

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему: «Пізнавальний ігровий додаток для дітей дошкільного віку»

за спеціальністю 122 «Комп'ютерні науки»,
освітньо-професійна програма «Інформаційні технології
проектування»

Виконавець роботи: студент групи ІТ-82-0 Яценко Сергій Віталійович

**Кваліфікаційна робота бакалавра
захищена на засіданні ЕК
з оцінкою**

_____ «__» _____ 2022 р.

Науковий керівник

(підпис)

к.т.н., Бойко О.В
(науковий ступінь, вчене звання, прізвище та ініціали)

Засвідчую, що у цій дипломній роботі немає
запозичень з праць інших авторів
без відповідних посилань.

Студент _____
(підпис)

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра інформаційних технологій
Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ

Зав. кафедри ІТ

_____ В. В. Шендрик
«__» _____ 2022 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА СТУДЕНТУ

Яценко Сергій Віталійович

1 Тема роботи Пізнавальний ігровий додаток для дітей дошкільного віку

керівник роботи Бойко Ольга Василівна, к.т.н. _____,

затверджені наказом по університету від «27»квітня 2022 р.
№0301-VI

2 Строк подання студентом роботи «13» червня 2022 р.

3 Вхідні дані до роботи Технічне завдання

4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Вступ, аналіз предметної області, огляд останніх досліджень і публікацій, аналіз програмних продуктів – аналогів, постановка задачі, моделювання та проектування, моделювання ігрового додатку в нотації IDEF0, діаграма використання ігрового додатку, проектування інформаційної системи, розробка програмного додатку, архітектура програмного додатку, програмна реалізація, використання програмного додатку, висновки.

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Контекстна діаграма з точки зору гравця, декомпозиція першого рівня в нотації IDEF0, діаграма варіантів використання, діаграма діяльності,

налаштування проекту в Unity, компоненти головного меню, компонентипанелі вибору тематики рівнів, реалізація панелі менеджменту рівнів, реалізація панелі налаштувань, реалізація рівня зі складання слів, структура з якою взаємодіє скрипт спавну, компоненти рівня з використання спавну, реалізація рівня з поворотом пазлів, робота менеджера рівнів, робота менеджера рівнів, після проходження першого рівня, використання першого рівня в ігровому додатку, робота рівня зі складанням слів, використання панелі налаштувань.

6. Консультанти розділів роботи:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7.Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз предметної області	24.01.2022-15.02.2022	
2	Вибір середовища	16.02.2022-23.02.2022	
3	Створення та пошук спрайтів	24.02.2022-10.04.2022	
4	Створення рівнів	07.03.2022-12.05.2022	
5	Програмування рівнів	14.03.2022-27.05.2022	
6	Тестування ігрового додатку	19.05.2022-02.06.2022	
7	Створення документації	03.06.2022-10.06.2022	

Студент

(підпис)

Яценко С.В.

Керівник роботи

(підпис)

к.т.н. Бойко О.В.

РЕФЕРАТ

Тема роботи «Пізнавальний ігровий додаток для дітей дошкільного віку»

Пояснювальна записка 36 с., 29 рисунків, 20 літературних джерел, 1 таблиця, 3 додатки.

Метою роботи є теоретично обґрунтувати, розробити та апробувати пізнавальний ігровий додаток для дітей дошкільного віку засобами сучасних технологій навчання. Під час роботи над кваліфікаційною роботою бакалавра було досліджено аспекти навчання дітей, проблеми, які існують при створенні ігрового додатку для дітей дошкільного віку, визначено мету та задачі та була виконана постановка задачі.

Проведено аналіз існуючих програмних продуктів, існуючих моделей, методів, технологій та виявлена необхідність у створенні додатку.

Виконано аналіз і проектування ігрового додатку, вибір засобів реалізації та програмна реалізація ігрового додатку.

Ключові слова: ігровий додаток, діти дошкільного віку, Unity, пазли, Android додаток.

ЗМІСТ

ВСТУП	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	8
1.1 Огляд останніх досліджень і публікацій	8
1.2 Аналіз програмних продуктів - аналогів	9
1.3 Постановка задачі	10
2 Моделювання та проектування	12
2.1 Функціональне моделювання ігрового додатку	12
2.2 Діаграма використання ігрового додатку	14
2.3 Діаграма діяльності	16
3 Розробка ігрового додатку	18
3.1 Архітектура програмного додатку	18
3.2 Програмна реалізація	25
3.3 Використання програмного додатку	29
ВИСНОВКИ	33
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	34
ДОДАТОК А	37
ДОДАТОК Б	43
ДОДАТОК Г	55

ВСТУП

У наше століття цифрових технологій неможливо уявити процес виховання і розвитку дитини без ігор в смартфоні чи комп'ютері. Останнім часом гра є найсприятливішою формою соціального контакту, найсуттєвіші з яких найчастіше відбуваються саме у грі. Вона є необхідним засобом соціального життя, об'єктивною основою існування. Незважаючи на всі свої мінуси, ігри мають і свої плюси. Головне - це підібрати хорошу гру для розвитку. Яскраві зображення допомагають малюку вивчити кольори, геометричні фігури, цифри і букви. Навчальні ігри покращують логічне мислення дітей, підвищують ерудицію. Дошкільний вік - це етап психологічного розвитку дітей, що охоплює період від 3 до 6-7 років. Він характерний тим, що провідною діяльністю є гра, як важливий елемент формування особистості дитини.

Дошкільне дитинство – найбільш сприятливий вік для розвитку дитини. Як вказують вчені, у цей період дитина легко запам'ятовує нову інформацію. Дитина не ставить перед собою за мету – щось запам'ятати, а прагне пізнати навколишній світ [1].

Гра - це відображення реального світу. Вона має великий потенціал у навчанні дітей. Одним з видів гри є пізнавальна. В ній пізнавальні завдання поєднано з ігровими. Через гру дитина, граючи, вчиться, при цьому поглиблюються вивчені знання та уявлення. Як правило, ігрові додатки розвивають конкретні навички. Наприклад, знайомство з літерами, навички у математиці чи розвиток пам'яті [2].

Актуальність розробки ігрового додатку для дошкільнят обумовлюється сприянням загальному розвитку дітей, їх уваги, швидкості реакцій, тренування пам'яті. Великою перевагою таких додатків при підготовці до школи є розвиток посидючості та цілеспрямованості.

Ринок навчальних ігрових додатків швидко отримує нові і нові продукти. Однак, як правило, у розробників на першому місці стоїть заробіток а не якість додатку.

Також встановлено, що більшість додатків доступних в Україні англо-або російськомовні. Таким чином, дитині часто буває складно в такі ігри грати.

Грунтуючись на вищезазначених фактах, можна сказати, що ігрові додатки зараз мають велику актуальність, не зважаючи на велику кількість додатків у цій ніші [3].

Метою роботи є теоретично обґрунтувати, розробити та апробувати пізнавальний ігровий додаток для дітей дошкільного віку. У проекті необхідно створити простий, зручний та приємний інтерфейс та розробити звуковий супровід та озвучування завдань, що зробить гру більш «живою».

Для досягнення поставленої мети потрібно:

- виконати аналіз предметної області для різних вікових груп дошкільників;
- провести огляд програмних засобів розробки пізнавальних ігрових додатків, як засобів розвитку дошкільників;
- розробити постановку задачі;
- провести функціональне моделювання ігрового додатку;
- розробити діаграму варіантів використання;
- спроектувати і реалізувати пізнавальний ігровий додаток для навчання та виховання дітей.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд останніх досліджень і публікацій

У звіті за дослідженням проведеним дослідниками з Центру дітей та сімей FIU`s зазначено, що навчальні додатки можуть бути корисні для дітей раннього віку [4]. В цих досліджах брали участь майже 5000 дітей, які грали в пізнавальні додатки. Дослідження показало, що краще за все дітям допомагають отримувати нові знання математичні ігри, після них йдуть ігри з вивчення букв та правопису. Провідний автор, Шейл Гриффіт зазначив, що це перше дослідження, яке спирається на результати, отримані під час навчання дітей раннього віку за допомогою інтерактивних освітніх додатків. Також в цьому дослідженні зауважується, що краще, коли батьки також приймають участь у грі разом з дитиною. Це дає можливість контролювати час, який діти проводять за гранням та покращує повчальну властивість додатків для дітей.

Інше дослідження підтверджує той факт, що спільна гра батьки з дітьми в навчальні ігрові додатки є кращим варіантом [5]. Дженніфер Зош зазначила, що ігри не повинні замінювати спілкування з реальними людьми та взаємодію за світом. «Батьки не повинні автоматично довіряти товарам з позначкою «Освітній»», - сказала Зош. Під час сумісної гри з дітьми з використанням додатків, необхідно розмовляти з ними про те, що відбувається під час гри, вказувати на аналогії реального світу. Це допоможе зорієнтуватися в новому цифровому дитинстві.

Отже, проаналізувавши два дослідження ми можемо зробити висновки, що ігрові додатки можуть нести повчальні функції, але набагато менші ніж спілкування з батьками та взаємодією з навколишнім середовищем. Тому краще, коли діти проводять час в навчальних іграх з батьками – це помножує ефект навчання від додатка.

1.2 Аналіз програмних продуктів - аналогів

Під час виконання дипломного проектування було знайдено ігрові додатки, які могли частково вирішити поставлені задачі. Далі було вибрано три додатки, які були досліджені.

«TinyPuzzle» – android-додаток, який має розвиваючі ігри різних категорій для дітей дошкільного віку. Цей додаток розрахований на розвиток пам'яті та логіки. Недоліком даного додатку є те, що він англomовний [6].

Гра «Baby shark Jigsaw puzzle fun». Гра з гарним дизайном, який приваблює дітей, але має однотипні рівні. Ця гра може допомогти дітям розвинути короточасну пам'ять та когнітивні навички. На мою думку однотипні рівні можуть швидко набриднути, а також це суттєво обмежує навички, які дитина може здобути під час гри [7].

«Kids puzzle» – навчальний додаток для дітей, основна ідея якого заснована на складанні тварин з пазлів, слухаючи звуки тварин. Гра допомагає розвивати пам'ять, увагу, логічне мислення та моторику рук. Але основною проблемою також є однотипність рівнів, які дозволяють вивчати тільки тварин [8].

У таблиці 1.1 представлена порівняльна характеристика зазначених додатків.

Таблиця 1.1 - Порівняння показників ігрових додатків

Показник/ можливість	«Tiny Puzzle»	«Baby shark Jigsaw puzzle fun»	«Kids puzzle»
Гнучкість	+	+	-
Вибір тематики	-	-	-
Головне меню	+	+	+

Продовження таблиці 1.1.

Українська локалізація	-	-	+
Звуковий супровід	+	+	+
Туроріал	+	-	-
Генерація рівнів	-	-	-

1.3 Постановка задачі

Розвиток логічного мислення у дітей ефективно покращує використання пізнавальних ігрових додатків з яскравими картинками і казковими героями. Адже цільовою аудиторією / користувачами є дошкільнята віком від 3 до 6 років [9].

Метою розробки є теоретично обґрунтувати, розробити та апробувати пізнавальний ігровий додаток для дітей дошкільного віку. На даний момент є потреба у розробленні ігрових додатків, які б могли забезпечити допомогу у розвитку дитини в різних напрямках, задовольняючи базові потреби у дошкільнят [10].

Для досягнення поставленої мети були окреслені такі завдання:

- виконати аналіз предметної області для різних вікових груп дошкільників;
- провести огляд програмних засобів розробки пізнавальних ігрових додатків, як засобів розвитку дошкільників;
- розробити постановку задачі;
- провести функціональне моделювання ігрового додатку;
- розробити діаграму варіантів використання;

- спроектувати і реалізувати пізнавальний ігровий додаток для навчання та виховання дітей.

Об'єктом дослідження є розробка пізнавального ігрового додатку для дітей дошкільного віку. Предмет дослідження - технології створення пізнавального додатку.

Також, варто звернути увагу на те, що якість контенту, яким наповнені пізнавальні ігрові додатки бажає кращого. Після аналізу програмних продуктів, було зроблено висновок, що всі продукти мають значні недоліки і багато з них не дуже схвальні відгуки від користувачів.

Технічне завдання на розробку продукту у повному обсязі наведено у додатку А.

Додаток буде орієнтуватися на дітей дошкільного віку, тому проект повинен мати простий інтерфейс і звуковий супровід, щоб гра здавалася більш веселою [11].

2 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ

2.1 Функціональне моделювання ігрового додатку

Моделювання ігрового додатку в нотації дозволить досить просто, графічно, показати опис систем, процесів та ресурсів, які потрібні для створення даного програмного продукту [12].

Функціональне моделювання ігрового додатку в нотації у вигляді контекстної діаграми представлено на рисунку 2.1.



Рисунок 2.1 – Контекстна діаграма з точки зору гравця

На діаграмі функціонального моделювання ігрового додатку в центральному блоці представлено функцію, яку повинен виконувати додаток.

Стрілками зліва показані вхідні ресурси з якими взаємодіє додаток, а саме бажання цікаво провести час. Стрілки знизу показують механізми, що

використовуються для роботи додатку (програмна та апаратна частини). Стрілки зверху показують механізми управління такі, як методи, правила, стандарти. На виході (стрілки праворуч) ми отримуємо те, що створив додаток, а саме розвиваючий досвід для дітей.

Декомпозиція дозволить поступово та структуровано перейти від великих складових до більш дрібних. Кожна дія розбивається на більш прості операції з переробки певної інформації і в кінцевому результаті ми повинні отримати на виході той самий продукт діяльності додатку.

