

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Чорноморський національний університет імені Петра Могили**  
**Факультет комп'ютерних наук**  
**Кафедра інженерії програмного забезпечення**

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри

\_\_\_\_\_ *Є. О. Давиденко*

«\_\_» \_\_\_\_\_ 2023 р.

**КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА**  
**Мобільна гра казуального жанру з використанням**  
**3D-моделювання**

Спеціальність «Інженерія програмного забезпечення»

121 – КРБ – 409.21910914

**Студентка:**

\_\_\_\_\_ *В. М. Мальцева*

«\_\_» \_\_\_\_\_ 2023 р.

**Керівник:** *кандидат технічних наук, доцент*

\_\_\_\_\_ *А. П. Бойко*

«\_\_» \_\_\_\_\_ 2023 р.

**Консультант:** *кандидат технічних наук, доцент*

\_\_\_\_\_ *А. О. Алексеєва*

«\_\_» \_\_\_\_\_ 2023 р.

**Миколаїв – 2023**

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Чорноморський національний університет імені Петра Могили**  
**Факультет комп'ютерних наук**  
**Кафедра інженерії програмного забезпечення**

ЗАТВЕРДЖУЮ

Зав. кафедри Є. О. Давиденко

«\_\_\_\_\_» \_\_\_\_\_ 2023р.

**ЗАВДАННЯ**

**на виконання кваліфікаційної роботи бакалавра**

Видано студентці групи 409 факультету комп'ютерних наук

Мальцевій Владиславі Михайлівні

---

*(прізвище, ім'я, по батькові студента)*

1. Тема кваліфікаційної роботи

Мобільна гра казуального жанру з використанням 3D-моделювання

---

Затверджена наказом по ЧНУ від «17» березня 2023 р. № 60

2. Строк представлення кваліфікаційної роботи «\_\_\_\_\_» \_\_\_\_\_ 2023 р.

3. Очікуваний результат роботи та початкові дані, якщо такі потрібні

Розроблена мобільна гра головоломка казуального жанру

---

4. Перелік питань, що підлягають розробці

Дослідження та аналіз предметної області, аналіз ігрового ринку,  
створення 3д моделей, розробка дизайну меню та ігрових рівнів

---

5. Перелік графічних матеріалів

Презентація

---

6. Завдання до спеціальної частини

Аналіз охорони праці на робочих місцях фахівців з інформаційних  
технологій

---

7. Консультанти:

Консультант	Кафедра (організація)	Частина роботи
Алексєєва А.О.	Кафедра екології	Спеціальна частина з охорони праці

Керівник роботи

\_\_\_\_\_ Канд. техн. наук, доцент Бойко Анжела Петрівна \_\_\_\_\_  
(посада, прізвище, ім'я, по батькові) (підпис)

«\_\_\_\_\_» \_\_\_\_\_ 2023 р.

Завдання прийнято до виконання

\_\_\_\_\_ Мальцева Владислава Михайлівна \_\_\_\_\_  
(прізвище, ім'я, по батькові) (підпис)

«\_\_\_\_\_» \_\_\_\_\_ 2023 р.

Дата видачі завдання «\_\_\_\_\_» \_\_\_\_\_ 2023 р.

**КАЛЕНДАРНИЙ ПЛАН**  
**виконання кваліфікаційної роботи**

**Тема:** Мобільна гра казуального жанру з використанням 3D-моделювання

№	Найменування роботи	Початок	Закінчення	Примітки
1.	Розробка та затвердження завдання на виконання КРБ	09.11.2022	06.12.2022	Виконано
2.	Огляд літератури за темою роботи	06.12.2022	13.01.2023	Виконано
3.	Складання календарного плану КРБ	06.03.2023	13.06.2023	Виконано
4.	Аналіз предметної області	02.03.2023	06.03.2023	Виконано
5.	Розробка проєктних рішень	27.01.2023	16.02.2023	Виконано
6.	Моделювання та конструювання ПЗ	11.02.2023	23.05.2023	Виконано
7.	Кодування, тестування та апробація розробленого ПЗ, аналіз результатів тестування, розробка керівництва користувача	16.03.2023	06.06.2023	Виконано
8.	Розробка спеціальної частини з охорони праці	28.05.2023	29.05.2023	Виконано
9.	Відгук керівника КРБ	06.06.2023	06.06.2023	Виконано
10.	Оформлення КРБ та презентації	06.06.2023	06.06.2023	Виконано
11.	Попередній захист	07.06.2023	07.06.2023	Виконано
12.	Рецензування	11.06.2023	11.06.2023	Виконано
13.	Завершення оформлення КРБ та презентації	13.06.2023	13.06.2023	Виконано
14.	Захист кваліфікаційної роботи	28.06.2023	28.06.2023	Виконано

Розробив студент Мальцева Владислава Михайлівна  
(прізвище, ім'я, по батькові) \_\_\_\_\_ (підпис)

«\_\_\_» \_\_\_\_\_ 2023 р.

Керівник роботи канд. техн. наук, доцент Бойко Анжела Петрівна  
(посада, прізвище, ім'я, по батькові) \_\_\_\_\_ (підпис)

«\_\_\_» \_\_\_\_\_ 2023 р.

## АНОТАЦІЯ

до кваліфікаційної роботи бакалавра

«Мобільна гра казуального жанру з використанням 3D-моделювання»

Студентка 409 гр.: Мальцева В.М.

Керівник: канд. техн. наук, доцент Бойко А.П

Мобільні ігри вже заповнили весь світ, в них грають як і діти, так і дорослі. Вони приваблюють користувачів своїм дизайном, сюжетом, персонажами та механікою. Їдучи в автобусі або чекаючи когось, чи іноді просто хочеться згаяти час, то ми звертаємось саме до мобільних ігор. Благо індустрія мобільних розваг зробила все, щоб така можливість у нас була. Зараз в AppStore і Google Play можна знайти ігри на будь-який смак.

Метою роботи є розвиток логічного мислення, уваги, стратегічних навичок за рахунок розробки казуальної гри головоломки з використанням тривимірного моделювання.

Об'єктом роботи є процес розробки мобільної гри казуального жанру з використанням тривимірного моделювання, використовуючи середовище розробки Unity.

Предметом роботи є сучасні технології розробки мобільних застосунків розважального характеру.

Для досягнення поставленої мети необхідно виконати наступні завдання:

- дослідити категорії ігор;
- проаналізувати ігровий ринок;
- сформулювати головні вимоги майбутнього застосунку;
- спроектувати необхідні 3д моделі;
- створити анімації;
- створити елементи дизайну інтерфейсу;
- реалізувати програмну частину застосунку;
- провести тестування.

Кваліфікаційна робота бакалавра складається з вступу, чотирьох розділів, висновків, спеціального розділу з охорони праці та переліку джерел посилання.

У вступі визначається актуальність, науково-практичне значення обраної теми, мету і завдання роботи, об'єкт та предмет дослідження.

У першому розділі кваліфікаційної роботи проводиться аналітична частина сфери ігрової розробки.

У другому розділі було описане моделювання та технічне проєктування проєкту за допомогою створень діаграм.

У третьому розділі розглядається моделювання та програмна реалізація.

У четвертому розділі продемонстровано тестування програмного забезпечення, калькуляція проєкту та мінімальні вимоги.

У спеціальній частині з охорони праці та безпеки в надзвичайних ситуаціях йдеться про техніку безпеки при роботі в офісних приміщеннях із комп'ютерним обладнанням та допоміжними приладами.

Структура та обсяг роботи. Кваліфікаційна робота бакалавра складається з анотації, вступу, чотирьох розділів, висновків, переліку джерел посилання з 17 джерел. Основна частина роботи становить 67 сторінок, серед яких 55 рисунків та 12 таблиць.

Ключові слова: *мобільна гра, головоломка, гра-головоломка, логіка, статистика, ігровий рушій, Unity, моделювання, проєктування, анімація, діаграма, програмне забезпечення.*

## ABSTRACT

of the Bachelor's Thesis

" A casual mobile game using 3D modeling "

Student of group 409: Maltseva Vladyslava Mykhailivna

Supervisor: Ph.D. tech. Sciences, Associate Professor Boyko A. P.

Mobile games have already taken over the whole world, they are played by both children and adults. They attract users with their design, story, characters and mechanics. When riding the bus or waiting for someone, or sometimes we just want to pass the time, we turn to mobile games. Fortunately, the mobile entertainment industry has done everything to ensure that we have this opportunity. Now in AppStore and Google Play you can find games for every taste.

The object is the process of developing a casual mobile game using three-dimensional modeling using the Unity development environment.

The subject of the study is modern technologies for the development of mobile entertainment applications.

The purpose of the work is the development of logical thinking, attention, and strategic skills through the development of a casual puzzle game using three-dimensional modeling.

To achieve the goal, the following tasks must be completed:

- explore the categories of games.
- analyze the gaming market.
- form the main requirements of the future application.
- design the necessary 3D models.
- create animations.
- create interface design elements.
- implement the software part of the application.
- conduct testing.

The bachelor's thesis consists of an introduction, four sections, conclusions, appendices, a special section on labor protection and a list of reference sources.

The introduction defines the relevance, scientific and practical significance of the chosen topic, the purpose and task of the work, the object and subject of the research.

In the first section of the qualification work, the analytical part of the field of game development is carried out.

The second chapter described the modeling and technical design of the project with the help of diagrams.

The third chapter deals with modeling and software implementation.

The fourth chapter demonstrates software testing, project costing, and minimum requirements.

The special part on occupational health and safety in emergency situations deals with safety techniques when working in office premises with computer equipment and auxiliary devices.

Structure and scope of work. The bachelor's qualification work consists of an abstract, an introduction, four sections, conclusions, a list of reference sources has 17 sources. The main part of the work consists of 67 pages, including 55 figures and 12 tables.

Keywords: *mobile game, puzzle, puzzle game, logic, statistics, Unity, modeling, design, animation, diagram, software.*



## ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	4
ВСТУП.....	5
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	7
1.1 Аналіз ігрового ринку .....	7
1.2 Аналіз програм для моделювання.....	9
1.3 Аналіз ігор-аналогів.....	10
1.4 Специфікація вимог.....	15
Висновки до розділу 1 .....	17
2 МОДЕЛЮВАННЯ ПРОГРАМНОГО ЗАСТОСУНКУ .....	18
2.1 Створення сценаріїв використання .....	18
2.2 Побудова діаграм варіантів використання .....	21
2.3 Побудова діаграми станів.....	22
2.4 Побудова діаграм діяльностей .....	24
2.5 Побудова діаграми взаємодії.....	27
2.6 Створення москір інтерфейсу застосунку .....	28
Висновки до розділу 2 .....	30
3 ПРОГРАМНА РЕАЛІЗАЦІЯ .....	31
3.1 Створення 3D моделей .....	31
3.2 Створення освітлення.....	37
3.3 Додавання звуку та музики, його налаштування.....	42
3.4 Створення анімацій в Unity .....	46
3.5 Генерація рівню .....	48
3.6 Розробка логіки гри .....	50
Висновки до розділу 3 .....	53
4 ТЕСТУВАННЯ ГРИ-ГОЛОВОЛОМКИ.....	55
4.1 Функціональне тестування та огляд інтерфейсу .....	56
4.2 Подальший розвиток гри .....	64

Кафедра інженерії програмного забезпечення  
Мобільна гра казуального жанру з використанням 3D-моделювання

Висновки до розділу 4 .....	65
ВИСНОВКИ.....	66
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	68

## **ПЕРЕЛІК СКОРОЧЕНЬ**

КРБ – кваліфікаційна робота бакалавра

ОС – операційна система

ПЗ – програмне забезпечення

API – application programming interface

UML – unified modeling language

UX – User Experience

## ВСТУП

**Актуальність теми** кваліфікаційної роботи бакалавра зумовлена тим, що мобільні ігри вже заповнили весь світ, в них грають як і діти, так і дорослі. Вони приваблюють користувачів своїм дизайном, сюжетом, персонажами та механікою. Їдучи в автобусі або чекаючи когось, чи іноді просто хочеться згаяти час, то люди звертаються саме до мобільних ігор. Благо індустрія мобільних розваг зробила все, щоб була така можливість. Зараз в AppStore і Google Play можна знайти ігри на будь-який смак та для будь-якого віку.

Чому саме мобільні ігри? Тому що є декілька переваг щодо розробки саме мобільних ігор ніж комп'ютерних або консольних.

Переваги смартфона як ігрового пристрою очевидні:

1. Мобільність. Люди майже не випускають телефон з рук, вони займають мінімальну кількість місця, ніж портативні консолі чи комп'ютер.

2. Доступність. Мобільні ігри можуть бути легко завантажені з магазину додатків і встановлені на смартфон або планшет. Вони доступні в будь-який час і в будь-якому місці, де є доступ до Інтернету або зарядженого акумулятора.

