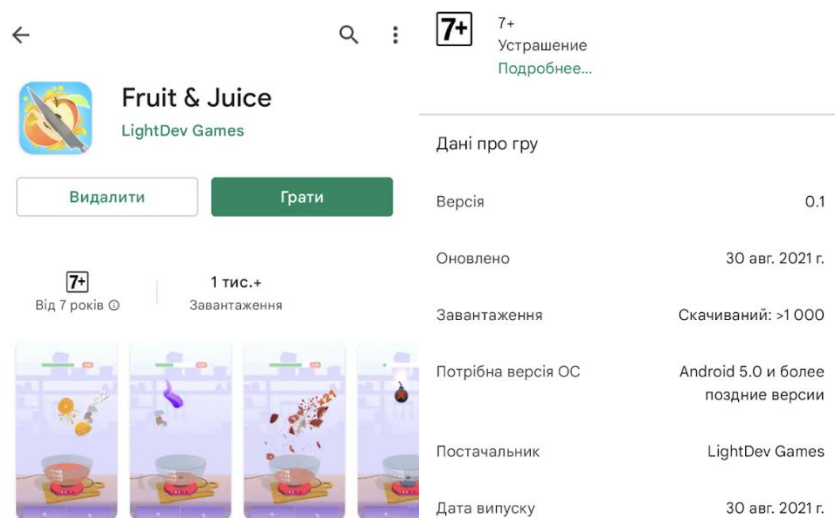


Рисунок 7.5 – Сторінка на Play Market

Після публікації було проведено тестування на мобільних пристроях та планшетах на операційній системі Android з різним співвідношенням сторін та розмірами екранів. Пристрої, на яких проводилось тестування: Google Pixel 4 на Android 13 з дисплеєм 5,7 дюймів роздільною здатністю 2280x1080 точок та Xiaomi Pad 5 Android 11 з дисплеєм 11” роздільною здатністю 2560x1600 точок. Результати тестування представлені на рис. 7.6 та рис. 7.7 відповідно. Гра відкривається та працює без нарікань, інтерфейс правильно масштабується, та залишається повністю функціональним.



Рисунок 7.6 – Робота гри на екрані 5,7 дюймів, 2280x1080 точок

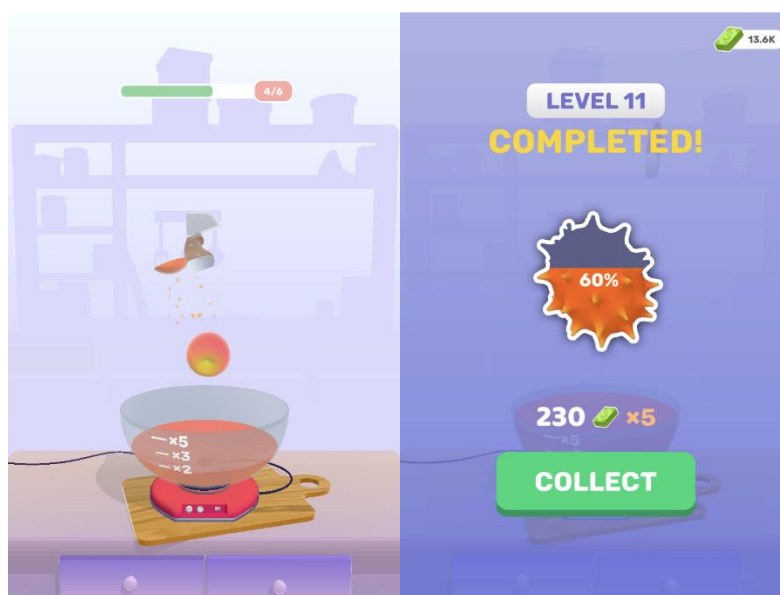


Рисунок 7.7 – Робота гри на екрані 11 дюймів, 2560x1600 точок

8 ЕКОНОМІЧНА ЧАСТИНА

У результаті виконання кваліфікаційної роботи розроблена гра Fruit and Juice, опублікована на платформах розповсюдження та монетизується за рахунок показу реклами між ігровими сесіями.

Економічна ефективність проекту.

Економічна ефективність проекту розраховується перед розробкою, та надає інформацію про необхідні ресурси. Також дає можливість оцінити ефективність підходу до проектування порівняно з аналогічними проектами.

Розглянемо переваги проектованої гри.

Гіперказуальний жанр гри спрямований на широку аудиторію, так як немає ніякої прив'язки до інтересів або вимог до навичок гравця. Це збільшує потенційні скачування і рентабельність розробки.