Декомпозиція контекстної діаграми ігрового додатку у нотації IDEF0 представлена на рисунку 2.2.

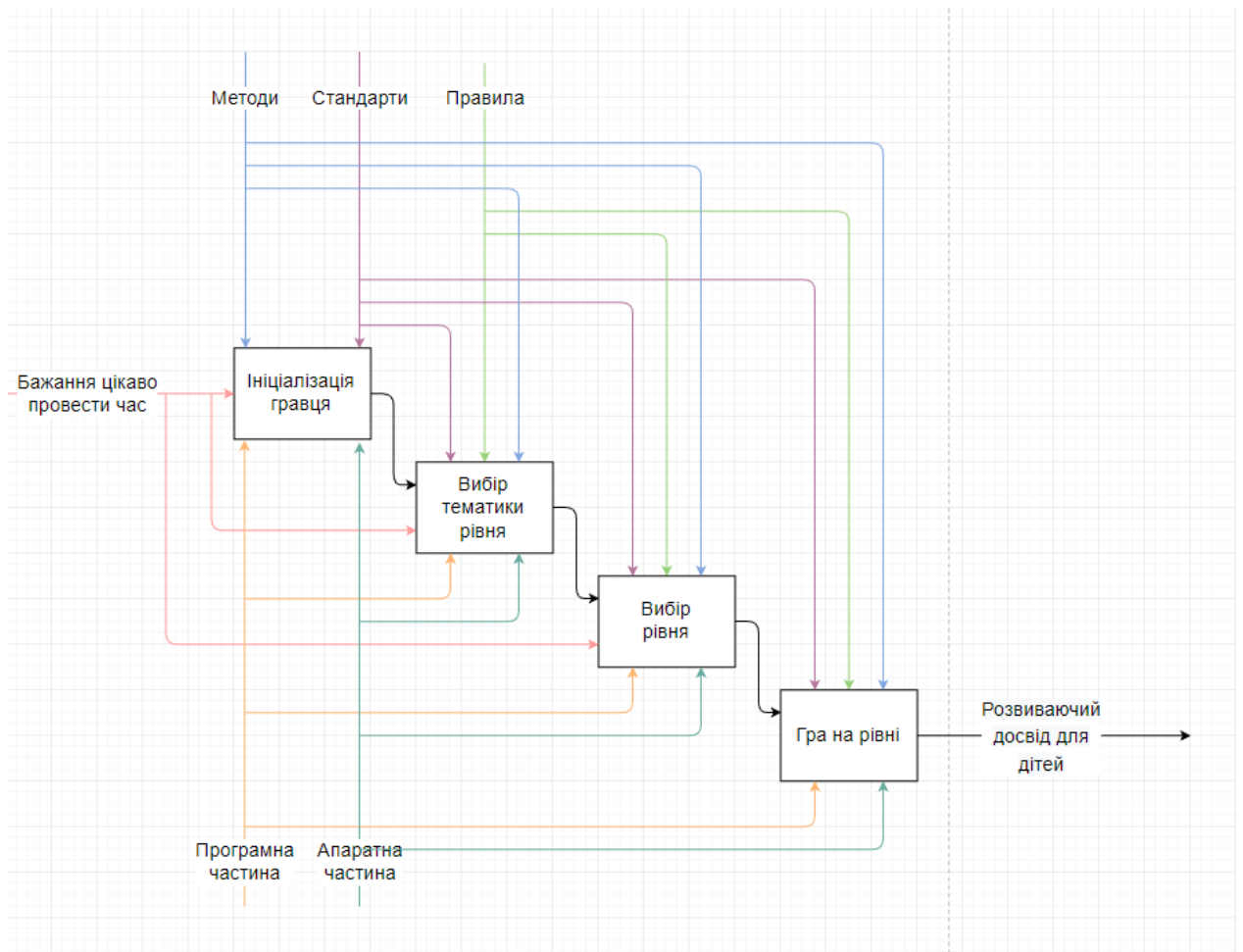


Рисунок 2.2 – Декомпозиція першого рівня в нотації IDEF0

В першому блоці даної діаграми представлена функція ініціалізації гравця, в другому блоці представлено функцію вибору тематики рівня, наступний блок показує функцію вибору рівня, останній блок відображає функцію гри на рівні. Бажання цікаво провести час - вхід з яким взаємодіють блоки ініціалізації, вибору тематики рівня та вибору рівня. Програмна та апаратна частини - механізми, що призначені для створення досвіду для дітей, ці механізми взаємодіють з усіма блоками. Методи, правила, стандарти – механізми управління. Методи і стандарти взаємодіють з усіма блоками, правила з блоками «Вибір тематики рівня», «Вибір рівня» та «Гра на рівні». На виході маємо розвиваючий досвід.

2.2 Діаграма використання ігрового додатку

Моделювання - пов'язані між собою етапи. Процес моделювання починається з концептуальної схеми, після якої створюються логічна та фізична моделі.

Для цього будується діаграма варіантів використання (use-case diagram), яка призначена для опису функціонального призначення системи. Концептуальну модель системи під час проектування забезпечує саме діаграма використання [13].

Вона представлена у вигляді безлічі акторів, що взаємодіють з системою ззовні за допомогою «варіантів використання». Варіанти використання показують дії, які виконуються програмою при взаємодії з актором.

Діаграма варіантів використання в нотації UML представлена на рисунку 2.3.

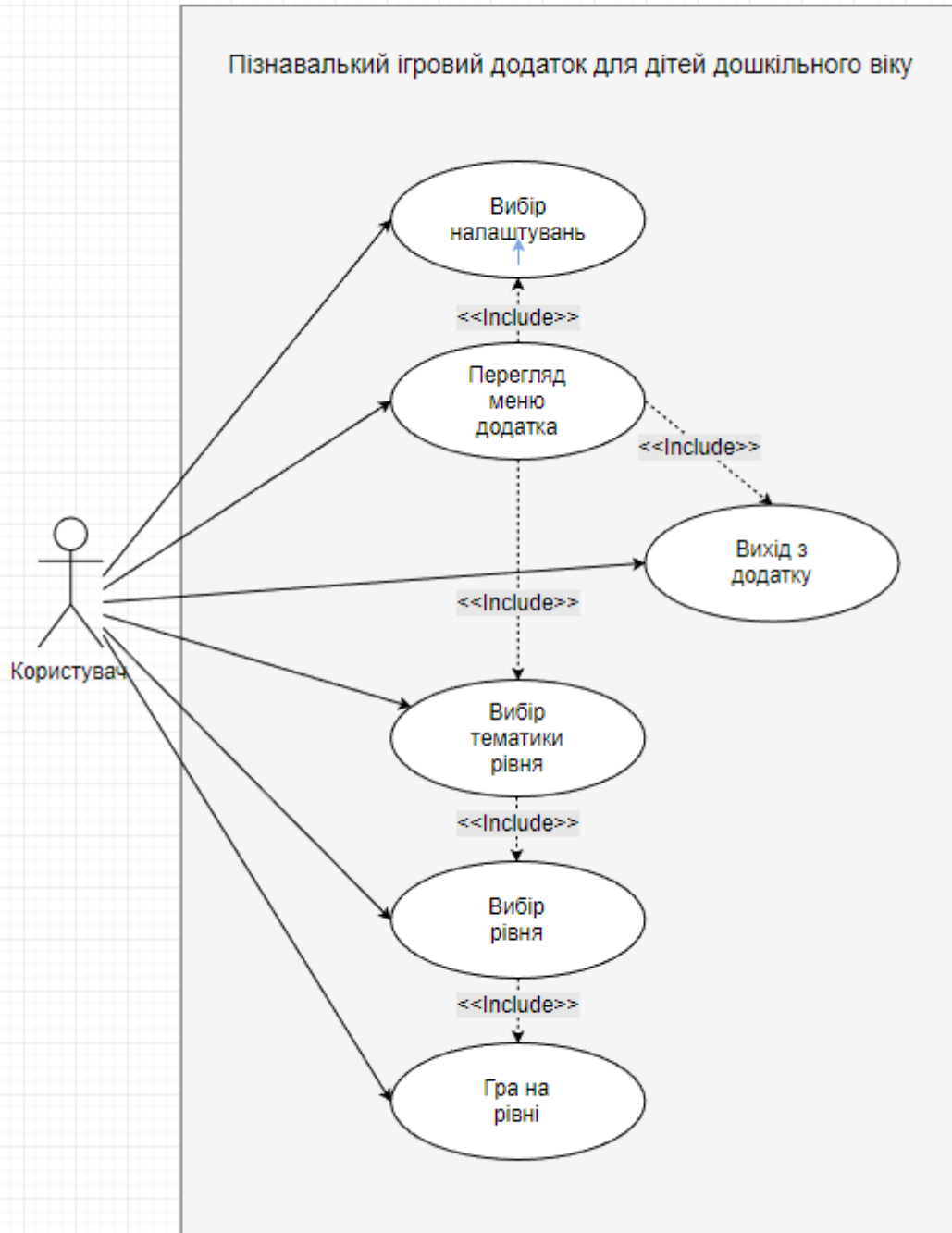


Рисунок 2.3 – Діаграма варіантів використання

На діаграмі варіантів використання присутній один актор – Користувач. Є такі варіанти використання як: перегляд меню додатка, вибір налаштувань, вихід з додатку, вибір тематики рівня, вибір рівня, гра на рівні. Перегляд меню додатка показує взаємодію користувача с програмою коли він буде знаходитися в головному меню. Вибір налаштувань показує взаємодію

користувача з додатком, з метою адаптування його під свої побажання. Вихід з гри – процес взаємодії, коли користувач виходить з ігрового додатку. Вибір тематики рівня відображає процес взаємодії користувача з додатком при виборі тематики рівня перед початком гри. Вибір рівня відображає процес взаємодії під час вибору рівня тієї тематики, яка була обрана раніше. Гра на рівні показує взаємодію користувача з додатком безпосередньо під час гри.

2.3 Діаграма діяльності

Діаграма діяльності може показати можливі сценарії діяльності користувачів в ігровому додатку. За допомогою цього можна побудувати зручну і інтуїтивно зрозумілу структуру додатку [14].

На рисунку 2.4 представлена діаграма діяльності ігрового додатку.

На даній діаграмі зображені такі варіанти подій, як вибір налаштування, вибір тематики рівня, вибір рівня та гра. Також показані події які можуть з'явитися після певного вибору. Показано такі події: налаштування відкриті, ігровий додаток відкритий, вибір тематики відкрито, вибір рівня відкрито, гра запущена.

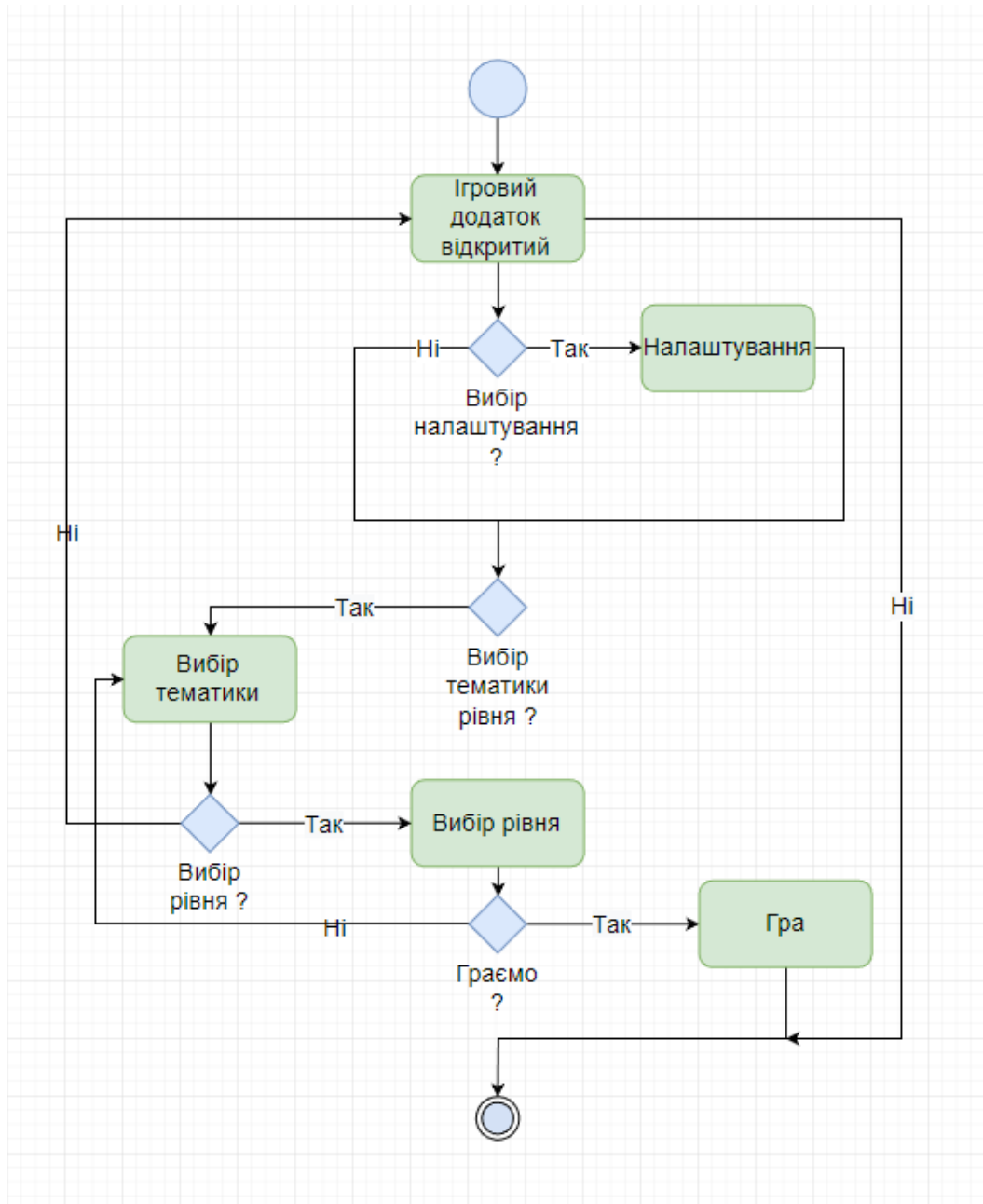


Рисунок 2.4 – Діаграма діяльності

3 РОЗРОБКА ІГРОВОГО ДОДАТКУ

3.1 Архітектура програмного додатку

Для побудови гри була обрана основна механіка – механіка пазлів. Виходячи з цього всі ігрові рівні використовують цю механіку різних типів. Всі завдання поділяються на два основних типи: завдання з літерами та словами та завдання з числами. Після обрання механіки гри та завдання, які будуть використовувати цю механіку переходимо до реалізації ігрового додатку [15]. Створюємо проект в Unity та в налаштуваннях File – Build Settings обираємо платформу Android та обираємо її (рис.3.1).