3. Простота. Багато мобільних ігор прості у використанні, з легким інтерфейсом і інтуїтивно зрозумілим геймплеєм. Це робить їх доступними для всіх користувачів, незалежно від рівня досвіду.

4. Різноманітність. В магазинах додатків є багато різних жанрів мобільних ігор, від стратегій і пригод до шутерів та головоломок.

**Об'єктом** є процес розробки мобільної гри казуального жанру з використанням тривимірного моделювання, використовуючи середовище розробки Unity.

**Предметом** є сучасні технології розробки мобільних застосунків розважального характеру.

**Метою роботи** є розвиток логічного мислення, уваги, стратегічних навичок за рахунок розробки казуальної гри головоломки з використанням тривимірного моделювання.

Для досягнення поставленої мети необхідно виконати наступні **завдання**:

- дослідити категорії ігор;
- проаналізувати ігровий ринок;
- сформулювати головні вимоги майбутнього застосунку;
- спроєктувати необхідні 3д моделі;
- створити анімації;
- створити елементи дизайну інтерфейсу;
- реалізувати програмну частину застосунку;
- провести тестування.

## 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

### 1.1 Аналіз ігрового ринку

Спочатку було проаналізовано саме якими пристроями користуються частіше для ігор. За нижче вказаною діаграмою можна спостерігати дохід ринку ігор за типом пристрою [1]. На ній видно, що мобільні пристрої приносять максимальний дохід протягом кількох років. Тому було обрано створення саме мобільної гри.

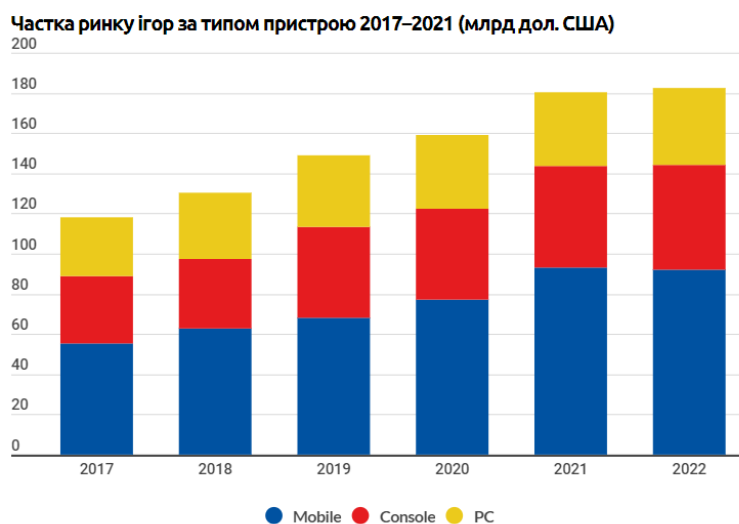


Рисунок 1.1 – Діаграма ринку ігор за типом пристрою [1]

У 2022 році світовий ринок мобільних ігор заробив близько 92,2 мільярда доларів. Це становить 52% загального світового доходу від ігор того року [2].

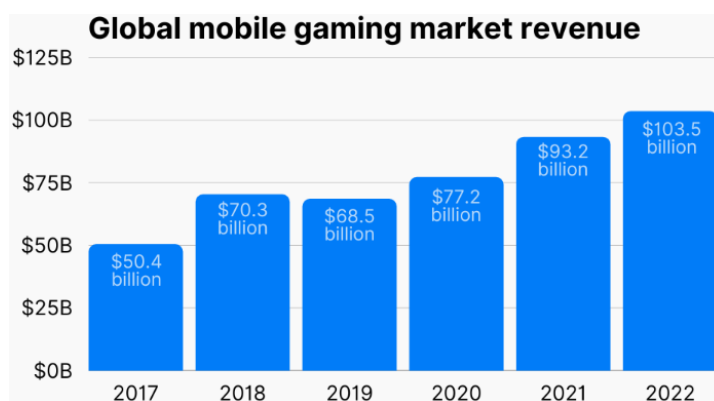


Рисунок 1.2 – Діаграма світового ринку мобільних ігор [2]

Для спрощення вибору жанру було використано наступну класифікацію:

- **казуальний жанр:** бойовики, пригоди, аркади, казуальні та головоломки.
- **класичний жанр:** настільні, карткові, казино, вікторини та словесні ігри.
- **середній жанр:** ігри для кількох гравців, рольові ігри, симулятори, стратегії, гонки та спортивні ігри.



Рисунок 1.3 – Статистика популярності жанрів мобільних ігор за 2022 рік

Казуальні ігри є дуже популярними серед гравців, і це не випадково. Є такі причини, чому казуальні ігри можуть бути дуже цікавими та класними.

Простота геймплею. Казуальні ігри зазвичай мають простий та легкий для зрозуміння геймплей, що дозволяє швидко зрозуміти та насолодитися грою навіть тим, хто ще не має досвіду у грі.

Легкість доступу. Казуальні ігри можна грати на різних платформах, включаючи смартфони, планшети, персональні комп'ютери та ігрові консолі. Крім того, вони зазвичай коштують менше та не вимагають великої кількості часу на гру.

Різноманітність. Казуальні ігри можуть включати в себе різні жанри та теми, від простих головоломок, стратегій та пригод до ігор з елементами різних жанрів. Це дозволяє гравцям знайти гру, яка їм подобається та задовольнити їхні потреби у грі.

Можливість грати з друзями. Казуальні ігри часто мають мультиплеєрний режим, що дозволяє гравцям грати з друзями або іншими гравцями з усього світу та насолоджуватися грою разом.

Відсутність стресу. Казуальні ігри зазвичай не містять великої кількості насильства та агресії, тому гравці можуть розслабитися та насолоджуватися грою без стресу та напруги.

## 1.2 Аналіз програм для моделювання

3Ds Max – програма для полігонального 3D моделювання та анімації. Використовується в дизайні, для створення відеоігор і спецефектів у кіно. Володіє величезною бібліотекою доповнень, які роблять програму корисною і в інших галузях.

Blender – Безкоштовний повнофункціональний аналог 3DMax з відкритим кодом. Також має величезну кількість доповнень, які створюються самим ком'юніті. Більше заточений під створення моделей для відеоігор.

Maya – редактор, який вміє все теж, що і 3D-max, але по-іншому. Найчастіше застосовується в кіноіндустрії і підходить для вирішення нестандартних завдань.

Cinema 4D – 3D-редактор і програма для створення візуальних ефектів і анімації, застосовується в відеомонтаж і кінематографі.

Zbrush – 3D-редактор для скульптінга і текстуровання.

Але для виконання проекту обиралося серед 3ds Max та Blender. Бо це два дуже популярні програмні засоби. Обидві програми мають свої переваги та недоліки, які залежать від потреб користувача.

### Переваги 3ds Max:

- висока продуктивність на великих проектах;
- повна інтеграція з іншими продуктами Autodesk;
- інтуїтивний і зрозумілий інтерфейс;
- велика кількість різних інструментів для моделювання, анімації та рендерингу;
- широкі можливості візуалізації за допомогою Vray, Mental Ray та інших рендерів.



### **Недоліки 3ds Max:**

- дорога програма зі значною вартістю ліцензії;
- обмежена можливість використання на інших операційних системах крім Windows;
- не всі функції працюють на повну потужність на слабкому обладнанні.

### **Переваги Blender:**

- безкоштовна програма з відкритим вихідним кодом;
- кросплатформенна, що дозволяє використовувати на різних операційних системах;
- можливість додаткової настройки і зміни різних параметрів;
- можливість розширення можливостей за допомогою плагінів та скриптів;
- підтримка різних форматів файлів та протоколів;
- наявність можливості скульптингу;
- можливість створення текстур.

### **Недоліки Blender:**

- інтерфейс може бути складним для новачків;
- обмежена підтримка відомих рендерерів;

Отже, було прийнято рішення використовувати Blender для моделювання об'єктів мобільної гри. Головними перевагами було виділено, що Blender є повністю безкоштовним та не так сильно навантажує комп'ютер.

## **1.3 Аналіз ігор-аналогів**

«Water Pipes» від Mobiloids є 2D грою сенс якої становить створити один ланцюг труб, повертаючи кожний елемент, з'єднавши початкову точку у вигляді крану та кінцеву точку у вигляді механізму [3]. За власними спостереженнями та відгукам інших користувачів головною перевагою є різноманітність рівнів, але

до недоліків було додано як незрозумілу оцінку проходження рівнів, відсутність музики та на високих рівнях дається мало часу на проходження.

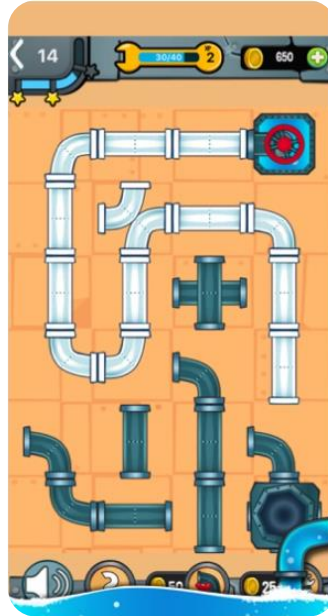


Рисунок 1.4 – Вигляд рівню «Water Pipes»

Таблиця 1.1 – Опис гри «Water Pipes»

Назва	Water Pipes
Розробник	Mobiloids
Архітектура	Mobile application
Мова реалізації	Невідомо
Перелік функцій, характеристик	Обертання фігур Рахування обертів Розрахунок кількості зірок Проходження рівнів на певну кількість зірок Перепроходження рівнів 2D
Переваги	Різноманітність рівнів Не обов'язкове підключення до інтернету Мала кількість реклами

Кінець таблиці 1.1

Недоліки	<p>Мала кількість рівнів</p> <p>Рівні з обмеженням на час, що не дозволяє розслабитися гравцю</p> <p>Незрозуміла оцінка проходження гри</p> <p>Відсутність музики</p>
Джерело інформації	<p><a href="https://play.google.com/store/apps/details?id=com.mobiloids.waterpipespuzzle&amp;hl=en_US&amp;gl=US">https://play.google.com/store/apps/details?id=com.mobiloids.waterpipespuzzle&amp;hl=en_US&amp;gl=US</a></p>

Наступним аналогом стала гра «Pipe - logic puzzles» від App2Eleven. Вона є також представником 2D ігор жанру головоломок у вигляді поєднань труб, але з особливістю, що труби різного кольору, тому шляхів з'єднань більше [4]. Гра має стару графіку, але яскраву картинку. Головними перевагами є велика кількість та різноманітність рівнів. Недоліками є набридлива музика та те що гра часто вилітає.



Рисунок 1.5 – Вигляд рівню «Pipe - logic puzzles»

Таблиця 1.2 – Опис гри «Pipe - logic puzzles»

Назва	Pipe - logic puzzles
Розробник	App2Eleven
Архітектура	Mobile application
Мова реалізації	Невідомо
Перелік функцій, характеристик	Обертання фігур Рахування обертів Проходження рівнів на певну кількість зірок Перепроходження рівнів Використання підказок 2D
Переваги	Різноманітність рівнів Є підказки Велика кількість рівнів Не обов'язкове підключення до інтернету
Недоліки	Велика кількість реклами Гра підгальмовує, підвисає Нерізноманітна музика, швидко набридає
Джерело інформації	<a href="https://play.google.com/store/apps/details?id=air.App2Eleven.LogicGamePuzzlesPipe&amp;hl=en_US&amp;gl=US">https://play.google.com/store/apps/details?id=air.App2Eleven.LogicGamePuzzlesPipe&amp;hl=en_US&amp;gl=US</a>

Гра «Aquavias» від Dreamy Dingo на відміну від попередніх має 3D графіку, це єдина 3D гра жанру головоломок труб, яку було знайдено на просторах інтернету [5]. Вона має дуже приємну графіку, гарний дизайн, але за власним спостереженням та відгукам інших користувачів вона має тільки 30 рівнів, а це дуже мало. Також до недоліків було додано неможливість послідовного проходження рівнів, після завершення рівню гравця відправляє на мапу рівнів, щоб він обрав наступний. Через 10 рівнів це починає набридати.



Рисунок 1.6 – Вигляд рівню «Aquavias»

Таблиця 1.3 – Опис гри «Aquavias»

Назва	Aquavias
Розробник	Dreamy Dingo
Архітектура	Mobile application
Мова реалізації	Невідомо
Перелік функцій, характеристик	Обертання фігур. Рахування обертів. Розрахунок кількості зірок. Вмикання/вимикання музики. Перепроходження рівнів. Встановлення обмежень на рівні. 3D .
Переваги	Різноманітність рівнів. Заспокійлива музика. Гарна графіка. Не обов'язкове підключення до інтернету.