При розробці використовується ефективний виробничий процес, при якому продукт набуває фінального вигляду за короткий період часу, що дає змогу швидко створити та протестувати концепцію гри.

При розробці використовуються програмне безкоштовне забезпечення що зменшує витрати на розробку.

Розглянемо джерела економії, доходу, джерела фінансування.

Для фірми-розробника джерелом доходу є розробка та підтримка гри певний час після публікації. Витрати фірми містять в собі витрати на розробку гри. Джерелом фінансування є видавець.

Для видавця джерелом фінансування є монетизація за рахунок показу реклами в іграх. Витрати видавця складаються з виплат багатьом фірмам-розробників та рекламуванні розроблених ігр. Економія в тому, що безпосередньо розробка ігр лежить на окремих багаточисленних групах розробників. Організація виробничого процесу, менеджмент ресурсів та задач по розробці гри лежить також на окремих фірмах-розробниках.

Розподілення задач розробки та видавництва дають значні переваги обом сторонам і є більш ефективною моделлю, ніж поєднання розробника та видавця з боку однієї компанії.

Розглянемо порядок виробничого процесу створення гри.

У загальному випадку розробка гри містить у собі наступні етапи:

– підготовчий етап, на якому створюється технічне завдання, визначаються виробничі процеси;

– етап розробки;

– етап розробки супроводжувальних матеріалів;

– заключний етап архівування, асетування матеріалів проекту.

Здійснимо розрахунок собівартості і ціни розробки гри.

У собівартість розробки гри входять наступні статті витрат:

– основна заробітна плата;

– додаткова заробітна плата;

– єдиний соціальний внесок;

– інші витрати.

Розробку гри проводять три фахівці: розробник Unity3D, дизайнер інтерфейсів та 3d художник. Зарплата програміста становить 235 грн./год, дизайнера інтерфейсів – 165 грн./год, 3d художника – 175 грн./год. При цьому тривалість робочого дня кожного з них становить 8 годин. Гра розробляється 7 днів.

Розрахунок основної заробітної плати наведено в табл. 8.1.

Додаткова заробітна плата – це винагорода за працю понад установлені норми, за трудові успіхи та винахідливість і за особливі умови праці. Вона включає доплати, надбавки, гарантійні та компенсаційні виплати, передбачені чинним законодавством; премії, пов'язані з виконанням виробничих завдань і функцій [10]. У даному випадку додаткова заробітна плата становить 20 % від основної:

$$28320 * 0,2 = 5664 \text{ грн.}$$

Таблиця 8.1– Розрахунок витрат на заробітну плату

Етап	Вид робіт	Виконавець		Годинн а ставка	Трива- лість вико- нання, дні	Заробітн а плата, грн.
		Кіль- кість	Посада			
1. Підготовчий	Формулювання ТЗ та задач	3	розробник	235	1	1880
			3d-художник	175	1	1400
			дизайнер інтерфейсів	165	1	1320
2. Розробка	Розробка програмної частини	1	розробник	235	5	9400
	Розробка графічного наповнення	1	3d-художник	175	5	7000
	Розробка інтерфейсу	1	дизайнер інтерфейсів	165	2	2640
3.Розробка супроводжувальних матеріалів	Розробка іконки та постеру	1	3d-художник	175	1	1400
4. Архівування, асетування матеріалів проекту	Асетування моделей	1	3d-художник	175	1	1400
	Оновлення фреймворку	1	розробник	235	1	1880
Разом					8	28320
Додаткова заробітна плата (20 %)						5664
Усього						33984

Ставка єдиного соціального внеску становить 22 % від величини основної і додаткової заробітної плати:

$$33984 * 0,22 = 7476 \text{ грн.}$$

До інших витрат слід віднести витрати на обслуговування ЕОМ і плату за електроенергію.

Витрати на електроенергію розраховуються, виходячи зі споживаної потужності устрою і тарифу на електроенергію. У даному випадку передбачається використання 3 комп'ютерів з потужністю 0,5 кВт/год. Вартість однієї кВт/год електроенергії прийнято у розмірі 1,68 грн. Час використання електроенергії в процесі розробки:

$$8 * 8 = 64 \text{ години.}$$

Отже, плата за електроенергію складе:

$$0,5 * 1,68 * 64 * 3 = 161,28 \text{ грн.}$$

Витрати на обслуговування ЕОМ визначаються з вартості ЕОМ і часу її експлуатації, після закінчення якого, вона підлягає заміні (звичайно цей час не перевищує 3-х років), протягом року ЕОМ використовує 254 робочих дні. Отже:

$$(25000 / (3 * 8 * 254)) * 64 * 3 = 787,5 \text{ грн.}$$

Розрахуємо суму прибутку від реалізації розробки (виходячи з рівня рентабельності 30 %):

$$42408,78 * 0,3 = 12722,63 \text{ грн.}$$

Розрахуємо ціну розробки гри без податку на додану вартість (ПДВ):

$$42408,78 + 12722,63 = 55131,01 \text{ грн.}$$

Розрахуємо суму ПДВ, вона рівна 20% від ціни:

$$55131,01 * 0,2 = 11026 \text{ грн.}$$

Розрахуємо ціну гри з урахуванням ПДВ:

$$55131,01 + 11026,20 = 66157,21 \text{ грн.}$$

Результати розрахунків наведено у табл. 8.2.