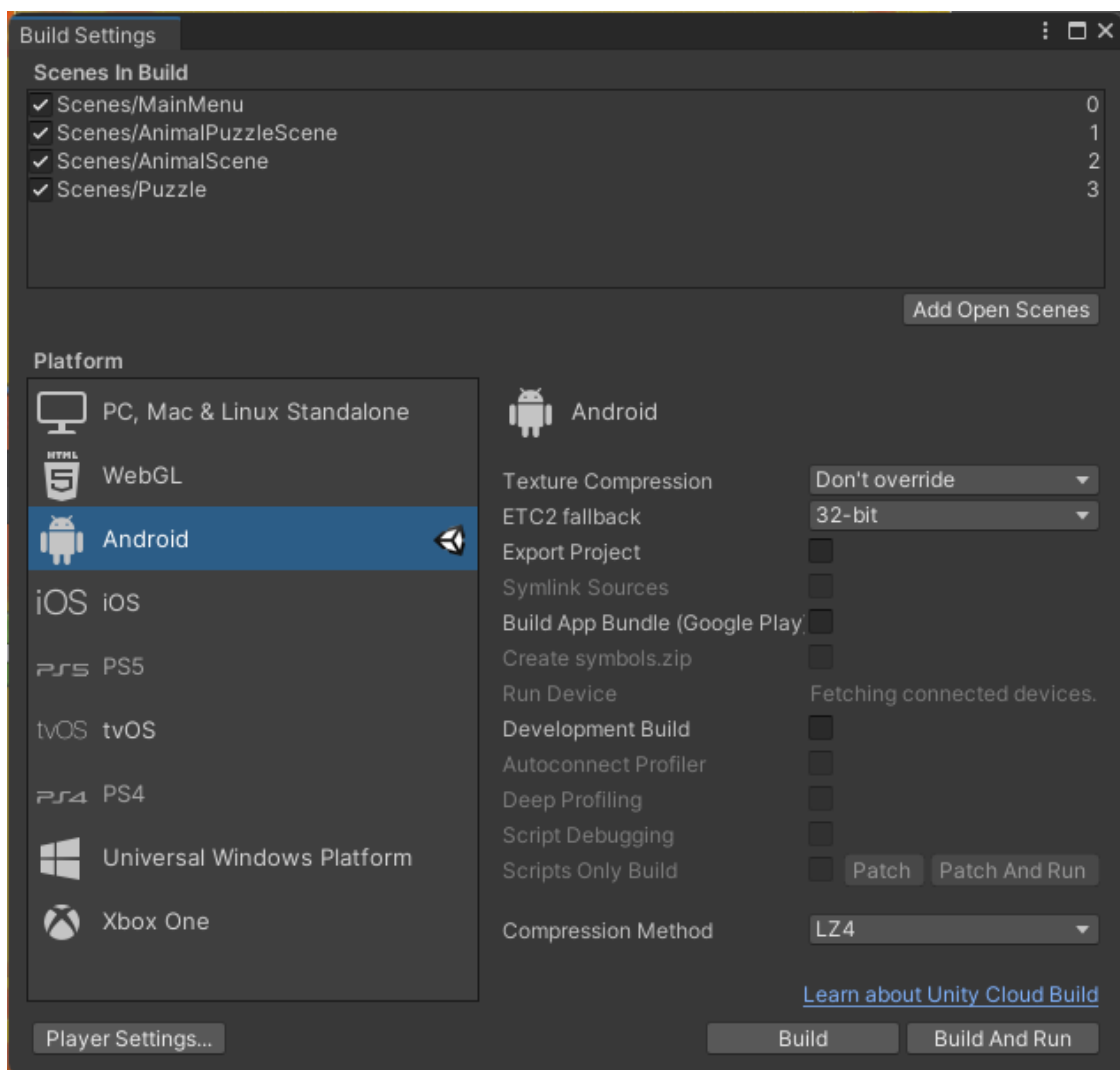


Рисунок 3.1 – Налаштування проекту в Unity

Основною сценою, що буде бачити гравець після запуску додатку – буде головне меню (рис.3.2). Меню має три кнопки – компонент `Button`, на них застосована проста анімація, і за допомогою них можна перейти до панелі вибору тематики рівнів, налаштувань чи вийти з додатку.

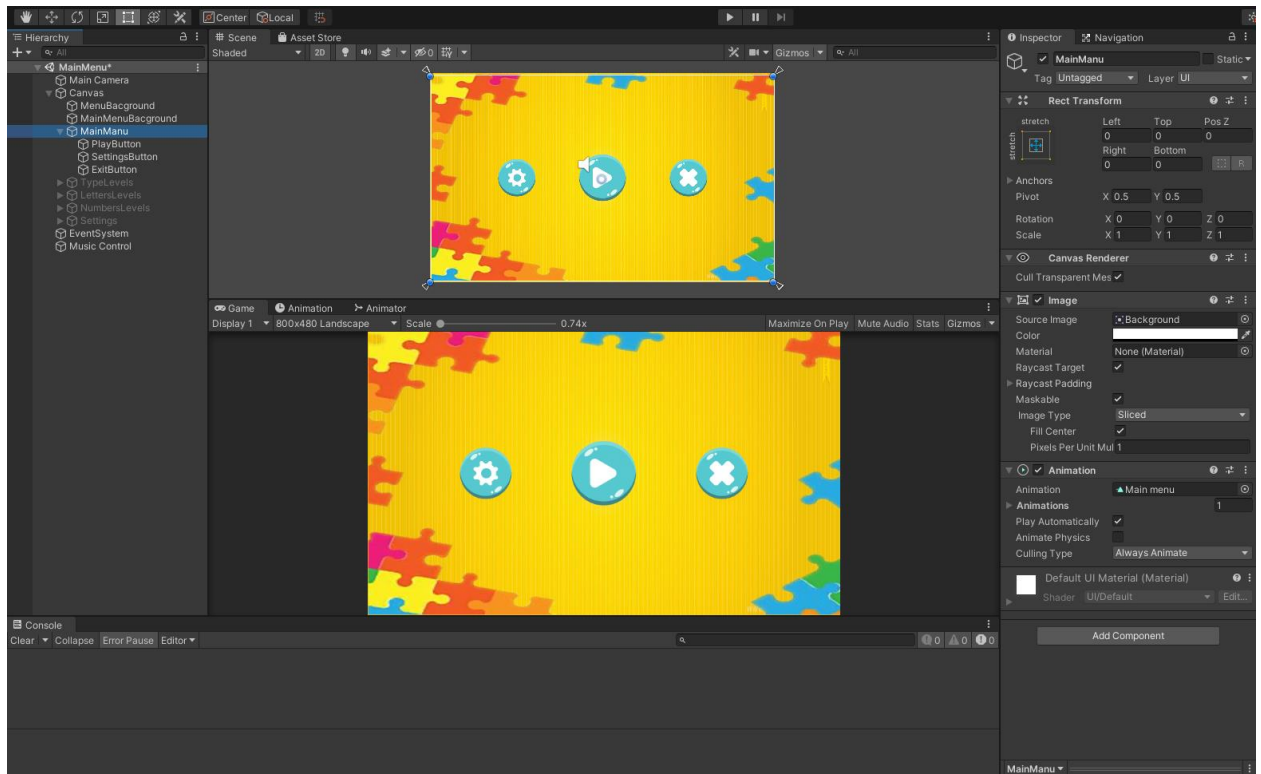


Рисунок 3.2 – Компоненти головного меню

Панель вибору тематики рівнів (рис.3.3) також має три кнопки, завдяки яким можна обрати тематику рівня чи повернутися до меню. Панелі використовують компонент `Panel`.

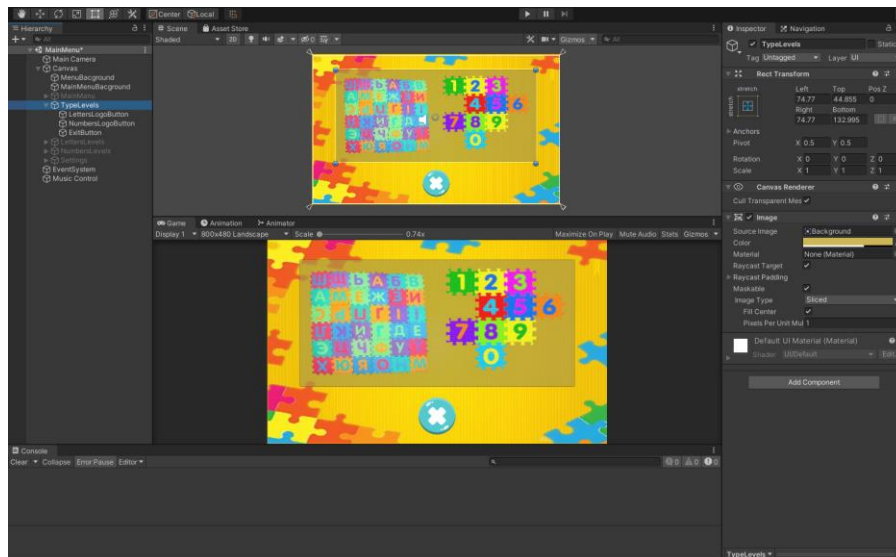


Рисунок 3.3 – Компоненти панелі вибору тематики рівнів

Рівні різної тематики мають ідентичні панелі (рис.3.4). З них можна перейти на потрібний рівень чи повернутися до панелі вибору тематики. Панелі на яких знаходяться кнопки переходу на різні рівні використовують компонент `Grid Layout Group`, який створює сітку в якій і розміщуються кнопки.

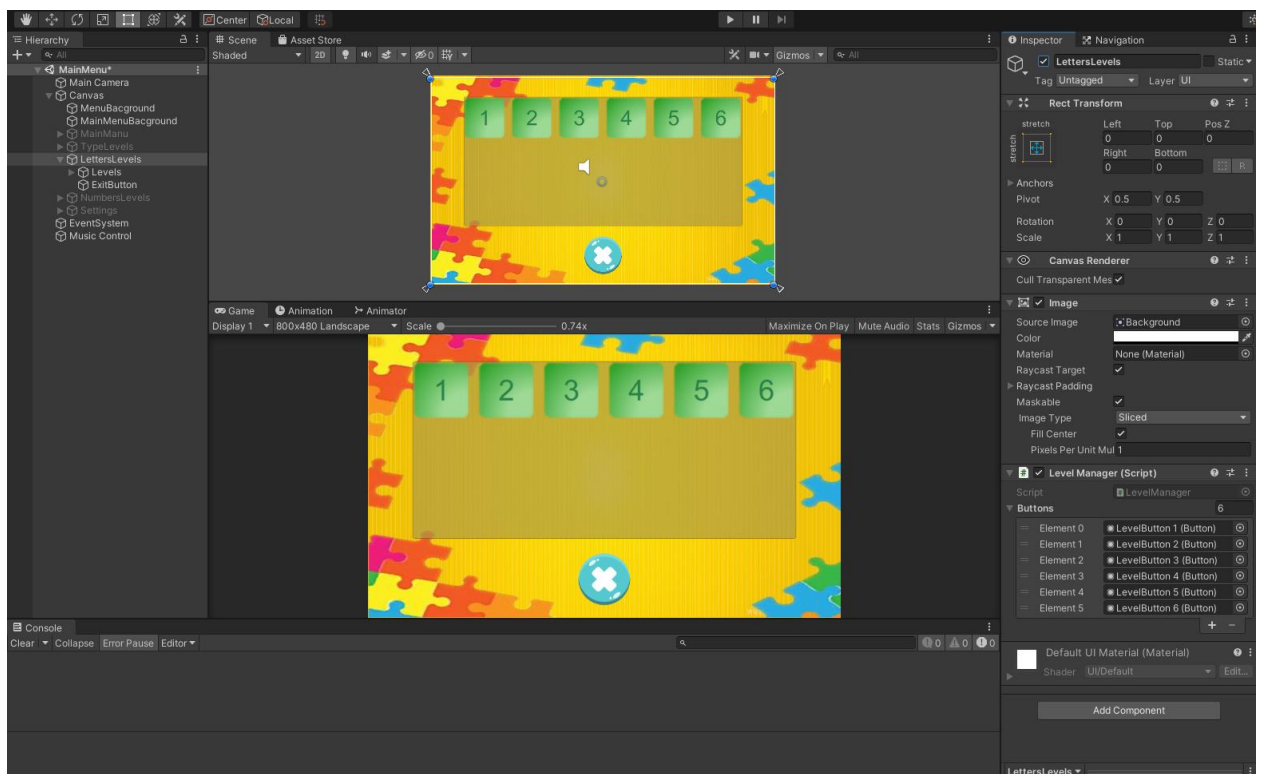


Рисунок 3.4 – Реалізація панелі менеджменту рівнів

Панель налаштувань використовується для налаштування гучності звуків в додатку (рис.3.5). Вона має кнопку повернення до головного меню, повзунок налаштування гучності (Slider) та кнопку вимкнення звуку. В усіх розділах головного меню є супровід з фонові музики, що реалізується за допомогою компонента Audio Source [16].