### Кінець таблиці 1.3

Недоліки	<p>Мала кількість рівнів.</p> <p>Реклама через кожен рівень.</p> <p>Рівні з обмеженням на час, що не дозволяє розслабитися гравцю.</p> <p>Неможливо проходити рівні послідовно, постійно виходить на мапу рівнів.</p>
Джерело інформації	<p><a href="https://play.google.com/store/apps/details?id=ua.krou.aqueducts&amp;hl=en_US&amp;gl=US">https://play.google.com/store/apps/details?id=ua.krou.aqueducts&amp;hl=en_US&amp;gl=US</a></p>

## 1.4 Специфікація вимог

Проаналізувавши ігри-аналоги від інших компаній, було визначено особливості створюваного застосунку. Так як на ринку було знайдено тільки одну гру жанру головоломки з трубами з 3D моделюванням, то використовуватиметься саме вона. Ще з особливостей буде великий плейлист заспокійливої музики, яка має менші ризики набридненню користувачеві. Також буде поступове проходження рівнів без виходу на мапу рівнів.

Далі було визначено основні функції:

- обертання фігур;
- зарахування правильно побудованого шляху;
- рахування обертів;
- розрахунок кількості зірок за пройдений рівень;
- вибір рівня;
- розблокування рівня;
- перехід з головного меню на мапу рівнів;
- вмикання/вимикання музики;
- перепроходження рівнів;
- встановлення обмежень або цілей на рівні;

**Призначення системи:** розвиток логічного мислення, уваги, стратегічних навичок за рахунок розробки казуальної гри головоломки з використанням тривимірного моделювання.

**Характеристики користувачів:** Людина будь-якого віку.

**Межі проєкту:** Крайня дата завершення роботи над ПЗ – 10.05.2023

**Сфера застосування:** Дана гра є програмним забезпеченням розважливого характеру, що сприяє розвитку логіки. Тому головною сферою застосування є розваги та заспокоєння.

**Загальною структурою і складом системи** є сам застосунок.

**Загальним обмеженням** є кількість достатнього місця на девайсі.

#### **Вимоги до інформаційного забезпечення**

**Джерела і зміст вхідної інформації (даних):** головним джерелом інформації користувач.

Нормативно-довідкова інформація (класифікатори, довідники тощо): вимоги відсутні.

Вимоги до способів організації, збереження та ведення інформації

Усі данні зберігаються на вбудовані пам'яті мобільного телефону.

**Вимоги до технічного забезпечення:** смартфон з версією Android від 7.0, будь-яким розширенням екрану та 300 Мбайт вільного місця.

#### **Вимоги до програмного забезпечення**

Системне програмне забезпечення: було використано ПЗ Unity.

Мережне програмне забезпечення: ОС Windows 10.

Мова і технологія розробки ПЗ: C#.

#### **Вимоги до зовнішніх інтерфейсів**

Інтерфейс користувача: інтерфейс з мінімалістичним дизайном задовольняє всім вимогам користувача

Апаратний інтерфейс: мінімальна версія Android 7.0.

Програмний інтерфейс: для створення гри на Unity було використано дві категорії Unity Scripting API :UnityEditor (Animations, Events тощо) та UnityEngine (Analytics, Audio тощо).

Комунікаційний протокол: відсутній.

### **Властивості програмного забезпечення**

Доступність: кожен користувач має доступ до гри.

Переносимість: мінімальна версія Android 7.0.

Продуктивність: залежить від характеристик смартфона або планшета

### **Висновки до розділу 1**

В даному розділі кваліфікаційної роботи бакалавра було проаналізовано ігровий ринок застосунків, розібрано жанри ігор, виявлено переваги та недоліки програм для моделювання 3д об'єктів, розглянуто та проведено аналіз аналогів ігор-головоломок казуального жанру. Були виокремлені переваги та недоліки кожного застосунку та сформовано функціонал та особливості створюваного застосунку. Також було розгорнуто сформовано специфікацію вимог для проекту.



## 2 МОДЕЛЮВАННЯ ПРОГРАМНОГО ЗАСТОСУНКУ

Розділ моделювання програмного застосунку є однією з ключових складових при проектуванні і розробці програмного забезпечення. Він визначає основні концептуальні та архітектурні аспекти створення програмного застосунку та надає відповіді на питання про те, як програма буде функціонувати, які будуть його основні функції та які вимоги він повинен задовольняти [6].

У цьому розділі розглянуто різні методи та техніки моделювання програмного застосунку, які можуть бути використані під час проектування та розробки програмного забезпечення.

Також досліджено різні методи моделювання програмного застосунку, такі як діаграми UML (Unified Modeling Language), потокові діаграми, діаграми класів та інші. Розглянуто, як ці методи допомагають уявити структуру та поведінку програми.

### 2.1 Створення сценаріїв використання

**Usecase 1.** Користувач завантажує гру. Обирає рівень на якому зупинився. Йде завантаження рівню. На етапі проходження рівню користувач вийшов в меню рівнів. Прогрес не збережено.

**Usecase 2.** Користувач завантажує гру. Обирає рівень, який хоче перепройти на кращий результат. Пройшовши рівень, зберігається найкращий результат. Користувач проходить до наступного рівню.

**Usecase 3.** Користувач завантажує гру. Обирає рівень. Завантажується рівень. Пройшовши рівень, користувач обирає пройти рівень спочатку. Завантаження рівню. Користувач проходить рівень.

**Usecase 4.** Користувач завантажує гру. Переходить до мапи рівнів. Обирає рівень для проходження. Рівень завантажується. Користувач переходить до налаштувань та налаштовує гучність музики та звуків.

### **Usecase 5 в короткій формі**

Користувач завантажує гру. Обирає рівень на якому зупинився. Йде завантаження рівню. Якщо користувач виконав всі умови, то рівень пройдено. Завантажується меню з результатом. Користувач переходить на новий рівень.

### **Usecase 5 в поверхневій формі**

Головний сценарій (успішний)

Користувач завантажує гру. Обирає рівень. Йде завантаження рівню. Якщо користувач виконав всі умови, то рівень пройдено. Завантажується меню з результатом. Користувач обирає «Перейти до наступного рівня».

Альтернативні сценарії:

1. Користувач проходив рівень раніше, то найкращий результат перезаписується.
2. Користувач натиснув на заблокований рівень, рівень не завантажується.
3. Користувач в меню результату рівня обирає перейти до головного меню.
4. Користувач в меню результату рівня обирає «Перепройти рівень».
5. Користувач під час проходження рівня вийшов з гри, прогрес не зберігся.

### **Usecase 5 в повній формі**

Use Case Name: Проходження гри

Scope: Гра-головоломка.

Level: Мета користувача (user-goal).

Primary Actor: Гравець.

Stakeholders and interests:

1. Гравець. Розвиток логічного мислення, розслаблення, скорочення часу за допомогою гри.
2. Розробник. Розробка застосунку розважального характеру, можливий заробіток встановлюючи рекламу.

Preconditions: Користувач завантажив застосунок.

Main Success Scenario:

1. Користувач завантажує гру.
2. Користувач може вимкнути музику.
3. Користувач обирає рівень.
4. Йде завантаження рівню.
5. Користувач виконує всі умови рівню.
6. Рівень пройдено.
7. Завантажується меню з результатом.
8. Користувач обирає «Перейти до наступного рівня».

Extensions:

1. Застосунок не може завантажитися або видає помилку:
  - a. Користувач перезавантажує застосунок.
2. Користувач обирає заблокований рівень:
  - a. Система видає попередження, що рівень заблокований.
3. Користувач перепроходить рівень:
  - a. Лічильник ходів стає 0.
  - b. Результат минулого проходження перезаписується.
4. Користувач пройшов рівень:
  - a. Користувач може пройти в головне меню.
  - b. Користувач може перепройти рівень.
  - c. Користувач може перейти до наступного рівня.
5. Користувач вийшов з гри під час проходження рівня:
  - a. Прогрес рівня не зберігається.

Special Requirements: Смартфон

Frequency of Occurrence: Застосунок може працювати безперервно.

Miscellaneous (Open Issues): Додати функціонал проектування рівнів для розробника.

## 2.2 Побудова діаграм варіантів використання

Діаграма використання (Use Case Diagram) - це один з типів діаграм UML, який використовується для моделювання функціональних вимог до системи з точки зору її користувачів або зовнішніх акторів [7].

У діаграмі використання актори представляють ролі, які взаємодіють з системою, а використання (use case) - це дії або послідовності дій, які користувачі можуть виконувати за допомогою системи для досягнення певної мети. Діаграма використання дозволяє описати, як користувачі взаємодіють з системою і які функції системи відповідають на їх запити [6].

Діаграми використання використовуються на етапі аналізу вимог до системи для визначення функціональності, яка повинна бути вбудована в систему, та для створення специфікації вимог.

Діаграма використання допомагає уникнути непорозумінь між командами проектування, виокремлює ключові функціональні вимоги до системи, сприяє кращому розумінню ролей, які взаємодіють з системою, та допомагає визначити, які функції можуть бути автоматизовані і як це може бути зроблено.

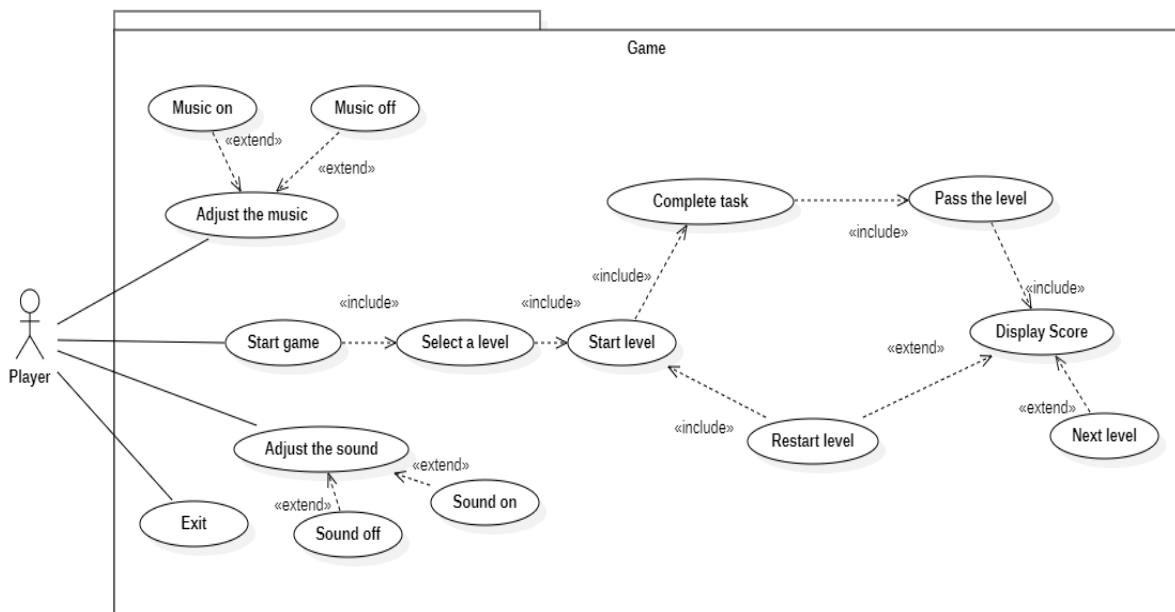


Рисунок 2.1 – Діаграма використання

Створювана гра має одну дійову особу – гравця. На діаграмі зображено, що гравець може розпочати гру, обрати рівень, проходити його. Коли гравець пройшов рівень, то він перейде до вікна з результатом, на якому може обрати перепроходження рівня або перейти до наступного. Також користувач може налаштувати музику та звуки, і вийти з самої гри.

### **2.3 Побудова діаграми станів**

Діаграма станів (state diagram) - це графічний засіб моделювання поведінки системи, що показує різні стани, у яких може перебувати система та переходи між ними. Діаграма станів дозволяє показати, як система поводить себе в різних ситуаціях, в залежності від зміни вхідних даних, дій користувачів чи інших факторів.

Діаграми станів використовуються в проектуванні програмного забезпечення для моделювання поведінки програми в різних ситуаціях, таких як старт, пауза, завершення роботи, помилки і т.д. Вони допомагають розуміти, як система працює в різних умовах, і дозволяють програмістам та аналітикам легше розробляти та тестувати програмне забезпечення [6].

Отже, побудова діаграм станів може допомогти візуалізувати та розуміти поведінку системи, а також виявити можливі проблеми в її функціонуванні.

Щоб побудувати діаграму станів, спочатку потрібно визначити всі можливі стани системи. Далі потрібно визначити, які події або вхідні дані можуть спричинити перехід між станами. Та нарешті, потрібно зобразити ці стани та переходи на графіці.