Таблиця 8.2 – Розрахунок витрат на проектування та ціни гри

Стаття витрат	Сума, грн.
Основна заробітна плата	28320,00
Додаткова заробітна плата	5664,00
Єдиний соціальний внесок	7476,72
Витрати на обслуговування ЕОМ	787,5
Витрати на електроенергію	161,28
Собівартість розробки гри	3819,87
Прибуток (рівень рентабельності 30 %)	12722,63
Ціна без ПДВ	55131,01
Податок на додану вартість (ПДВ)	11026
Ціна з урахуванням ПДВ	66157,21

На основі розрахунків повна вартість розробки гри складає 66157,21 грн. Розробка виконується протягом 8 робочих днів командою з трьох фахівців: розробника Unity3D, 3d художника та дизайнера інтерфейсів. Очікувана сума прибутку складає 12722,63 грн.

ВИСНОВКИ

В ході роботи було розроблено мобільну гру в гіперказуальному жанрі «Fruit and Juice».

В ході роботи розроблено графічне наповнення для мобільної гри гіперказуального жанру, виконано налаштування контенту в ігровому движку. Розроблено графічні супутні матеріали для гри. Було розглянуто та обране програмне забезпечення, яке використовується для проектування та менеджменту контенту. Під час розробки використані оптимальні підходи до структури проекту та розробки матеріалів. Продукт тестувався під час розробки для поліпшення кінцевого результату, та після завершення для перевірки правильної роботи на різних пристроях. Гра була протестована та успішно опублікована. Було обгрунтовано економічну доцільність розробки.

Стиль графічного рішення розроблений на основі жанрових особливостей та мети проекту. Розроблений стиль легко зчитується та не сильно напружує гравця, розроблений контент цілісний за стилем і виконаний за одним виробничим процесом. З технічного боку було використано оптимальні методи створення, які спрямовані на оптимізацію виробничого процесу та простоту внесення змін на всіх етапах.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. What Are Hypercasual Games? URL: <https://adjoe.io/glossary/hypercasual-games/> (дата звернення: 23.05.2022).
2. Chto takoe giperkazual`ny`e igry` i pochemu oni populyarny`. URL: <https://media-xyz.com/ru/articles/794-chto-takoe-giperkazualnye-igry-i-pochemu-oni-p> (дата звернення: 23.05.2022).
3. What is UV Mapping & Unwrapping? URL: <https://conceptartempire.com/uv-mapping-unwrapping/> (дата звернення: 23.05.2022).
4. Головні метрики. Lifetime. URL: <https://www.devto.dev.com/education/articles/ru/212/glavnie-metriki-lifetime> (дата звернення: 23.05.2022).
5. Quick Start Guide to Blockouts. URL: https://www.worldofleveldesign.com/categories/level_design_tutorials/guide-to-blocktober.php (дата звернення: 23.05.2022).
6. Зуєвський Д.Р., Кулішова Н.Є. Створення кольорного атласу, особливості створення розгорток для його використання // Поліграфічні, мультимедійні та web-технології: матеріали Молодіжної школи-семінару VII Міжнар. наук.-техн. конф. (17-21 травня 2022, м. Харків). 2022. Т. 2. С. 118-130.
7. Lyashenko V., Deineko Zh., Zeleniy O., Tabakova I. Wavelet ideology as a universal tool for data processing and analysis: some application examples // International Journal of Academic Information Systems Research (IAISR). – 2021. – Vol. 5(9). – pp. 25-30.
8. 7 Different Types of Game Testing Techniques. URL: <https://www.qable.io/7-different-types-of-game-testing-techniques/> (дата звернення: 23.05.2022).
9. Draw Call Batching. URL: <https://docs.unity3d.com/ru/2019.4/Manual/DrawCallBatching.html> (дата звернення: 23.05.2022).
10. Полозова Т.В. Економіка та бізнес: комплекс навчально-методичного забезпечення. Харків: ХНУРЕ, 2022. 47 с.