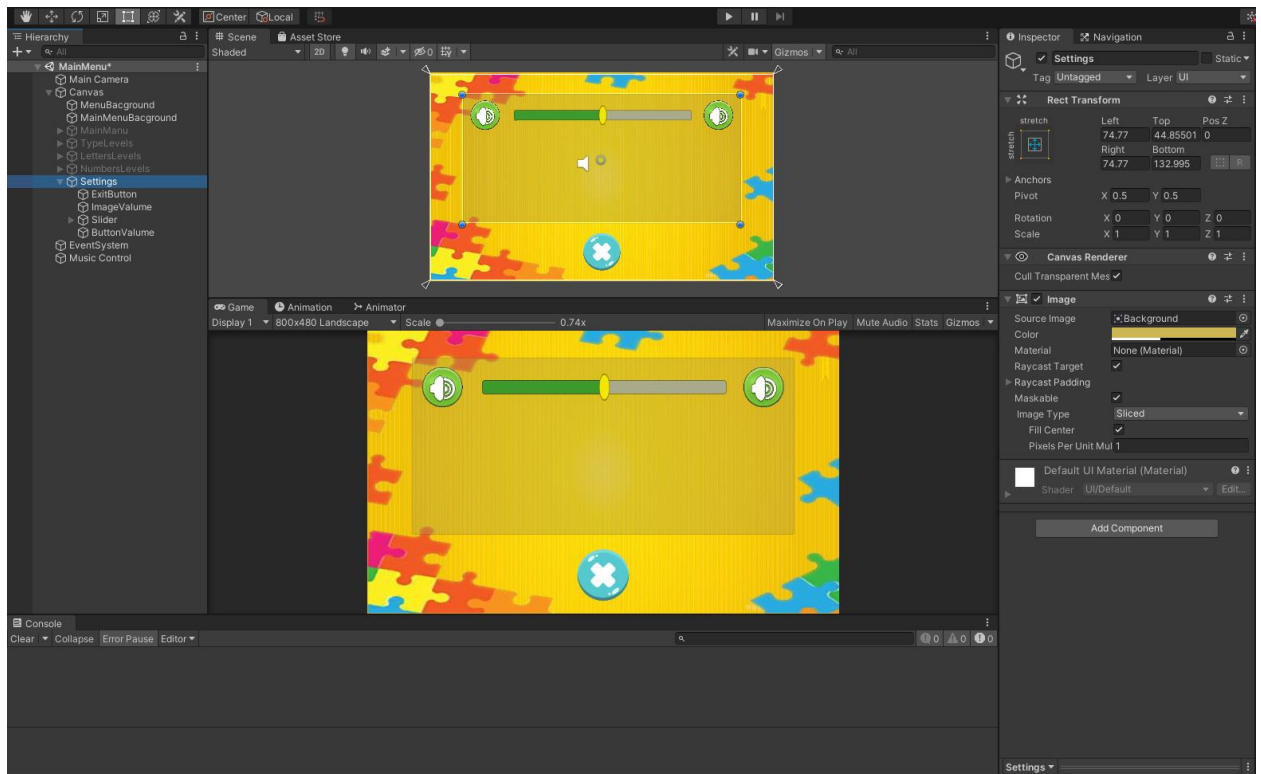


Рисунок 3.5 – Реалізація панелі налаштувань

Одним з типів ігрових рівнів – є рівень складання слів за картинкою (рис.3.6). Він складається з декількох панелей (компонент Panel), завдяки яким реалізується складання різних слів на одному рівні. На кожній панелі є компонент Image на якому розташоване зображення тварини, назву якої потрібно скласти зі слів. Пазли з літерами та слоти для встановлення літер також використовують компонент Image. Музика та звуки на рівні реалізовані за допомогою компонента Audio Source. Рівень має фонову музику, при виборі літери також гравець чує її вимовляння. Встановлення пазлів та інші взаємодії також супроводжуються звуковими ефектами. Також рівень має кнопку (Button) виходу до головного меню [17].

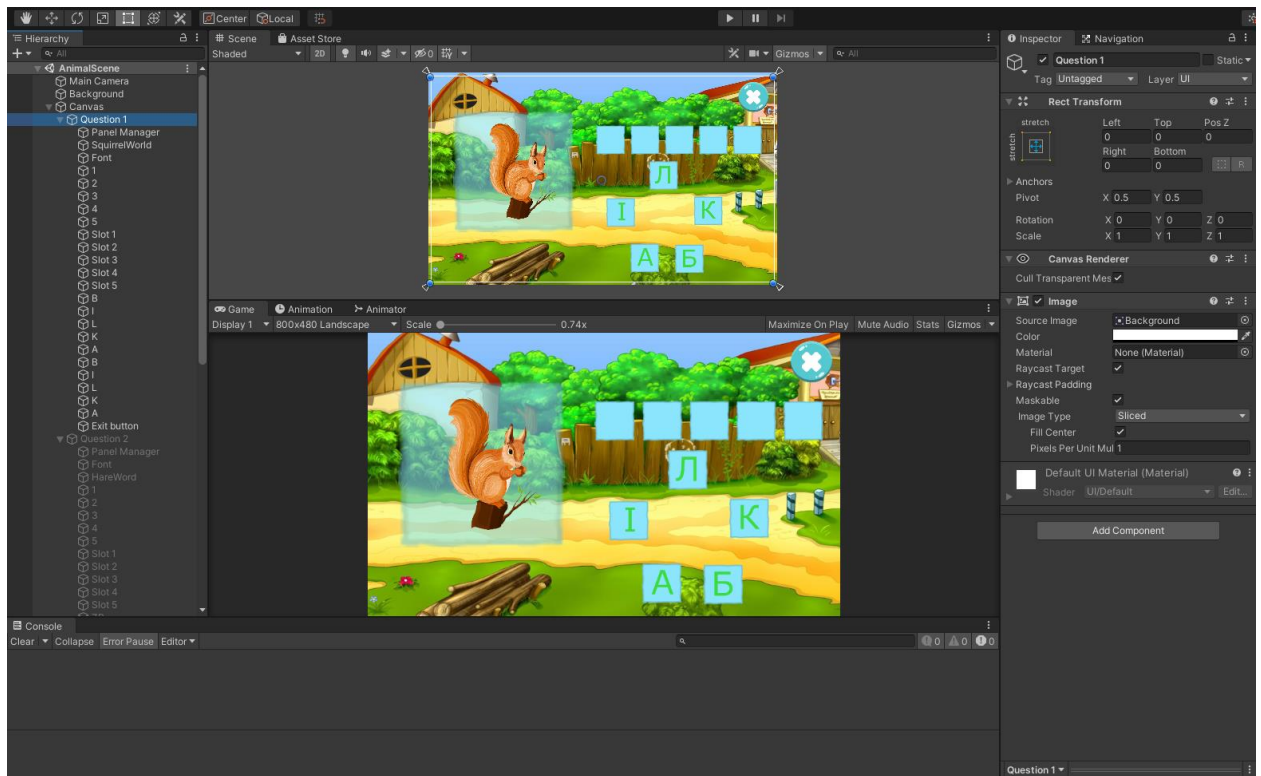


Рисунок 3.6 – Реалізація рівня зі складання слів

Ще один з типів рівнів полягає в розміщенні пазлів, які створюються у заданій кількості, але випадковим чином з великої кількості можливих варіантів. Даний рівень включає в себе сцену, яка побудована з різних спрайтів, об'єкт "Animal parent", що включає в себе місця спавну пазлів, об'єкт «Slot Parent», який містить в собі місця спавну слотів для пазлів та кнопку виходу з рівня в головне меню. Для можливості взаємодії з пазлами додаємо до них компонент Box Collider 2D. Приклад такого рівня показано на рисунку 3.7 та 3.8. Рівень також має звукове супроводження, що реалізується компонентами Audio Source [18].

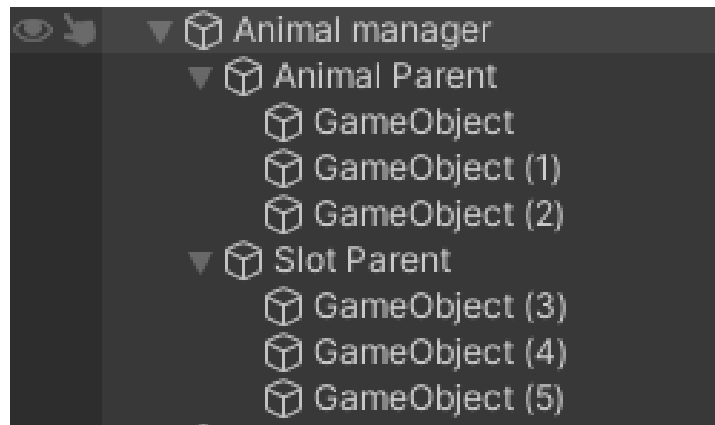


Рисунок 3.7 – Структура з якою взаємодіє скрипт спавну

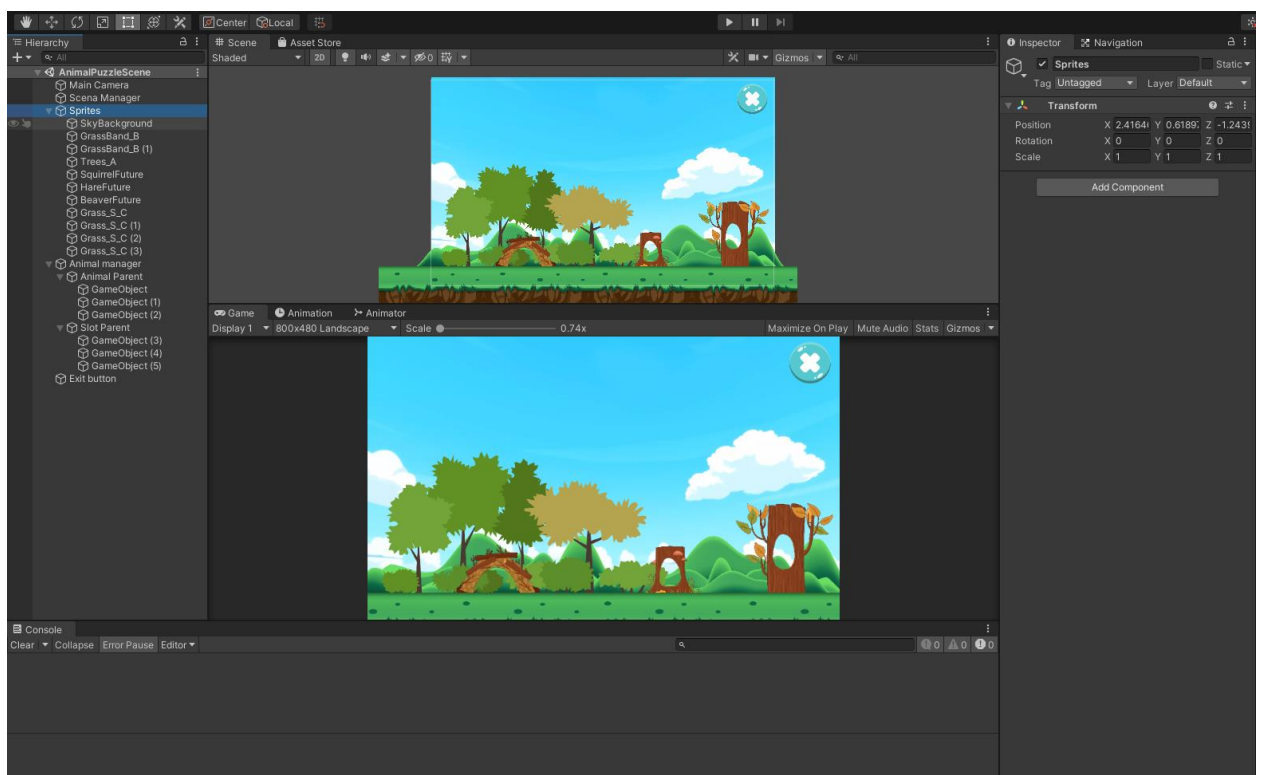


Рисунок 3.8 – Компоненти рівня з використання спавну

Ще один рівень, який в додатку використовується, як логічне продовження деяких попередніх рівнів – це складання картинки шляхом повороту її частин (рис.3.9). Для створення цього рівня на сцену завантажується картинка і потім за допомогою інструмента Sprite Editor картинка розрізається на потрібні частини. Далі ці частини з'єднуються і повертаються по осі, щоб зображення мало не правильну форму. Для того, щоб потім з цими частинами можна було взаємодіяти застосовуємо до них

компонент Box Collider 2D, також для створення звукового супроводження додаємо до компонентів Audio Source. Для контролю за поворотом пазлів створюємо пустий об'єкт (Create Empty) Game Control до якого застосовуємо відповідний скрипт [19].