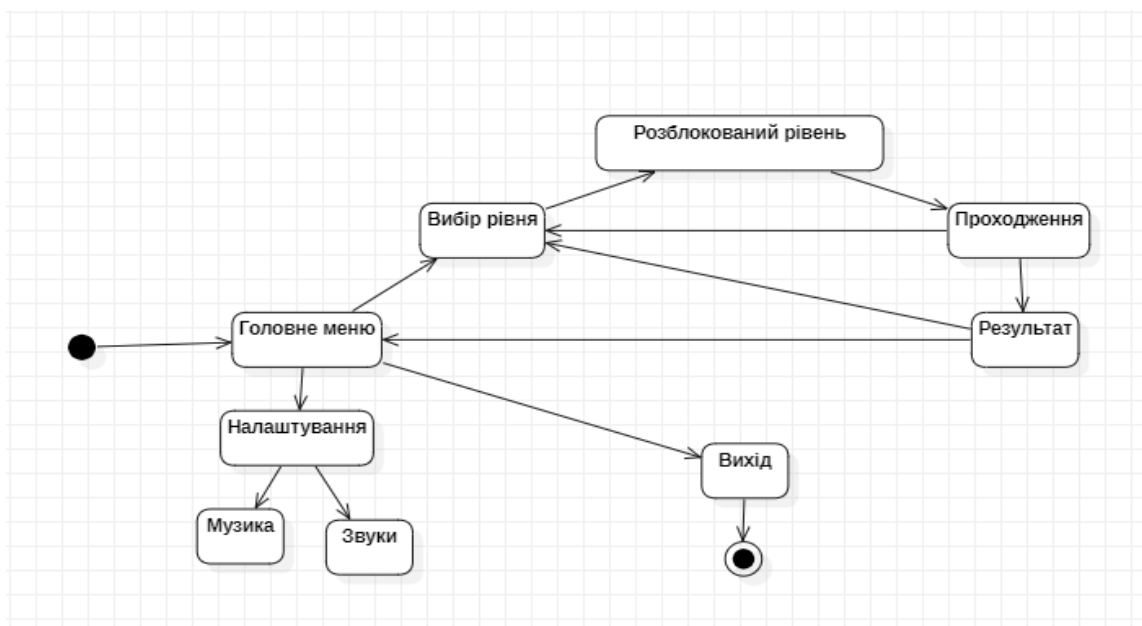


Рисунок 2.2 – Загальна діаграма станів та переходів гри

На загальній діаграмі станів відображено стани переходів користувача по етапам гри. Відвідування головного меню, вікна налаштувань та перехід на мапу рівнів. Після вибору рівню, Гра його запускає, відбувається проходження гри гравцем. Під час проходження гравець може вийти на мапу рівнів. Якщо рівень пройдено, то висвічується результат.

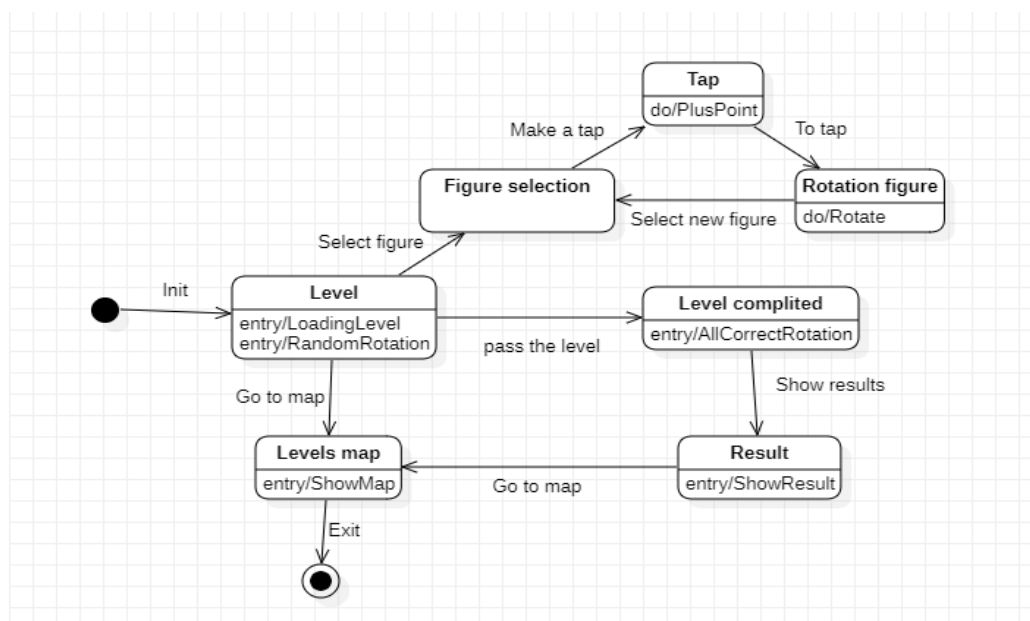


Рисунок 2.3 – Діаграма станів та переходів елементу проходження рівня

Наступна діаграма станів відображає елемент проходження рівня. Як тільки рівень завантажено гравець проходить рівень, обертаючи фігури тапом. Якщо рівень пройдено, то висвічується результат.

## 2.4 Побудова діаграм діяльностей

Діаграма діяльностей (activity diagram) - це графічний засіб моделювання, що показує послідовність дій, що відбуваються в системі в процесі виконання діяльності. Вона дозволяє представити процеси та діяльності системи, включаючи дії, рішення та переходи між ними [10].

Основною метою діаграми діяльностей є відображення логіки процесу або алгоритму. Вона дозволяє розбити складні процеси на прості кроки та ілюструвати послідовність виконання цих кроків. Кожен крок представляється у вигляді дії, яку виконує об'єкт або система.

Діаграма діяльностей складається з елементів, таких як дії, рішення, розвилки, злиття та переходи між ними. Дії відображають окремі кроки або дії, які виконуються в процесі виконання діяльності. Рішення відображають точки в графі, де процес повинен змінити свій шлях в залежності від умов, які вказуються на вході до процесу. Розвилки та злиття відображають місця в графі, де процес розділяється на два або більше шляхів або де кілька шляхів знову об'єднуються в один. Переходи відображають послідовність дій та переходів між ними.

Щоб побудувати діаграму діяльностей, спочатку потрібно визначити всі можливі дії та їх послідовність. Далі потрібно зобразити ці дії на графіці та з'єднати їх лініями, щоб показати послідовність. Нарешті, потрібно додати умови або розгалуження, які можуть відбутися в процесі.

На даній діаграмі зображено вибір рівня (рис. 2.4 (а)) Відтворюється вибір рівня, якщо він розблокований, то обирається наступний рівень, якщо ні, то запускається попередній розблокований.

Наступна діаграма на рисунку 2.4 (б) відображає діяльність самого застосунку. Спочатку запускається гра, вона працює, та гравець може її завершити, і потім якщо трапляється помилка, то гра закривається.

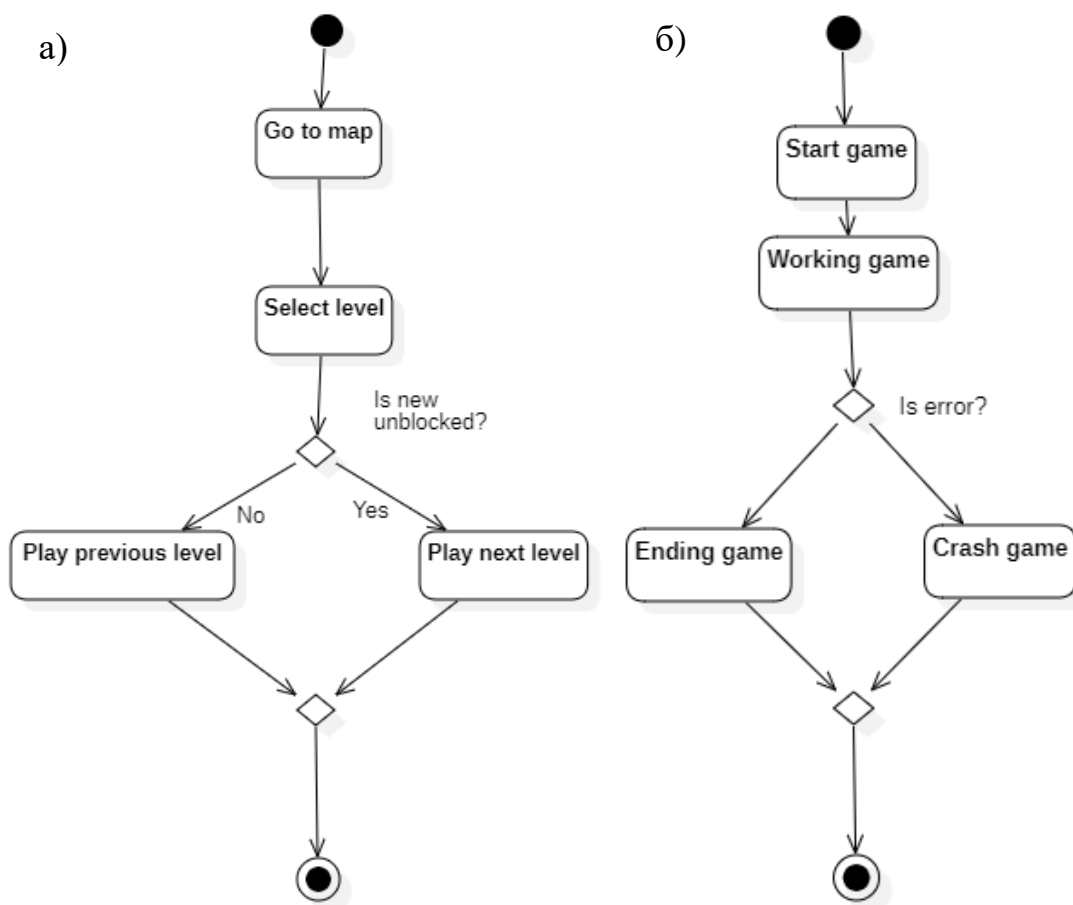


Рисунок 2.4 – Діаграми діяльності вибору рівня (а) та роботу застосунку (б)



Третя діаграма відображає сам процес геймплею, що система генерує рівень, гравець виконує n- кількість ходів для досягнення перемоги. Кожне натискання рахується. Якщо всі фігури правильно повернуті, то рівень пройдено.

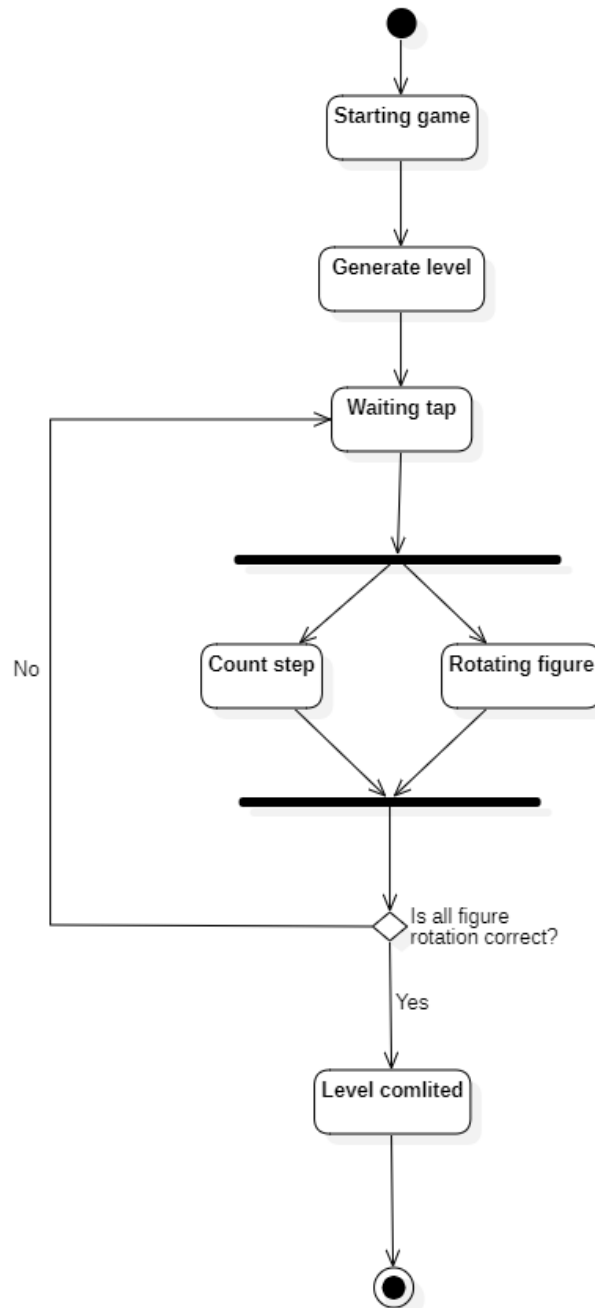


Рисунок 2.5 – Діаграма діяльності

Отже діаграми діяльності дозволяють нам представити процеси та діяльності системи, включаючи дії, рішення та переходи між ними.