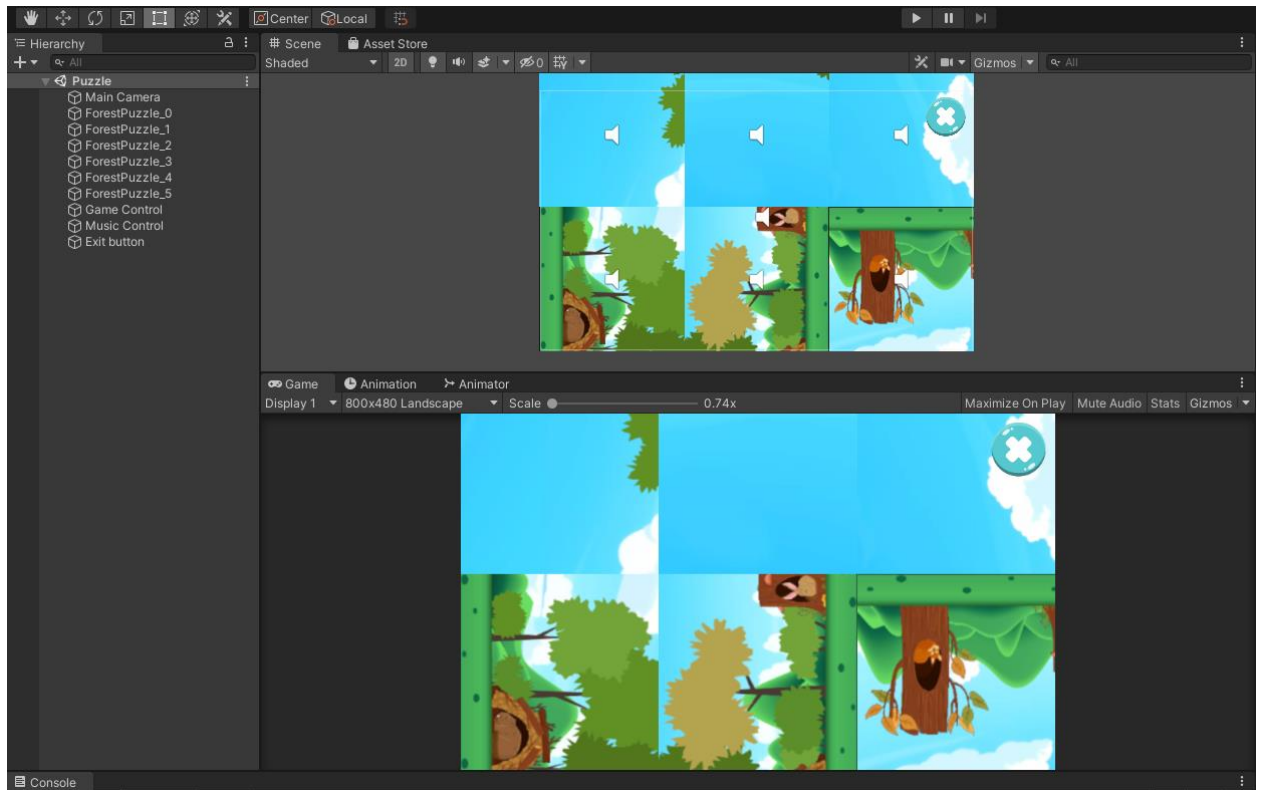


Рисунок 3.9 – Реалізація рівня з поворотом пазлів

3.2 Програмна реалізація

Скрипт MusicControl.

Даний скрипт відповідає за відтворення фонової музики на рівнях і при завантаженні нових рівнів та гарантує що ігровий об'єкт, що відповідає за фонову музику не буде дублюватися. Ще однією з функцій даного скрипту – є вимкнення звуку. Має поле musicSource для встановлення потрібної фонової музики (рис.3.10).

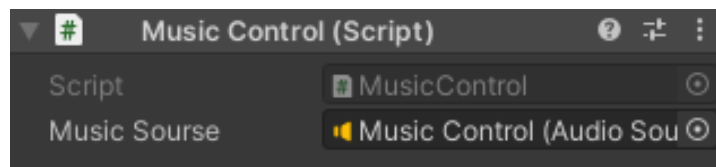


Рисунок 3.10 – Використання поля musicSource

Скрипт EndScene.

Цей скрипт перевіряє правильність розташування всіх пазлів на рівні і переходить на новий рівень, якщо всі пазли на правильних місцях. Застосовується на всіх рівнях де потрібно переміщувати пазли.

Скрипт Manager.

Скрипт відповідає за відображення сцен на одному рівні. Якщо завдання на одній сцені виконане, то скрипт ховає дану сцену та відображає нову. В поля скрипту обираються: панель яку потрібно сховати, панель яку потрібно показати та кількість слотів для пазл на панелі (рис.3.11).

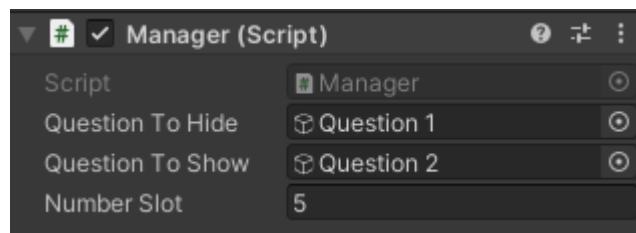


Рисунок 3.11 – Використання полів в скрипті Manager

Скрипт **PuzzlePiece**.

Цей скрипт відповідає за переміщення пазлів після їх спавну, звуковий супровід при діях над пазлами та перевірки відстаней до потрібних місць розташування пазлів. В полях вказується: префаб, який потрібно відображати, звук при натисканні на пазл та при відпусканні пазла (рис.3.12).

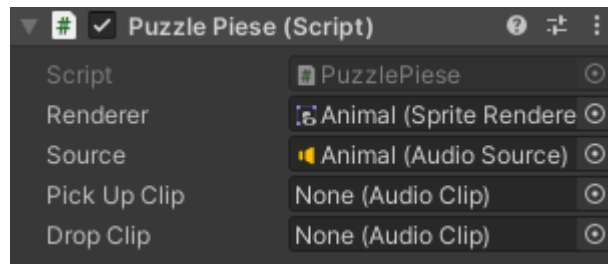


Рисунок 3.12 – Поля скрипту PuzzlePiece

Скрипт **PuzzleSlot**.

Даний скрипт відповідає за відображення префаба слота та звуковий супровід при встановленні пазла в слот. Відповідно в полях можна обрати спрайт, який потрібно показувати та звуковий ресурс (рис.3.13).

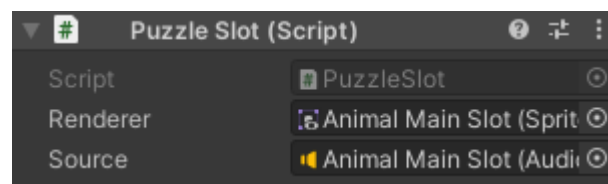


Рисунок 3.13 – Поля скрипту PuzzleSlot

Скрипт **PuzzleManager**.

Скрипт відповідає за створення потрібної кількості пазл та відповідних слотів для них на рівні. В полях даного скрипта вказується префаб пазла, групу точок спавна пазлів та групу точок спавна слотів (рис.3.14).

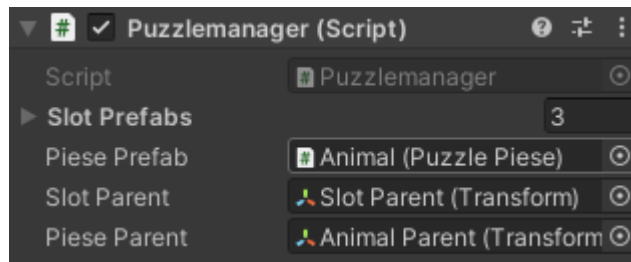


Рисунок 3.14 – Використання полів в скрипті PuzzleManager

Скрипт WordPieze.

Даний скрипт виконує аналогічні функції, що і скрипт PuzzlePieze, але для сцен в яких не потрібно використовувати спавну. В полях вказуються: полотно (Canvas) на якому розташовуються всі елементи та id пазла. Використовується на рівнях, де складаються слова (рис.3.15).

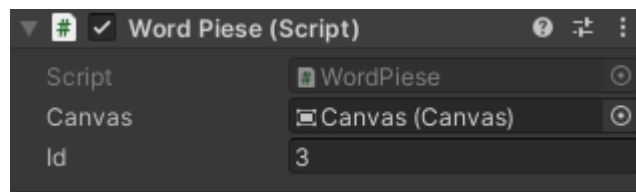


Рисунок 3.15 – Використання полів в скрипті WordPieze

Скрипт WordSlot.

Цей скрипт перевіряє id пазла та id слота, і при співпадінні їх id поміщає пазл в слот. Якщо id не співпадає, пазл в слот не вставляється. В полі вказується id слота. Також використовується на рівнях зі складанням слів (рис.3.16).

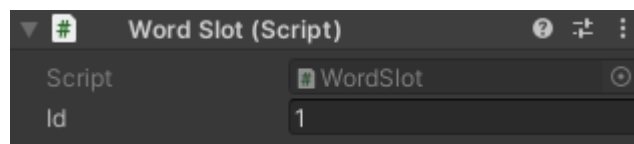


Рисунок 3.16 – Використання полів в скрипті WordSlot

Скрипт TouchRotate.

Скрипт відповідає за поворот пазла на 90 градусів при натисненні на нього, та відповідний музичний супровід. Має поля для компонента Audio

Source та звукового файлу на відтворення звуку при натиску на картинку. Використовується на рівнях, де потрібно правильно повернути частини пазлу (рис.3.17).

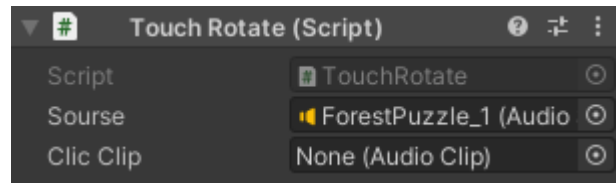


Рисунок 3.17 – Використання полів в скрипті TouchRotate

Скрипт TouchControl.

Даний скрипт перевіряє правильність повороту пазлів. Використовується на рівнях, де потрібно правильно повернути частини пазлу. Має поле де вказуються елементи пазлу (3.18).

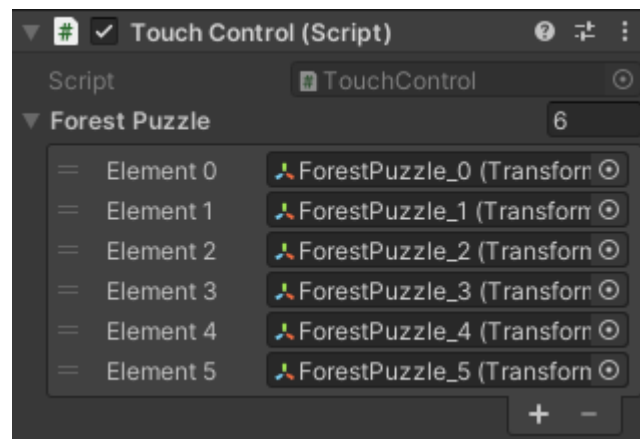


Рисунок 3.18 – Використання полів в скрипті TouchControl

Скрипт ToggleAudio.

Скрипт відповідає за налаштування гучності звуків у ігровому додатку та за різне відображення кнопку вимкнення звуку в залежності від її значення. Скрипт має поля, які приймають компонент кнопки та два спрайти, які повинні відобразитися на кнопці (рис.3.19).

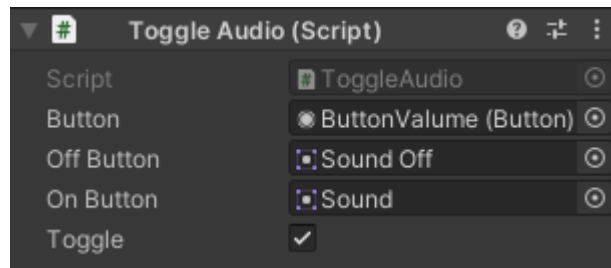


Рисунок 3.19 – Використання полів в скрипті ToggleAudio

Скрипт VolumSlider.

Цей скрипт відповідає за функціонування слайдеру, який регулює гучність звуків у грі. Даний скрипт має поле де вказується безпосередньо слайдер з яким він взаємодіє (рис.3.20).

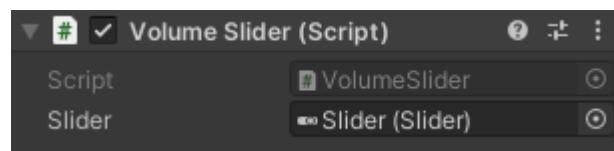


Рисунок 3.20 – Використання полів в скрипті volumSlider

3.3 Використання програмного додатку

Після розробки програмного продукту було протестовано його функціонування на всіх рівнях. Даний ігровий додаток використовує дуже поширенні ігрові механіки, які присутні в інших схожих додатках останніх років, але має свої особливості [20]. Ці особливості дозволяють логічно пов'язувати рівні та краще засвоювати матеріал дітям.