## 2.5 Побудова діаграми взаємодії

Діаграми взаємодії - це графічні зображення, які допомагають візуалізувати взаємодію об'єктів та подій в процесі обробки інформації варіантів використання. Їх головними елементами є об'єкти, які інкапсулюють дані та поведінку, та повідомлення, які вказують на виклик функцій іншими об'єктами.

Діаграми послідовності відображають взаємодію об'єктів у послідовності подій у часі. Кожен об'єкт має свою лінію життя, яка представляє час життя об'єкта на діаграмі, та візуалізується вертикальною штриховою лінією. Повідомлення відображаються між лініями життя об'єктів, показуючи, що один об'єкт викликає функцію іншого.

Кооперативні діаграми, натомість, організовані навколо об'єктів та їх взаємодії. Кожен об'єкт має свій власний простір, де відображається його поведінка та взаємодія з іншими об'єктами.

У кожного об'єкта є свої атрибути та операції. Атрибути - це дані, які інкапсулюються в об'єкті, а операції - це функції, які об'єкт може виконувати. Клас, до якого належить об'єкт, визначає його дані та поведінку.

Взаємодія між гравцем та грою може бути описана в термінах подій. Гравець розпочинає рівень, гра його завантажує. Гра відображає розташування елементів на ігровому полі. Гравець обертає фігури, натискаючи на них. Гравець завершує гру, коли всі фігури знаходяться у правильному напрямку. Далі він або перепроходить рівень, або переходить до наступного.

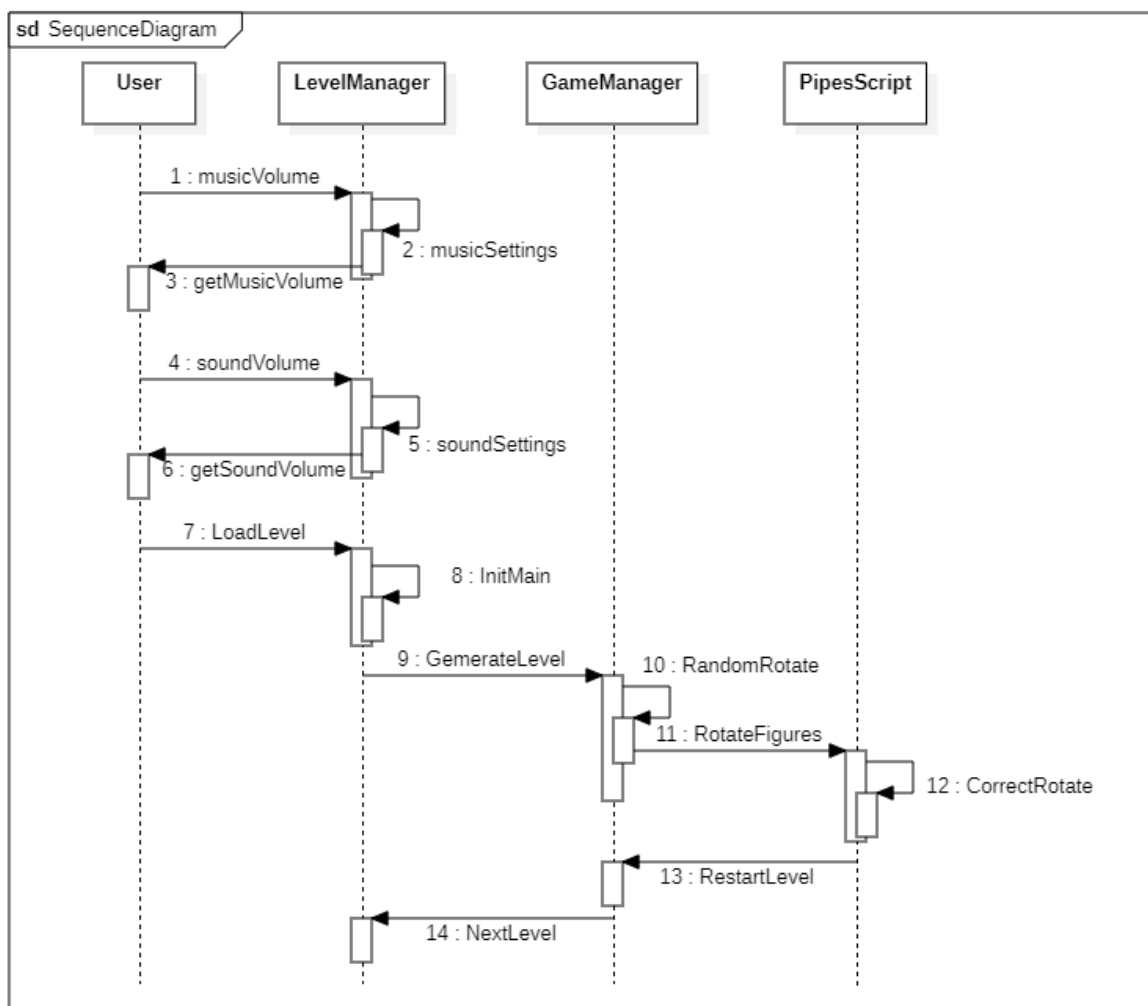


Рисунок 2.6 – Діаграма взаємодії

Узагальнюючи, діаграми взаємодії допомагають візуалізувати взаємодію об'єктів та подій в процесі обробки інформації варіантів використання.

## 2.6 Створення макету інтерфейсу застосунку

Макет (макет) – це візуальне представлення інтерфейсу або дизайну веб-сайту, мобільного застосунку або іншого програмного продукту. Він служить для демонстрації та тестування концепції, розміщення елементів на сторінці, взаємодії з користувачем та інших важливих аспектів дизайну. Макети можуть бути статичними зображеннями (наприклад, у форматі PSD або PNG) або інтерактивними прототипами, які дозволяють виконувати певні дії та переходи.

Навігація відображає спосіб, яким користувачі переходять від одного вмісту до іншого на веб-сайті, застосунку або прототипі. Вони визначають шляхи, за якими користувачі можуть переходити по різних сторінках, розділах або екранах, що створюється. Це може бути досягнуто за допомогою різних елементів інтерфейсу, таких як кнопки, посилання, меню або жестів (на сенсорних пристроях).

У макетах можна використовувати переходи по сторінкам, щоб показати, як користувачі будуть переміщатися від однієї сторінки до іншої при взаємодії з додатком або веб-сайтом. Це дає змогу протестувати логіку навігації, перевірити зручність використання та зробити оцінку загального досвіду користувача.

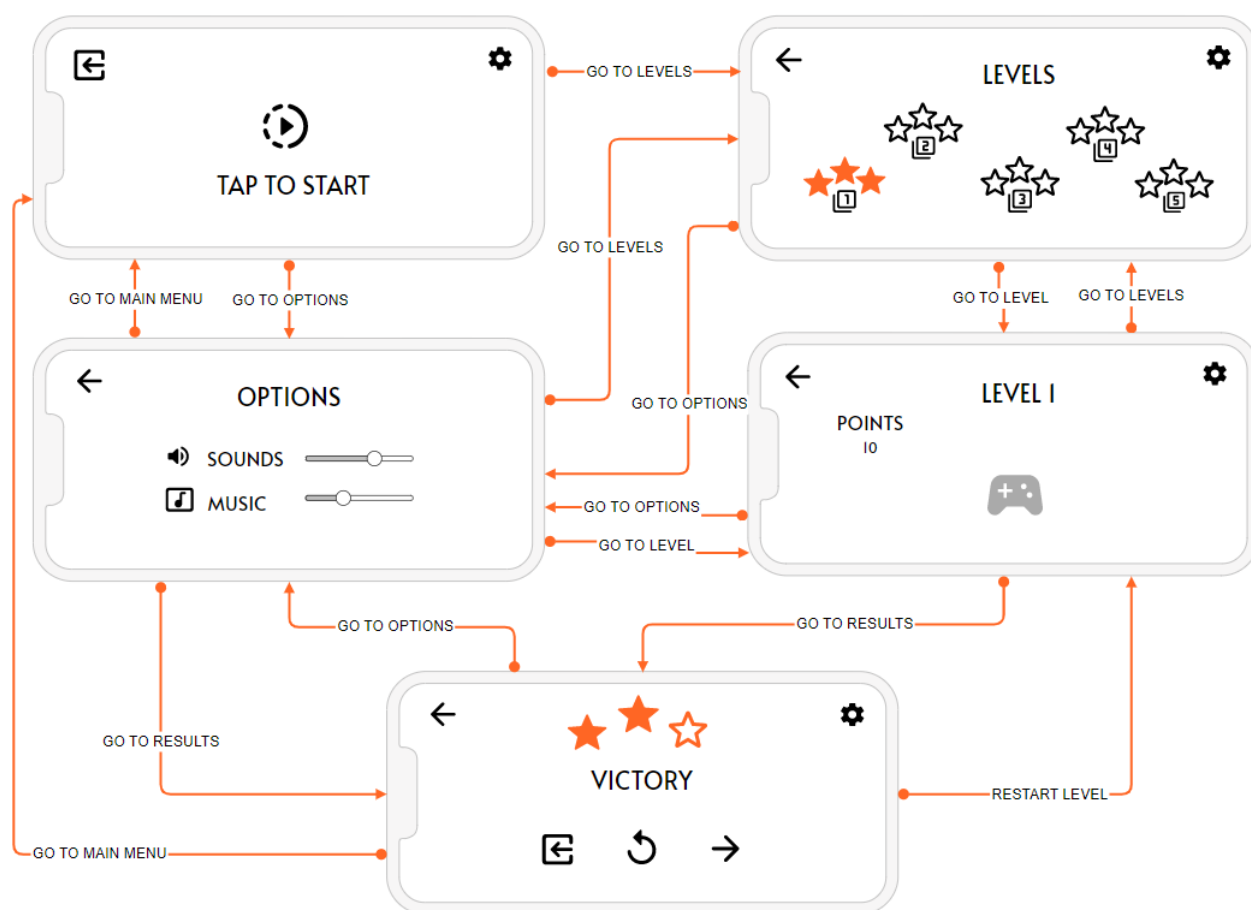


Рисунок 2.7 – Створений москір інтерфейсу з навігацією

Вікно 1. Головне меню. Цей екран відображається при завантаженні гри або користувач хоче з меню результатів повернутися на головний екран. На

цьому екрані відображені кнопки «Налаштування», «Розпочати гру» та «Вихід». Натиснувши на екран користувач потрапить у вікно «Рівні».

Вікно 2. Мапа рівнів. На це меню потрапити можна з головного екрану або з конкретного рівня. На цьому екрані розташовані мапа рівнів, кнопки «Налаштування» та «Вихід». Обравши рівень, користувач перейде на вікно обраного рівня.

Вікно 3. Рівень. На цей екран потрапити можна тільки обравши рівень на мапі рівнів. На ньому користувач буде проходити рівень. Вікно має кнопки «Налаштування» та «Повернутися до попередньої сторінки», кількість створених ходів. Пройшовши рівень користувач перейде до вікна з результатом.

Вікно 4. Результат. На екрані відображається кількість набраних зірочок, кнопки «Вихід», «Перепройти рівень», «Перейти на наступний рівень» та «Повернутися до попередньої сторінки».

Вікно 5. Налаштування. В цьому меню користувач може налаштувати гучність звуків та музики. На цю сторінку користувач може потрапити з всіх вікон.

## **Висновки до розділу 2**

Діаграми станів, взаємодії та діяльностей є важливим інструментом для моделювання процесів та систем, які можуть бути складними або недоступними для прямого спостереження. В другому розділі було продемонстровано використання діаграм взаємодії, стану та діяльності. Вся ці діаграми допомогли представити процеси та діяльності системи, включаючи дії, рішення та переходи між ними та вони допомогли у зрозумінні того, як працює система, та у виявленні проблем на ранніх етапах розробки.

### 3 ПРОГРАМНА РЕАЛІЗАЦІЯ

Unity є однією з найпопулярніших та потужних платформ для розробки комп'ютерних та мобільних ігор, що надає розробникам широкі можливості для творчості та створення захопливих ігрових досвідів.

У цьому розділі детально розглянуто процес програмної реалізації мобільної гри-головоломки на платформі Unity. Розглянуто різні аспекти розробки гри, починаючи зі створення проекту, розміщення графічних та звукових ресурсів, програмування логіки гри, взаємодії з користувачем, оптимізації продуктивності та багато іншого.