Першим протестованим рівнем було головне меню. При тестуванні перевірявся його дизайн, анімацію та функціонування кнопок. Також було протестовано панель вибору тематики рівнів та панелі вибору рівнів, які також знаходяться на сцені головного меню. Працездатність менеджменту рівнів показано на рисунках 3.21 та 3.22.



Рисунок 3.21 – Робота менеджера рівнів



Рисунок 3.22 – Робота менеджера рівнів, після проходження першого рівня

Після перевірки працездатності головного меню було перевірено перший тип ігрового рівня, а саме відображення спрайтів, спавн префабів, переміщення та встановлення елементів, звуковий супровід та перехід на наступний рівень (рис.3.23).



Рисунок 3.23 – Використання першого рівня в ігровому додатку

Наступним кроком була перевірка ще одного ігрового рівня. При перевірці було перевірено реакцію додатку на правильне та неправильне розміщення літер, звуковий супровід відповідно до дій користувача та кнопку повернення до меню (рис.3.24).



Рисунок 3.24 – Робота рівня зі складанням слів

Останньою перевіркою стала перевірка використання налаштувань в грі (рис.3.25). Для перевірки, після зміни налаштувань в меню, були відвідані різні рівні, щоб впевнитися в тому, що зміни застосовуються.

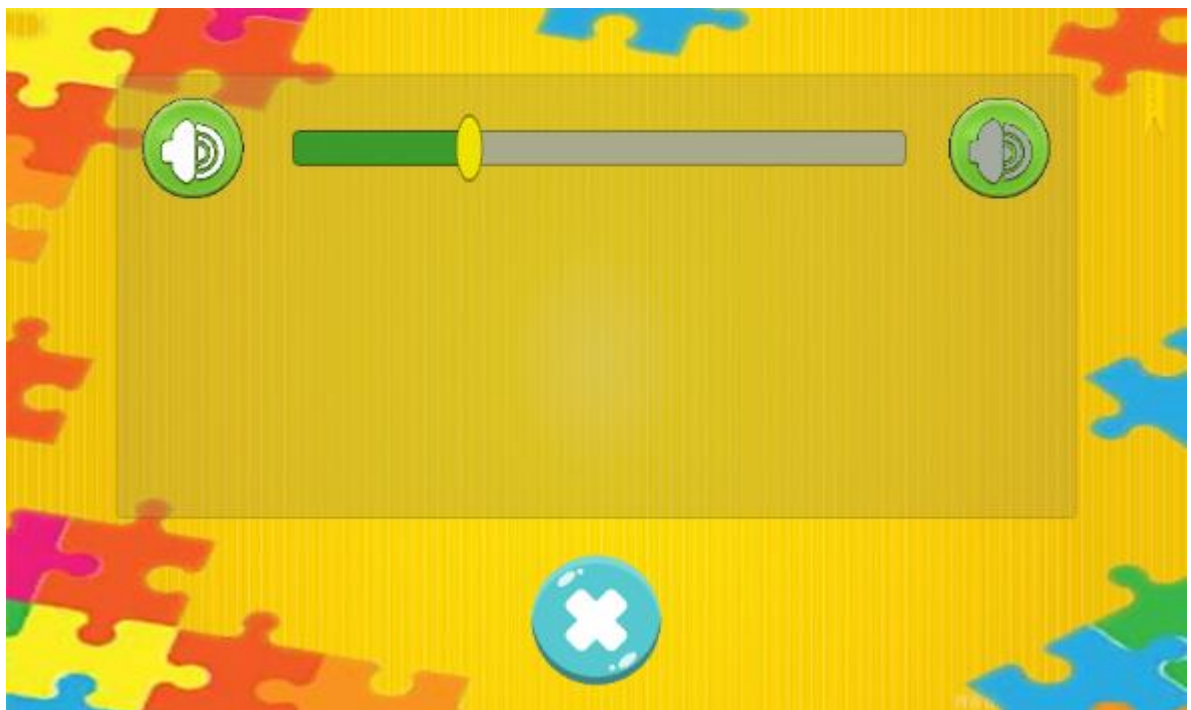


Рисунок 3.25 – Використання панелі налаштувань

ВИСНОВКИ

Під час дипломного проектування було проведено аналіз предметної та виявлено проблеми предметної області. Встановлено, що діти дошкільного віку дуже швидко ростуть та змінюються, тому додаток повинен мати рівні, які цікаві для дітей з різним діапазоном віку. Також було встановлено, що більшість популярних додатків є російськомовними, тому актуальним є розробка україномовного. Інша проблема полягала в тому, що дрібна моторика у дітей дошкільного віку не достатньо розвинена, тому додаток повинен бути зручним у керування.

Для вирішення цих проблем перш за все було проаналізовано програмні аналоги. Було виявлено переваги та недоліки різних ігрових механік та технологій, що покращують ігровий додаток та полегшують його розробку. Також було перевірено доцільність розробки додатку такого типу саме на мобільних пристроях. Дослідження показало, що технології і засоби розробки додатків для мобільних пристроях дозволяють вирішити виявлені проблеми. Обрані засоби розробки дозволяють мати у ігровому додатку велику кількість рівнів, що вирішує проблему з різницею у віці.

В результаті було отримано багаторівневий готовий продукт. Додаток розрахований на використанні дітьми у звичайному побуті, наприклад дома чи у подорожі. В подальшому він може удосконалюватися шляхом збільшення кількості та різноманітності рівнів, тим самим зможе охоплювати більшу кількість вікових діапазонів. Також додаток можна легко налаштувати, для використання на інших операційних системах, що використовуються у мобільних пристроях.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Лисенко Н. В. Педагогіка українського дошкілля: У 3-х частинах. : навч. посіб. / Н. В. Лисенко, Н. Р. Кирста. – К. : видавничий Дім "Слово", 2010. – Ч. 2. – 360 с.
2. Артемова Л В. Вчися граючись / Л В. Артемова. – Київ: Томіріс, 1995. –. 112 с.
3. Ладивір С. Пізнавальний розвиток: пошук ефективних шляхів / С. Ладивір // Дошкільне виховання. – 2002. - №10. – С. 4-6.
4. Castro R. Educational apps can benefit young children, study finds. *Florida International University*. URL: <https://news.fiu.edu/2019/educational-apps-can-benefit-young-children,-study-finds> (date of access: 25.05.2022).
5. Bohn K. Top educational apps for children might not be as beneficial as promised. *PennState*. URL: <https://www.psu.edu/news/research/story/top-educational-apps-children-might-not-be-beneficial-promised/> (date of access: 25.05.2022).
6. Tiny Puzzle для Андроїд. *APKPure.com*. URL: <https://apkpure.com/ru/tiny-puzzle-learning-games/com.blackbean.tiny-puzzle> (дата звернення: 02.06.2022).
7. Company T. P. Baby Shark Jigsaw Puzzle Fun. *Android Apps on Google Play*. URL: https://play.google.com/store/apps/details?id=kr.co.smartstudy.kidspuzzlefun_android_googlemarket&hl=uk&gl=US (date of access: 11.06.2022).
8. Kids Puzzle Games | Digipuzzle.net. *Educational games and puzzles | Digipuzzle.net*. URL: <https://www.digipuzzle.net/main/kids/> (date of access: 11.06.2022).

9. Панфилов А.П. Игровое моделирование в деятельности педагога / А.П. Панфилов «Учебное пособие для студентов высших учебных заведений издательский центр Академия»: 2007.
10. Брежнева О. Г. Жива математика або Технологія інтегрованих дидактичних модулів для логіко-математичного розвитку дітей // Методична скабничка вихователя. – 2015. - № 9-10. – С. 24-32, 58-65.
11. Програма розвитку дітей старшого дошкільного віку «Впевнений старт» / кер. проекту Б. М. Жебровський. – К., 2012. – 37 с
12. The Complete Guide To Understand IDEF Diagram | EdrawMax Online. *Edrawsoft*. URL: <https://www.edrawmax.com/article/the-complete-guide-to-understand-idef-diagram.html> (date of access: 11.06.2022).
13. Застосування UML в дипломних роботах. *Державний університет телекомунікацій*. URL: https://dut.edu.ua/ua/news-1-626-7758-zastosuvannya-uml-v-diplomnih-robotah_kafedra-kompyuternih-nauk-ta-informaciynih-tehnologiy (дата звернення: 11.06.2022).
14. Петелька. Uml діаграма діяльності. *StudFiles*. URL: <https://studfile.net/preview/5200239/page:7/> (дата звернення: 11.06.2022).
15. ph_piter. Книга «Unity в действии. Мультиплатформенная разработка на С#. Хабр. URL: <https://habr.com/ru/company/piter/blog/440292/> (дата звернення: 03.06.2022).
16. Основные ошибки при проектировании главного меню игры // Хабрахабр. URL: <https://habrahabr.ru/post/195608/>
17. Seif E. Kawem. How to create a Quiz app for mobile (Unity 3d), 2019. *YouTube*. URL: <https://www.youtube.com/watch?v=ageE5PC-trU> (date of access: 01.06.2022).
18. Tarodev. How to make a drag and drop game in Unity, 2021. *YouTube*. URL: https://www.youtube.com/watch?v=o_qEXZhQR-M (date of access: 03.06.2022).

19. Alexander Zotov. Simple Educational Puzzle Game For Kids For Android | Unity Tutorial, 2018. *YouTube*. URL: https://www.youtube.com/watch?v=TMQrO3Hy_LE (date of access: 11.06.2022).

20. Games software revenues to reach \$110 billion by 2018. [Electronic resource] / James Brightman.– Access mode: <http://www.gamesindustry.biz/articles/2015-05-04-games-software-revenues-to-reachusd110-billion-by-2018-digi-capital>. – 08.11.2015.

ДОДАТОК А

ТЕХНІЧНЕ ЗАВДАННЯ
на розробку інформаційної системи
«Пізнавальний ігровий додаток
для дітей дошкільного віку»

ПОГОДЖЕНО:

Доцент кафедри комп'ютерних наук

_____ Бойко О.В.

Студент групи ІТ-82-0

_____ Яценко С.В.

1. ПРИЗНАЧЕННЯ Й МЕТА СТВОРЕННЯ ІГРОВОГО ДОДАТКУ

1.1 Призначення ігрового додатку

Ігровий додаток повинен дозволити дітям дошкільного віку цікаво засвоювати базові знання, які пов'язані з літерами, словами та цифрами.

1.2 Мета створення ігрового додатку

Метою є теоретично обґрунтувати, розробити та апробувати пізнавальний ігровий додаток для дітей дошкільного віку засобами сучасних технологій навчання.

1.3 Цільова аудиторія

Цільова аудиторія даного ігрового додатку складається з дітей віком до 5 років.

2 ВИМОГИ ДО ІГРОВОГО ДОДАТКУ

2.1 Вимоги до ігрового додатку в цілому

2.1.1 Вимоги до структури й функціонування ігрового додатку

Ігровий додаток повинен складатися з окремих рівнів, які поступово стають доступні. Додаток повинен працювати на операційній системі Android.

2.1.2 Вимоги до збереження інформації

Вся інформація, яка зберігається в додатку зберігається безпосередньо на пристрої де цей додаток встановлено.

2.1.3 Вимоги до розмежування доступу

Додаток доступний для всіх користувачів. Всі користувачі мають однакові права і можуть отримувати доступ до всіх функцій.

2.2 Структура ігрового додатку

2.2.1 Загальна інформація про структуру ігрового додатку

Структура ігрового додатку – це набір різноманітних рівнів, які пов’язані через головне меню. В головному меню є можливість обрати тематику рівнів та конкретні рівні даної тематики.

2.2.2 Навігація

Так як основна навігація реалізована в головному меню, на кожному рівні є кнопка переходу до головного меню, щоб забезпечити швидкий доступ до навігації.

2.2.3 Наповнення ігрового додатку

Наповнення ігрового додатку додатковим контентом можливо лише розробником, безпосередньо через середовище Unity.

2.2.4 Дизайн та структура додатку

Так, як додаток призначений для дітей дошкільного віку, його дизайн виконаний у яскравих кольорах, таких, як: жовтий, салатовий та блакитний. Навігація по додатку зроблена максимально просто, щоб дітям було максимально просто зрозуміти додаток.

2.2.5 Короткий опис головного потоку виконання програми

При запуску додатку ми потрапляємо в головне меню. З головного меню ми можемо почати грати, зайти в налаштування чи вийти з гри. Коли ми починаємо грати, то потрапляємо в на панель вибору тематики рівня. Після вибору тематики користувач потрапляє на панель вибору рівнів. Вибравши рівень ми потрапляє на обраний рівень. Після проходження рівня користувач потрапляє на наступний рівень. На кожному рівні є можливість повернутися до головного меню.

2.3 Вимоги до функціонування системи

2.3.1 Потреби користувача

Потреби користувача наведені у таблиці А.1.

Таблиця А.1 – Потреби користувача

ІД	Потреби користувача	Джерело
UN-01	Можливість обирати тематику рівнів	Розробник
UN-02	Можливість налаштування додатку	Розробник
UN-03	Наявність звукового супроводу при взаємодії з додатком	Розробник

Продовження таблиці А.1.