Програмна реалізація мобільної гри на Unity вимагає розуміння основних принципів програмування та знання мови програмування, такої як C#. В розділі розглянуто основні концепції програмування в Unity, такі як створення об'єктів, скриптів, компонентів, робота зі сценами, ігровими об'єктами та анімація.

#### 3.1 Створення 3D моделей

У 3D-моделюванні об'єкти створюються з використанням різних форм та геометричних елементів, таких як куби, сфери, циліндри тощо. Можна змінювати їх розміри, форму та положення. Крім того, також є можливість надати їм різні текстури та кольори, щоб зробити їх вигляд реалістичним.

Програми для 3D-моделювання, наприклад Blender, Autodesk Maya або SketchUp, дозволяють використовувати інструменти для створення, редагування та маніпулювання об'єктами у 3D-просторі. За допомогою них можна створювати складні моделі, такі як будівлі, автомобілі, персонажі, предмети та багато іншого. Отриману 3D-модель можна використовувати для різних цілей, наприклад, для візуалізації, анімації, виробництва прототипів або навіть для використання у віртуальній реальності та іграх [13].

Для моделювання моделей для цього проекту було використано саме Blender.



Рисунок 3.1 – Логотип Blender

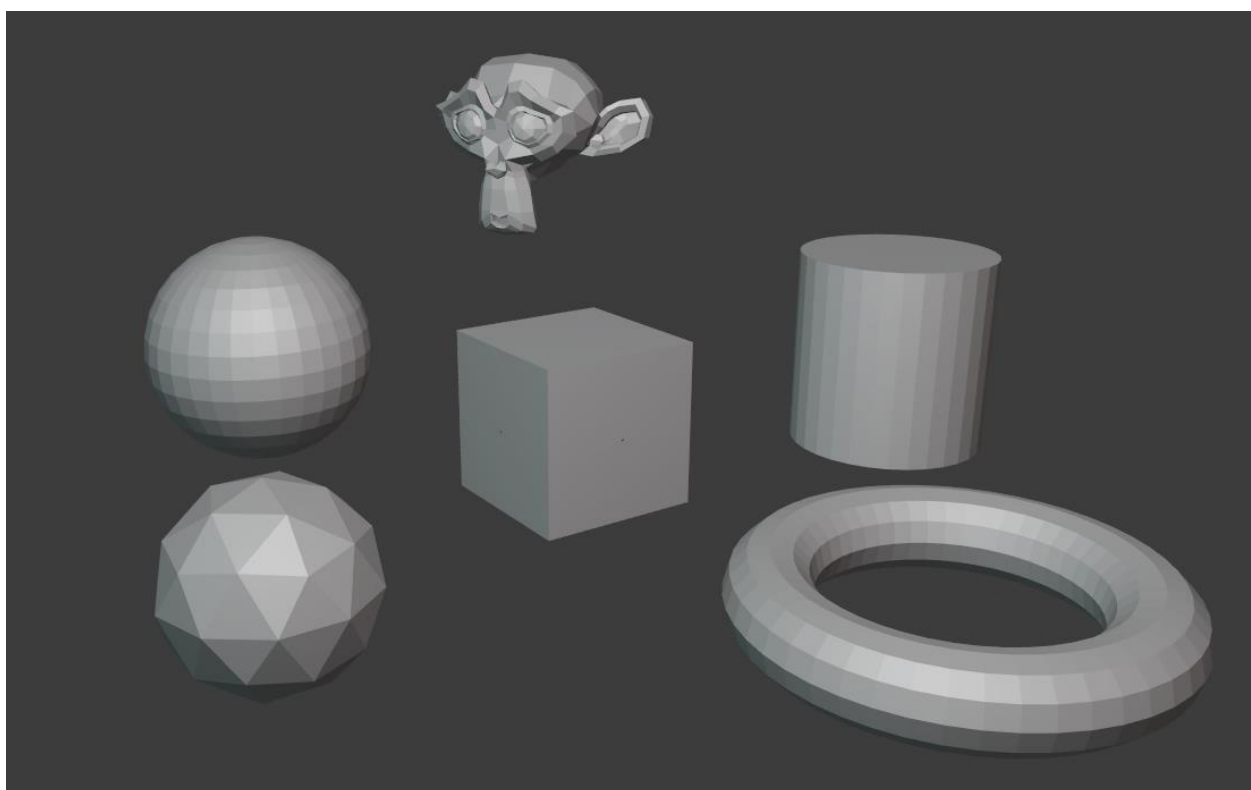


Рисунок 3.2 – Примітивні об'єкти Blender

Для гри-головоломки було обрано тематику нічного неонових міста. Тому потрібно було створити такі елементи як: пряма дорога, перехрестя доріг та дорогу, яка повертає. Саме з цими об'єктами користувач буде взаємодіяти. В якості декору потрібно побудувати будівні з неоновими вставками та підставки на яких вони будуть розміщені на сцені.

### **Побудова прямої дороги**

Для створення такої дороги використовується примітивний куб. За допомогою модифікатора площі збільшуємо довжину куба в два рази. В

результаті маємо паралелепіпед з розмірами 2 м × 4 м × 2 м. Щоб додати трохи деталей моделі, потрібно виділити два верхні найдовших ребра та застосувати модифікатор *Bevel*, який обріже гострі кути. Далі створюємо розмітку. Для цього створюємо примітив *Plane*. Вказуємо потрібні нам розміри, використовуємо модифікатор *Array* з кількістю елементів 5. Центруємо отриманий об'єкт відносно дороги. Додаємо матеріали та текстури.



Рисунок 3.3 – Створена модель прямої дороги

### **Створення T-подібної та L-подібної дороги.**

Для створення L-подібної дороги використовується примітивний куб. За допомогою модифікатора *Extrude* видовжуємо на 1 м дві бокові грані куба. Щоб створити плавний поворот, потрібно виділити потрібне ребро, використати модифікатор *Bevel* зі створенням 6 граней і відповідним радіусом для зовнішнього або внутрішнього вигину. Щоб додати трохи деталей моделі, потрібно виділити верхні ребра, окрім двох найбільших, та застосувати модифікатор *Bevel*, який обріже гострі кути. Далі створюємо розмітку. Для цього створюємо примітив *Plane*. Вказуємо потрібні нам розміри, використовуємо модифікатор *Array* з кількістю елементів 2. Центруємо отриманий об'єкт



відносно дороги. Да дублюємо об'єкт та повертаємо його на  $90^\circ$  по осі Y. Використовуємо раніше створенні матеріали.

Для створення T-подібної дороги використовуємо інструкцію вище, але за допомогою модифікатора Extrude видовжуємо на 1м три бокові грані куба. А модифікатор Bevel для вигину використовуємо вже для двох паралельних граней.



Рисунок 3.4 – Створені L-подібна та T-подібна дороги

### **Створення будівлі.**

Цілий будинок побудований з двох частин: нижня частина з дверима, частина з поверхами.

Для створення нижньої частини використовується примітивний куб з довжиною 3 м, висотою 0.5 м та шириною 2 м. Виділивши верхню грань, використовуємо модифікатор Extrude, щоб збільшити висоту на 0.01 метр. Виділивши верхній полігон, використовуємо модифікатор площі, розширивши

полігон на потрібну відстань. Опускаємо полігон та використовуємо модифікатор Extrude на 0.2 м вгору. Створюємо ще один примітивний куб для дверей. Виставляємо на відповідне місце та відповідну глибину в основну модель. Використовуємо модифікатор Boolean, щоб вирізати роз'єм дверей.

Для створення частини з поверхами використовується примітивний куб з довжиною 3 м, висотою 0.5 м та шириною 2 м. Повторюємо маніпуляцію з Extrude як для нижньої частини. Створюємо куб для вікон та використовуємо Array з кількістю елементів 9 по осі X. Виставляємо на відповідне місце та відповідну глибину в основну модель. Використовуємо модифікатор Boolean, щоб вирізати вікна. Для зменшення навантаження на систему видаляємо полігони нижній та верхній полігон. Модель поверху готова, тому використовуємо Array, щоб створити 7 поверхів ввєрх. На останньому поверсі за допомогою Extrude та зміни площі полігонів створюємо дах.



Рисунок 3.5 – Створена модель одного з будинків

Таким способом створені всі інші будинки. Для заокруглених будинків в основу створювався примітив циліндру і для різноманітності сплющувався за допомогою модифікатора площі по осі Y. Для різноманітності також було створено декілька варіантів дахів і таким чином можна було комбінувати основну частину з поверхами та частину дахами. Це урізноманітнювало кількість моделей, при цьому мінімізувавши час на їх створення.

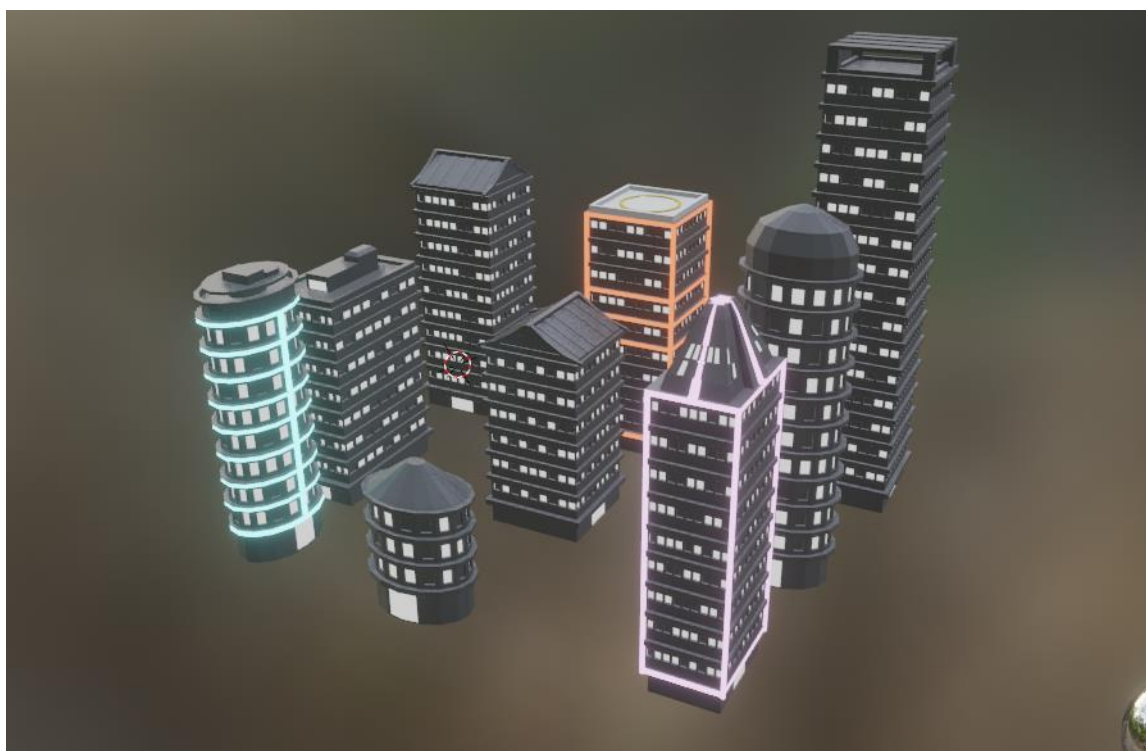


Рисунок 3.6 – Вигляд усіх створених моделей будинків

В цілому, 3D-моделювання дозволяє створювати тривимірні об'єкти на комп'ютері, використовуючи різноманітні інструменти та техніки, що і було вище продемонстровано. Тому за допомогою low poly моделювання було створено різноманітні моделі в гру.

Щоб використовувати створені моделі в двигуні Unity потрібно їх експортувати у форматі .FBX. Це для того щоб створені моделі перенеслися в застосунок разом з матеріалами.

Так як будинки створювалися в одному проєкті, в Unity їх потрібно розділити на окремі префаби.

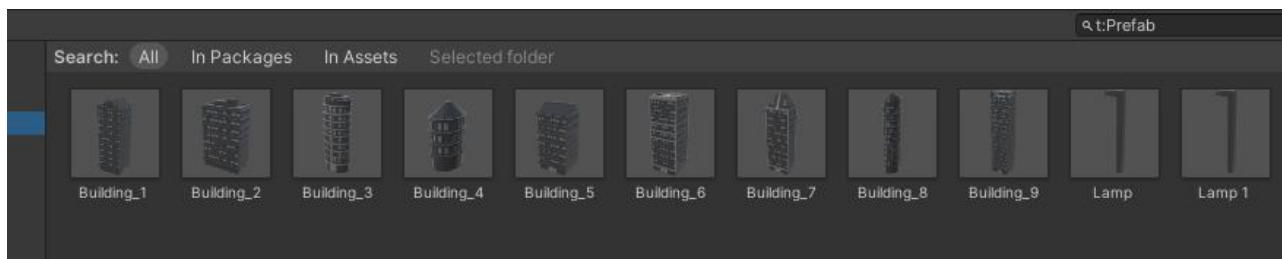


Рисунок 3.7 – Розподіл моделей на префаби

### 3.2 Створення освітлення

Post-processing в Unity – це процес використання післяобробки (post-processing) для покращення візуального вигляду гри або застосунку. Він використовується для застосування різних ефектів до графічного виводу після того, як він пройшов через основний рендеринг [13].

Unity має вбудований пакет післяобробки, який надає набір готових ефектів та інструментів для створення візуальних змін.

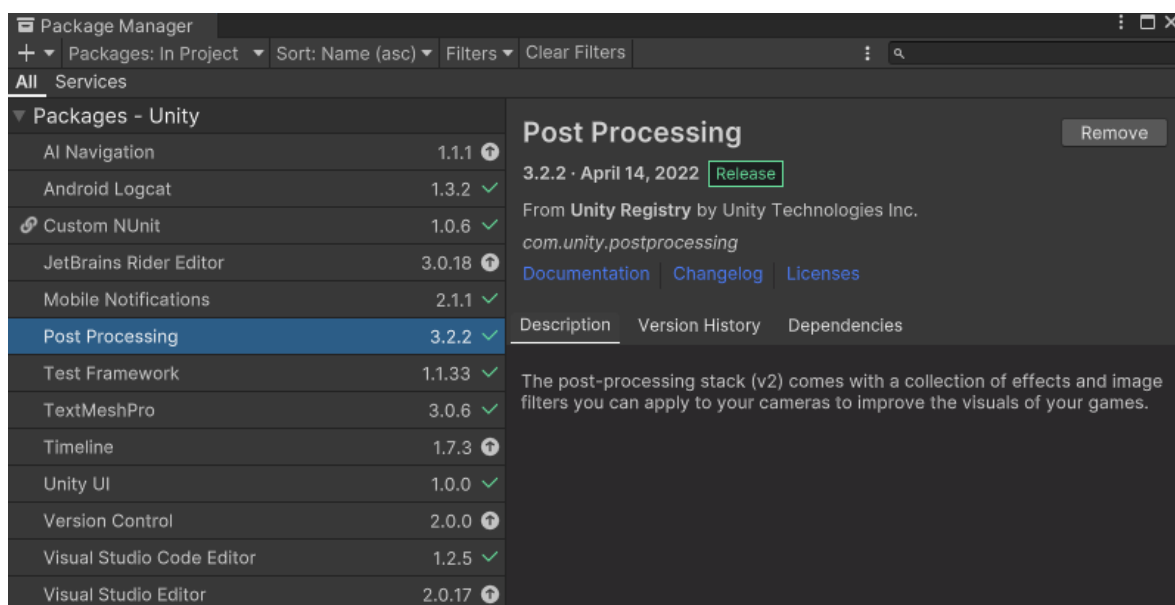


Рисунок 3.8 – Завантаження пакету в Package Manager

Основні можливості post-processing в Unity включають [14]:

- Кольорові фільтри. Змінювати кольори та налаштовувати контрастність, яскравість, насиченість тощо, щоб створити бажаний візуальний ефект.
- Можливості освітлення. Застосовувати ефекти освітлення, такі як bloom (розсіювання світла), глибина різкості (depth of field), мерехтіння (lens flares) та інші, щоб надати сценам більш реалістичний вигляд.
- Ефекти розмиття. Unity має можливості розмиття, такі як мозаїка (mosaic), розмиття руху (motion blur), розмиття глибини (depth blur) і так далі, щоб створювати ефект руху або глибини у вашій грі.
- Корекція кольору. Використовувати інструменти для корекції кольору, щоб змінити баланс кольорів, виправити кольорові артефакти або створити певний настрій в сцені.
- Ефекти частинок та шуму. Unity дозволяє додавати ефекти частинок, такі як дим, вогонь, сніг, а також ефекти шуму, які можуть змінювати візуальний вигляд сцени.

Для створення неонових освітлень потрібно створити профайл післяобробки та додати до нього ефект Bloom. В створеному ефекті ставимо галочки навпроти полів Intensity, Threshold, Soft Knee.

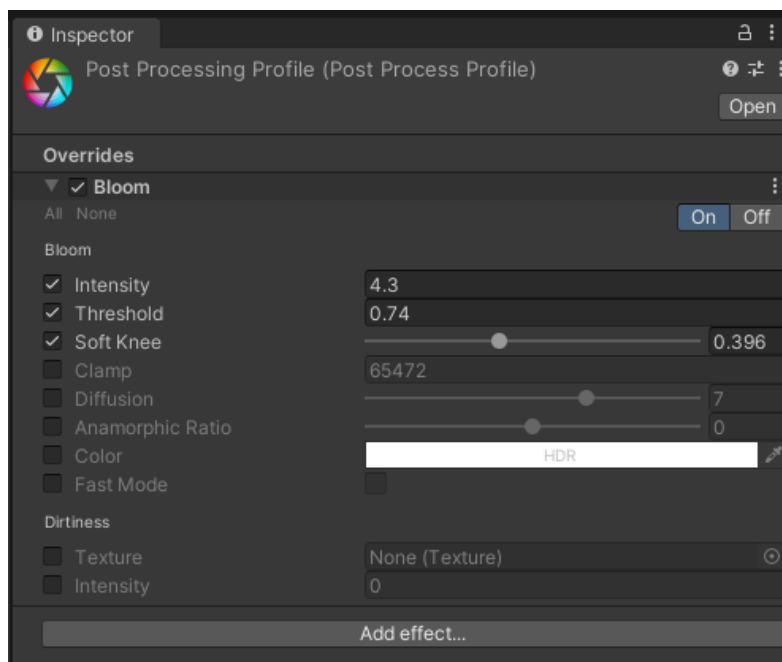


Рисунок 3.9 – Створений профайл зі значеннями

Щоб ефект відображався на камері проекту, потрібно їй задати два основних компонента, які використовуються для реалізації post-processing ефектів: Post Processing Layer (шар післяобробки) та Post Processing Volume (об'єм післяобробки).

Post Processing Layer – це компонент, який додається до камери у сцені. Він відповідає за збір зображення, що рендериться камерою, та застосування до нього ефектів післяобробки. За допомогою Post Processing Layer можна налаштувати порядок, в якому ефекти будуть застосовуватися до зображення, і контролювати, які ефекти використовуються.

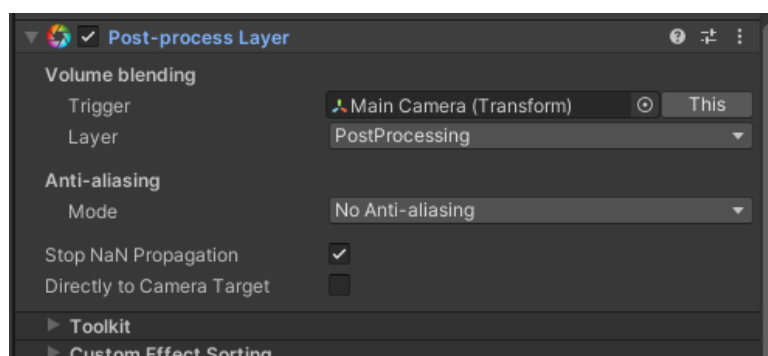


Рисунок 3.10 – Створений компонент Post Processing Layer

Щоб використати цей компонент потрібно також створити окремий шар з післяобробкою.

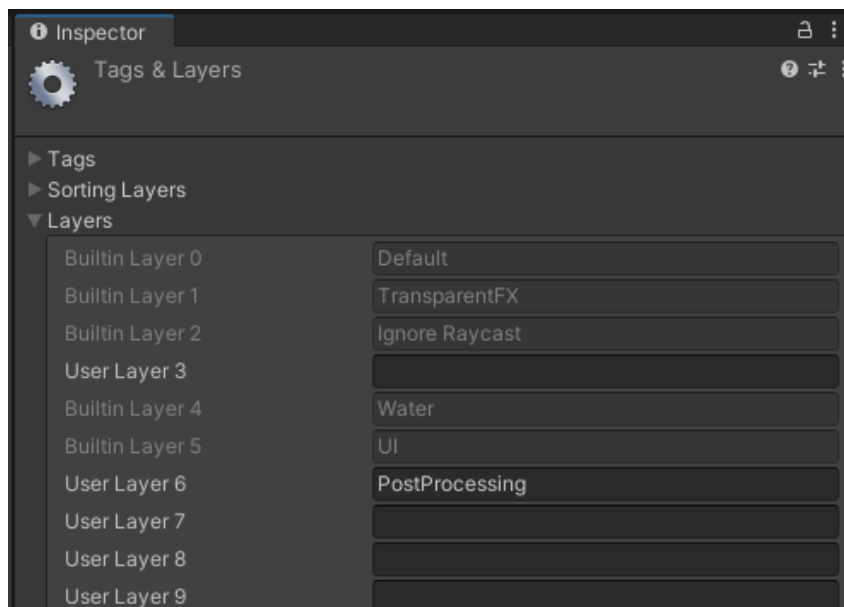


Рисунок 3.11 – Створений новий шар для Post Processing

Post Processing Volume – це об'єкт, який охоплює певний простір у сцені. Він використовується для налаштування параметрів післяобробки для цього об'єму простору. Можна додавати кілька Post Processing Volume в різних частинах сцени і налаштовувати їх параметри незалежно один від одного.

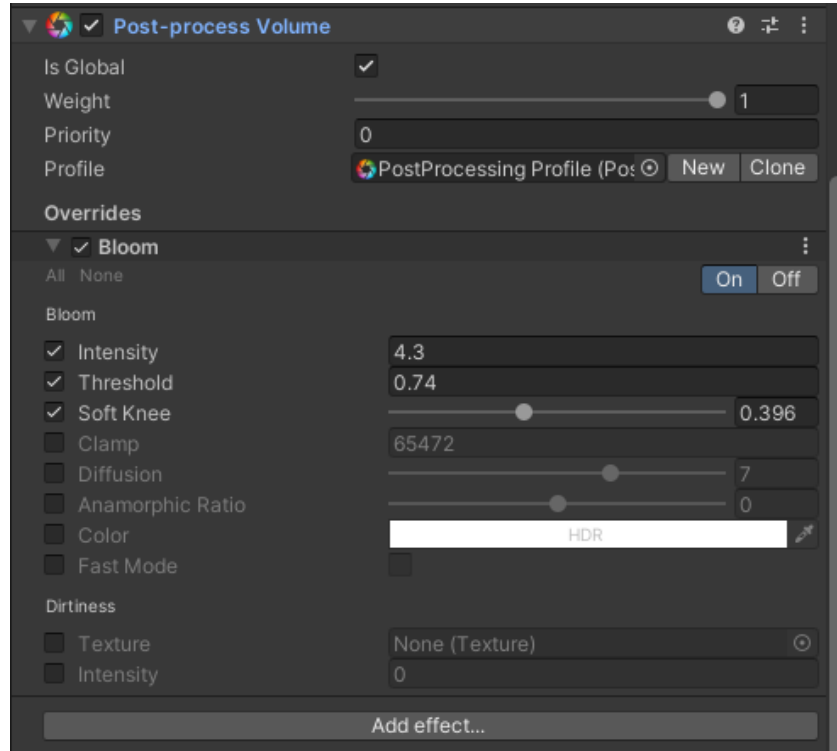


Рисунок 3.12 – Створений компонент Post Processing Volume

Щоб ефекти застосувалися та були видні потрібно на камері встановити виществорений шар.