UN-04	Поступовий доступ до рівнів	Розробник
UN-05	Можливість швидкого повернення до головного меню	Розробник

2.3.2 Функціональні вимоги

На основі потреб користувача визначаємо такі вимоги до функціональних характеристик:

- наявність панелі з обрання тематики рівнів;
- наявність панелі налаштувань;
- наявність панелі з вибору рівнів;
- реалізація поступового доступу до рівнів;
- наявність кнопки повернення до меню на всіх рівнях;

2.3.3 Вимоги до вхідних та вихідних даних

Вхідними даними ігрового додатку є налаштування звуку, які обрав користувач, вибір тематики рівня та вибір рівня.

Вихідні дані – це гра, згідно обраних вхідних даних.

2.3.4 Вимоги до програмного та апаратного забезпечення

Додаток працює на пристроях в операційною системою Android 5.1 і вище та з будь якими технічними характеристиками.

2.4 Вимоги до видів забезпечення

2.4.1 Вимоги до інформаційного забезпечення

Для створення ігрового додатку використовується:

- Unity 2022.1.4
- C#
- GIMP 2.10.22

2.4.2 Вимоги до лінгвістичного забезпечення

Ігровий додаток використовує українську мову.

2.4.3 Склад і зміст робіт зі створення web-додатку

Послідовність створення ігрового додатку наведена в таблиці А.2.

Таблиця А.2 – Етапи створення ігрового додатку

№	Склад і зміст робіт	Строк розробки (у робочих днях)
1	Пошук та створення спрайтів для додатка	3 дні
2	Створення рівнів	3 дні
3	Програмування рівнів	5 днів
4	Реалізація головного меню	2 дні
5	Тестування	2 дні

ДОДАТОК Б

Планування робіт

З кожним роком стрімко зростає розвиток інноваційних технологій, зокрема комп'ютерних ігор, без яких неможливо уявити процес виховання і розвитку дитини. Працюючі батьки через брак часу не можуть в повній мірі приділяти час для навчання своїх дітей, тому створююся навчальні ігрові додатки для дітей. Комп'ютерні ігри викликають у дітей величезний інтерес, тому за певний час гри дитина паралельно набуває деякі навички або підсвідомо вивчає матеріал. Створення навчальних ігрових додатків для дошкільного віку дасть змогу навчити дитину аналітично мислити в нестандартних ситуаціях, класифікувати і узагальнювати поняття, розвивати дрібну моторику рук і зорово-моторну координацію, набути впевненості в собі.

Деталізація мети проекту методом SMART. Для того, щоб готовий проект відповідав нашим початковим цілям потрібно на етапі планування правильно визначити мету за допомогою SMART-методу. Результати деталізації SMART-методом наведені в таблиці Б.1.

Таблиця Б.1. – Деталізація мети методом SMART

Specific (конкретна)	Всебічний розвиток дітей дошкільного віку
Measurable (вимірювана)	Розробити ігровий додаток з мінімальним використанням ресурсів
Achievable (досяжна)	Мета досяжна, є узгоджена тема проекту та навички у роботі з середовищами розробки ігор
Relevant (реалістична)	Додаток дозволить дітям проводити час за гаджетами з користю в україномовному середовищі
Time-framed (обмежена в часі)	Проект повинен бути виконаний до 10.06.2022

Планування змісту робіт. WBS (Work Breakdown Structure – Ієрархічна структура робіт) – структура розбиття на роботу. Дана структура призначена для розділення роботи на частини, за якими закріплені люди. Також ця структура дозволяє ретельно стежити за просуванням роботи: стежити за фінансами, проводити оцінку виконаної роботи та контролювати графіки роботи.

На найвищому рівні знаходиться безпосередньо продукт проекту. На нижчому рівні зафіксовані загальні дії, які спрямовані на досягнення мети проекту. Таким чином декомпозиція виконується до тих пір, доки дії не стануть простими і можуть бути виконаними однією особою. На рисунку Б.1. показано WBS – структуру з розробки ігрового додатку для дітей дошкільного віку.

Планування структури виконавців. OBS-структура проекту – це структура виконавців проекту. Вона представляє графічне відображення учасників проекту задіяних в його розробці. Співробітники відповідають за організацію та виконання елементарних робіт, які були зазначені у WBS-структурі.

На рисунку Б.2. показано OBS-структуру планування проекту. В таблиці Б.2. зазначено список виконавців причетних до розробки програмного продукту.

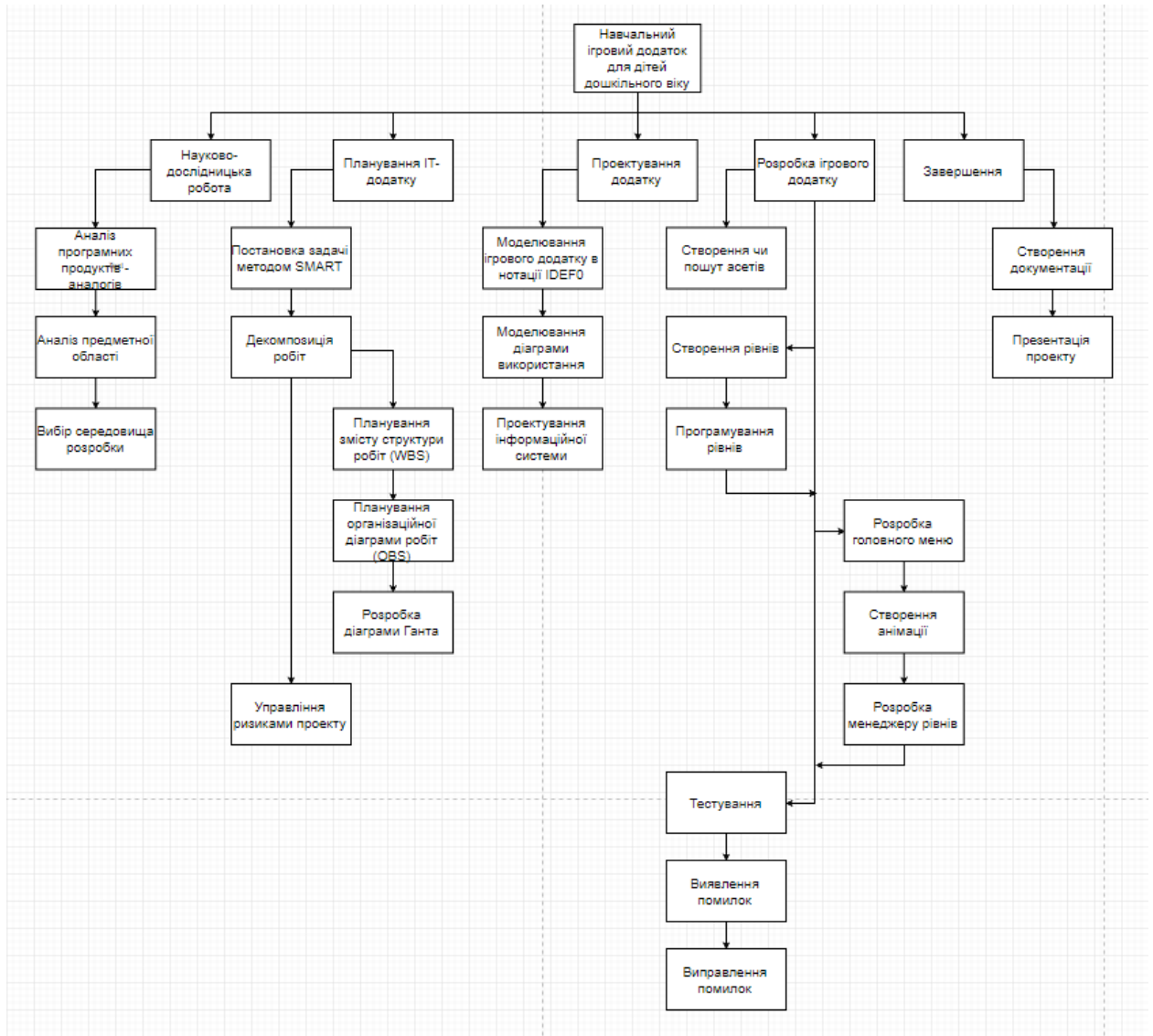


Рисунок Б.1. – WBS-структура проекту

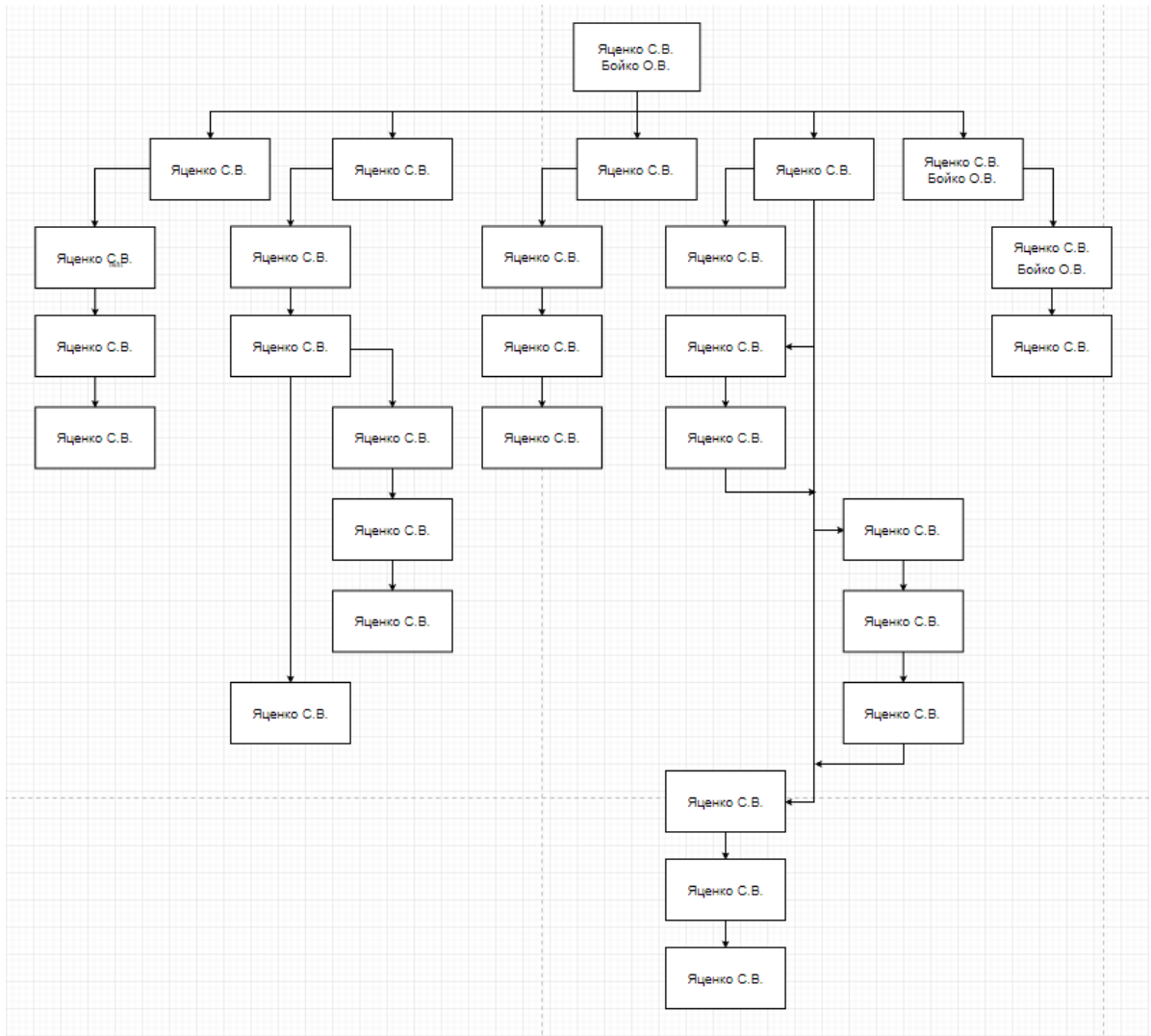


Рисунок Б.2. – OBS структура проекту

Таблиця Б.2.- Виконавці проекту

Роль	Ім'я	Проектна роль
Розробник	Яценко С.В.	Виконує розробку ігрового додатку
Проектувальник	Яценко С.В.	Виконує проектування структури ігрового додатку
Тестувальник	Яценко С.В.	Відповідає за тестування ігрового додатку
Керівник проекту	Бойко О.В.	Формує завдання проекту
Менеджер проекту	Яценко С.В.	Виконує аналіз даних та відповідає за розподіл ресурсів, стежить за виконанням термінів

Діаграма Ганта. Для того, щоб мати уявлення про терміни виконання проекту розробляється календарний план. Він виглядає як розклад з реальними датами і дозволяє отримати уявлення про тривалість процесів з урахуванням вихідних та святкових днів. Календарний графік проекту представлений на рисунку Б.3.

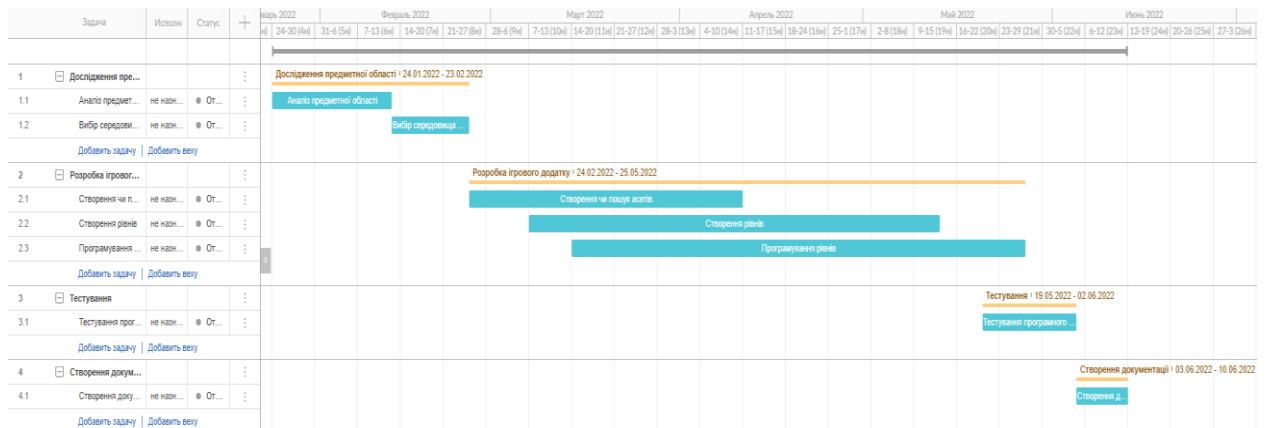


Рисунок Б.3 – Календарний графік проекту

Управління ризиками проекту. У будь якому проекті є ризики. Але, якщо спланувати можливі ризики, які можуть виникнути під час виконання проекту, то більшість з них можна буде уникнути чи зменшити їх негативний вплив до мінімуму. Є якісна та кількісна оцінка ризиків. Під час якісної оцінки визначається ступінь важливості ризику та реагування на нього. Кількісна оцінка ризиків оцінює варіанти виникання різних ризиків. У таблиці Б.3. представлено класифікацію ризиків за ймовірністю виникнення та їх впливом.

Таблиця Б.3 – Шкала оцінювання ризиків за впливом та ймовірністю виникнення.

Оцінка	Ймовірність	Вплив	Тип
1	Низька	Низький	Прийнятні
2	Середня	Середній	Виправдані
3	Висока	Високий	Недопустимі

Для мінімізації негативного впливу ризиків на проект потрібно виробити стратегію реагування на них. Оцінювання відбувається за даними з таблиці Б.3. В результаті було отримано матрицю впливу ризиків, що показано в таблиці Б.4.

Таблиця Б.4 – матриця ймовірності та впливу

Ймовірність			
Висока	-	RS7	RS1, RS6
Середня	-	RS9	RS2
Низька	RS10	RS3, RS4, RS5	RS8
	Низький	Середній	Високий
	Вплив		

Далі ми сплануємо всі можливі ризики та занесемо їх в таблицю Б.5.

Таблиця Б.5 – Ризики та стратегії реагування

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Тип ризику	План А	Тип стратегії реагування	План Б
RS1	Відкритий	Неоптимальний розподіл часу	Висока	Високий	Недопустимий	Звернути увагу на розподіл часу та дотримуватися календарного плану	Пом'якшення	Змінити пріоритети робіт за змінити об'єм наповнення продукту
RS2	Відкритий	Недостатня кваліфікація розробника	Середня	Високий	Недопустимий	Підвищити рівень знань в потрібній сфері	Пом'якшення	Переглянути процес розробки схожих програмних продуктів

Продовження таблиці Б.5.

RS3	Відкритий	Непорозуміння між виконавцем і керівником	Низька	Середній	Допустимий	Налагодити спілкування з керівником	Попередження	При непорозумінні в'яснити причину та виправити дану ситуацію
RS4	Відкритий	Не правильна оцінка масштабу	Низька	Середній	Допустимий	Провести правильний аналіз проекту та дослідити час виконання схожих проектів	Пом'якшення	Зміна масштабу проекту на менший

Продовження таблиці Б.5.

RS5	Відкритий	Вибір незадовільних технологій реалізації	Низька	Середній	Допустимий	Проаналізувати, які технології використовують аналогічні програмні продукти	Пом'якшення	Пошук варіантів, які допоможуть покращити вибрані технології
RS6	Відкритий	Помилки в програмному продукті	Висока	Високий	Недопустимий	Виконувати тестування після кожної глобальної зміни, зберігати проміжні версії	Пом'якшення	Відновитися до попередньої версії

Продовження таблиці Б.5.

RS7	Відкритий	Неточності в проектуванні	Висока	Середній	Недопустимий	Постійно комунікувати з керівником проекту	Пом'якшення	Здійснювати проміжний контроль в ході виконання проекту
RS8	Відкритий	Вихід з ладу робочого комп'ютера	Низька	Високий	Виправданий	Мати доступ до іншого робочого комп'ютера	Пом'якшення	Відремонтувати робочий комп'ютер та змінити графік виконання проекту

Таблиця Б.5 - Продовження

RS9	Відкритий	Проблеми з підключенням до мережі інтернет	Середня	Середній	Виправлений	Мати можливість підключитися до іншої мережі	Пом'якшення	Виконувати роботу, яка не потребує підключення до мережі доки не з'явиться підключення
RS10	Відкритий	Збої в роботі програмного забезпечення	Низька	Низький	Прийнятний	Перевстановити програмне забезпечення	Пом'якшення	Перевірити систему та драйвери на наявність оновлень та оновити їх

Класифікація ризиків за рівнем впливу на виконання проекту наведено в таблиці Б.6.

Таблиця Б.6 – Шкала оцінювання ризиків за впливом

№	Назва	Номера ризиків
1	Прийнятні	10
2	Виправдані	2, 3, 4, 5, 7, 8, 9
3	недопустимі	1, 6

ДОДАТОК Г – ЛІСТИНГ ПРОГРАМНОГО КОДУ

Код скрипту MusicControl.cs

```

using UnityEngine;

public class MusicControl : MonoBehaviour
{
    public AudioSource musicSource;

    public static MusicControl Instance;

    private void Awake()
    {
        if (Instance == null)
        {
            Instance = this;
            DontDestroyOnLoad(gameObject);
        }
        else
        {
            Destroy(gameObject);
        }
    }

    public void ChangeMasterValume(float value)
    {
        AudioListener.volume = value;
    }

    public void ToggleMusic()
    {
        musicSource.mute = !musicSource.mute;
    }
}

```

Код скрипту EndScene.cs

```

using UnityEngine;
using UnityEngine.SceneManagement;

public class EndScene : MonoBehaviour
{
    public static int numberOfPuzzle;
    public static int puzzleInPlace;

    void Update()
    {
        int nextLevel = SceneManager.GetActiveScene().buildIndex + 1;
        if (puzzleInPlace == numberOfPuzzle)
        {
            if (PlayerPrefs.GetInt("levelsUnlocked", 1) < nextLevel) ;
        }
    }
}

```

```

    {
        PlayerPrefs.SetInt("levelsUnlocked", nextLevel);
        SceneManager.LoadScene(nextLevel);
    }
}
}
}

```

Код скрипту **Manager.cs**

```

using UnityEngine;

public class Manager : MonoBehaviour
{
    public GameObject QuestionToHide, QuestionToShow;
    public int NumberSlot;

    public void Update()
    {
        if(WordSlot.inPlace == NumberSlot)
        {
            WordSlot.inPlace = 0;
            LoadNextPanel();
        }
    }

    public void LoadNextPanel()
    {
        QuestionToHide.SetActive(false);
        QuestionToShow.SetActive(true);
    }
}

```

Код скрипту **PuzzlePiece.cs**

```

using UnityEngine;

public class PuzzlePiese : MonoBehaviour
{
    [SerializeField] private SpriteRenderer _renderer;
    [SerializeField] private AudioSource _source;
    [SerializeField] private AudioClip _pickUpClip, _dropClip;

    private bool _dragging, _placed;
    private Vector2 _offset, _originalPosition;
    private PuzzleSlot _slot;

    public void Init(PuzzleSlot slot)
    {
        _renderer.sprite = slot.Renderer.sprite;
        _slot = slot;
    }
}

```



```

private void Awake()
{
    _originalPosition = transform.position;
}

void Update()
{
    if (_placed) return;
    if (!_dragging) return;
    var mousePosition = GetMousePos();
    transform.position = mousePosition - _offset;
}

void OnMouseDown()
{
    _dragging = true;
    _source.PlayOneShot(_pickUpClip);
    _offset = GetMousePos() - (Vector2)transform.position;
}

void OnMouseUp()
{
    if (Vector2.Distance(transform.position, _slot.transform.position) < 3)
    {
        transform.position = _slot.transform.position;
        _slot.Placed();
        _placed = true;
        EndScene.puzzleInPlace++;
    }
    else
    {
        transform.position = _originalPosition;
        _source.PlayOneShot(_dropClip);
        _dragging = false;
    }
}

Vector2 GetMousePos()
{
    return Camera.main.ScreenToWorldPoint(Input.mousePosition);
}
}

```

Код скрипту PuzzleSlots.cs

```

using UnityEngine;

public class PuzzleSlot : MonoBehaviour
{
    public SpriteRenderer Renderer;
}

```

```
[SerializeField] AudioSource _source;

public void Placed()
{
    _source.Play();
}
}
```

Код скрипту PuzzleManager.cs

```
using System.Collections.Generic;
using System.Linq;
using UnityEngine;

public class Puzzlemanager : MonoBehaviour
{
    [SerializeField] private List<PuzzleSlot> _slotPrefabs;
    [SerializeField] private PuzzlePiese _piesePrefab;
    [SerializeField] private Transform _slotParent, _pieseParent;

    void Start()
    {
        Spavn();
    }

    void Spavn()
    {
        var Set = _slotPrefabs.Take(3).ToList();
        for (int i = 0; i < Set.Count; i++)
        {
            var spavnSlot = Instantiate(Set[i], _slotParent.GetChild(i).position, Quaternion.identity);
            var spavnPiese = Instantiate(_piesePrefab, _pieseParent.GetChild(i).position,
Quaternion.identity);
            spavnPiese.Init(spavnSlot);
            EndScene.numberOfPuzzle++;
            Debug.Log(EndScene.numberOfPuzzle);
        }
    }
}
```

Код скрипту WordPiese.cs

```
using UnityEngine;
using UnityEngine.EventSystems;

public class WordPiese : MonoBehaviour, IPointerDownHandler, IBeginDragHandler,
IEndDragHandler, IDragHandler
{
    private RectTransform rectTransform;
    public Canvas canvas;
    private CanvasGroup canvasGroup;
```

```

public int id;
private Vector2 initPos;

void Start()
{
    rectTransform = GetComponent<RectTransform>();
    canvasGroup = GetComponent<CanvasGroup>();
    initPos = transform.position;
}

public void OnBeginDrag(PointerEventData eventData)
{
    canvasGroup.blocksRaycasts = false;
}

public void OnDrag(PointerEventData eventData)
{
    rectTransform.anchoredPosition += eventData.delta / canvas.scaleFactor;
}

public void OnEndDrag(PointerEventData eventData)
{
    canvasGroup.blocksRaycasts = true;
}

public void OnPointerDown(PointerEventData eventData)
{
}

public void ResetPosition()
{
    transform.position = initPos;
}
}

```

Код скрипта WordSlot.cs

```

using UnityEngine;
using UnityEngine.EventSystems;

public class WordSlot : MonoBehaviour, IDropHandler
{
    public int id;
    public static int inPlace;

    public void OnDrop(PointerEventData eventData)
    {
        if (eventData.pointerDrag != null)
        {
            if (eventData.pointerDrag.GetComponent<WordPiese>().id == id)

```

```

    {
        eventData.pointerDrag.GetComponent<RectTransform>().anchoredPosition =
this.GetComponent<RectTransform>().anchoredPosition;
        eventData.pointerDrag = null;
        inPlace++;
    }
    else
    {
        eventData.pointerDrag.GetComponent<WordPiese>().ResetPosition();
    }
}
}
}
}

```

Код скрипту TouchRotate.cs

```

using UnityEngine;

public class TouchRotate : MonoBehaviour
{
    [SerializeField] private AudioSource _source;
    [SerializeField] private AudioClip _clieClip;

    private void OnMouseDown()
    {
        transform.Rotate(0f, 0f, 90f);
        _source.PlayOneShot(_clieClip);
    }
}

```

Код скрипту TouchControl.cs

```

using UnityEngine;

public class TouchControl : MonoBehaviour
{
    [SerializeField] private Transform[] ForestPuzzle;

    private void Update()
    {
        if (ForestPuzzle[0].rotation.z == 0 &&
            ForestPuzzle[1].rotation.z == 0 &&
            ForestPuzzle[2].rotation.z == 0 &&
            ForestPuzzle[3].rotation.z == 0 &&
            ForestPuzzle[4].rotation.z == 0 &&
            ForestPuzzle[5].rotation.z == 0)
        {
            Debug.Log("You win");
        }
    }
}

```

Код скрипту ToggleAudio.cs

```

using UnityEngine;
using UnityEngine.UI;

public class ToggleAudio : MonoBehaviour
{
    public Button button;
    public Sprite offButton;
    public Sprite onButton;

    private bool isOn = true;

    public bool toggle;

    public void Toggle()
    {
        if (toggle)
        {
            MusicControl.Instance.ToggleMusic();
        }
    }
    public void clickButton()
    {
        if(isOn)
        {
            button.image.sprite = offButton;
            isOn = false;
        }
        else
        {
            button.image.sprite = onButton;
            isOn = true;
        }
    }
}

```

Код скрипту VolumSlider.cs

```

using UnityEngine;
using UnityEngine.UI;

public class VolumeSlider : MonoBehaviour
{
    public Slider slider;

    private void Start()
    {
        MusicControl.Instance.ChangeMasterVolume(slider.value);
        slider.onValueChanged.AddListener(val =>
        MusicControl.Instance.ChangeMasterVolume(val));
    }
}

```