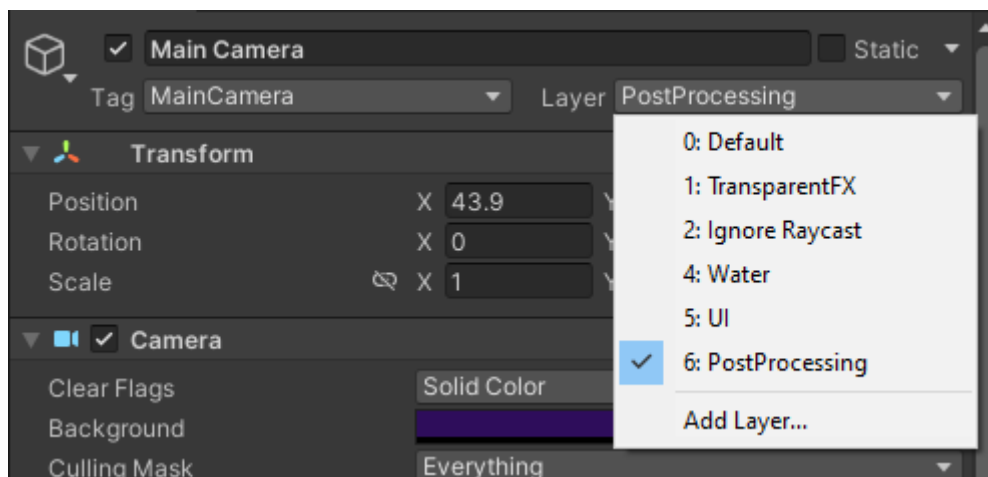


Рисунок 3.13 – Вказання шару для камери



Кафедра інженерії програмного забезпечення  
Мобільна гра казуального жанру з використанням 3D-моделювання



Рисунок 3.14 – Створені матеріали для неонових світла

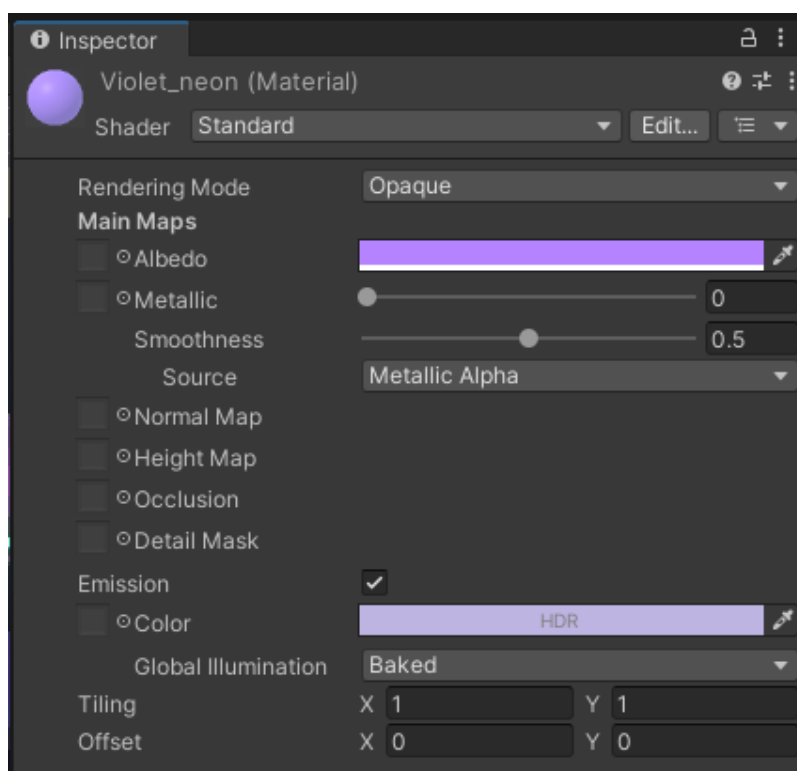


Рисунок 3.15 – Налаштування матеріалів для неонових світла

Використання Post Processing в Unity дозволяє значно покращити візуальний вигляд гри. Це дозволяє додати різноманітні ефекти, такі як кольорові фільтри, освітлення, розмиття та багато інших, щоб зробити сцени більш реалістичними, привабливими та емоційними.

### 3.3 Додавання звуку та музики, його налаштування

Музика в мобільних іграх головоломках відіграє важливу роль під час її використання. Наприклад, спокійна та релаксуюча музика може підкреслити

характер гри та допомогти гравцю зосередитися. Вона додає ритм та темп до геймплею. Музика може створити емоційне занурення гравця у гру, викликати певні почуття, такі як спантеличення, радість або напруження. Також її наявність може диктувати ритм гри та впливати на темп розв'язання головоломок.

До проекту Unity було додано два файли, один це спокійна музика, а другий – це звук натискання на дорогу.

Audio Mixer в Unity - це інструмент, який дозволяє керувати звуковими ефектами, міксами та налаштуваннями аудіо в проекті. Він дає можливість створювати складні мікси звуку, контролювати гучність, створювати ефекти панорамування та додавати різні звукові ефекти [12].

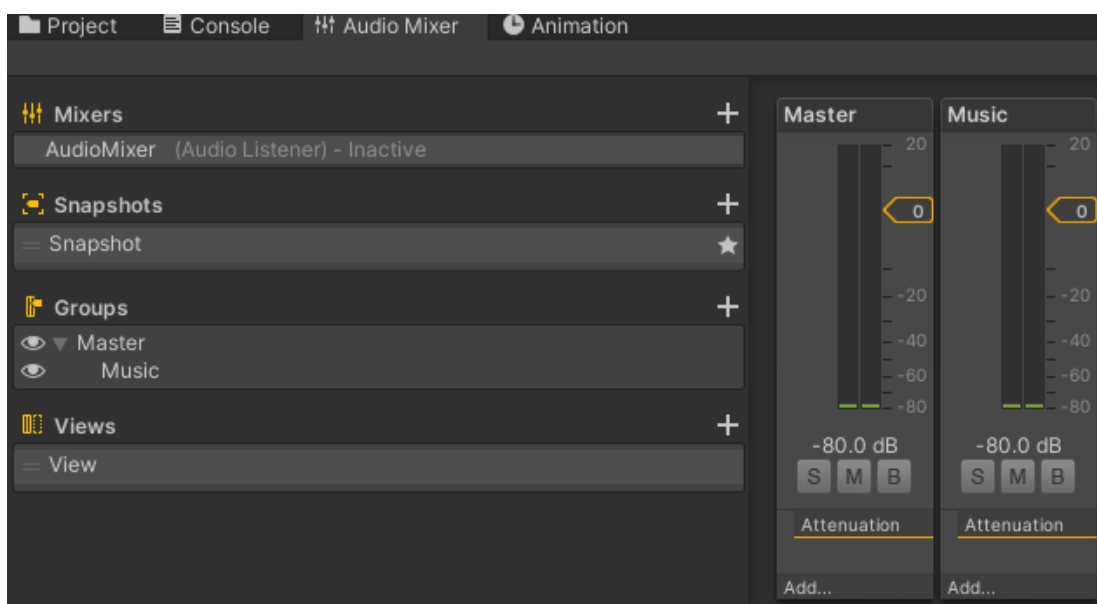


Рисунок 3.16 – Створення аудіоміксеру

Кафедра інженерії програмного забезпечення  
Мобільна гра казуального жанру з використанням 3D-моделювання

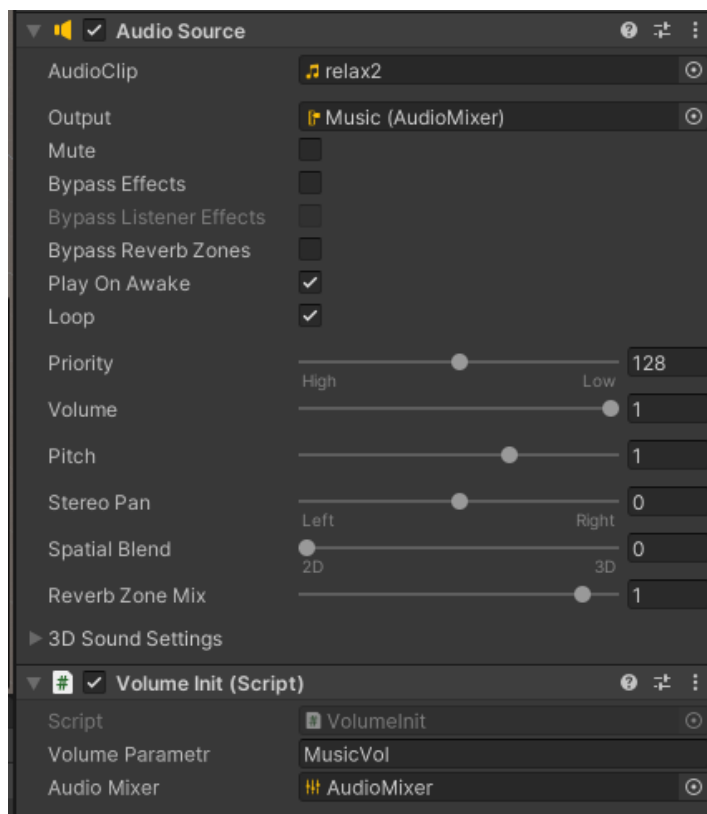


Рисунок 3.17 – Вказання компоненту на джерело музики

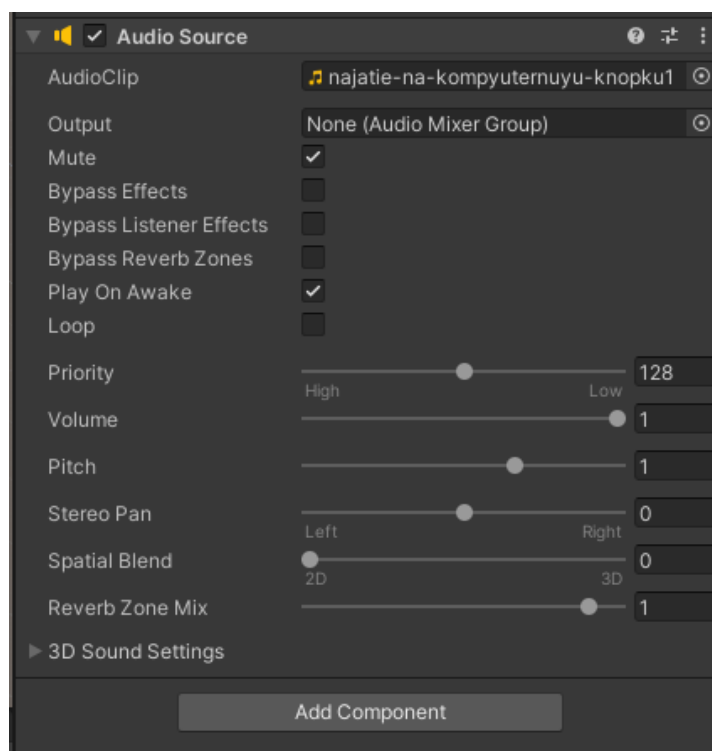


Рисунок 3.18 – Вказання компоненту на джерело звуку

Для контролю звуку за допомогою слайдеру потрібно створити новий компонент VolumeController, в якому потрібно створити методи для вирахування гучності. В методі Start ми отримуємо значення звуку з міксерів та задаємо їх слайдеру, для того щоб при перезапуску гри налаштування звуку не змінювалися. А метод HandleSliderValueChanced отримує значення зі слайдеру та задає його міксеру.

```

Ссылка: 1
private void HandleSliderValueChanged(float value)
{
    _volumeValue=Mathf.Log10(value) * _multiplier;
    audioMixer.SetFloat(volumeParametr, _volumeValue);
}

Сообщение Unity | Ссылка: 0
void Start()
{
    _volumeValue=PlayerPrefs.GetFloat(volumeParametr, Mathf.Log10(slider.value) * _multiplier);
    slider.value = Mathf.Pow(10f, _volumeValue);
}

```

Рисунок 3.19 – Методи для роботи зі звуком

Ініціалізація звуку з міксерів за допомогою коду. Якщо це перше включення і збережень ще не існує, то використовуємо значення по замовченню.

```

public string volumeParametr = "MasterVolume";
public AudioManager audioMixer;
// Start is called before the first frame update
Сообщение Unity | Ссылка: 0
void Start()
{
    var volumeValue = PlayerPrefs.GetFloat(volumeParametr, volumeParametr=="MusicVol"? 0f: -80f);
    audioMixer.SetFloat(volumeParametr, volumeValue);
}

```

Рисунок 3.20 – Методи для роботи зі звуком

Скрипт DontDestroy створений для того, що коли користувач переходив по сценам, звук не зникав. Це виходить через знищення джерел звуку. Таким чином вони зможуть зберігатися.

```
public static DontDestroy instance;
// Start is called before the first frame update
Сообщение Unity | Ссылка: 0
void Start()
{
    if(instance != null)
    {
        Destroy(gameObject);
    }
    else
    {
        instance = this;
        DontDestroyOnLoad(gameObject);
    }
}
```

Рисунок 3.21 – Створення скрипту для збереження джерел звуку

Audio Mixer в Unity є потужним інструментом для керування звуком в іграх. Використання Audio Mixer дозволяє змішувати звукові доріжки, контролювати гучність, створювати ефекти панорамування та додавати звукові ефекти. Він дозволяє створювати різні аудіо-сценарії, забезпечувати плавні переходи між звуками та налаштовувати звукову атмосферу в грі. Завдяки ієрархічній структурі та снапшотам, можна зручно керувати звуковими ефектами та налаштуваннями. Використання Audio Mixer дозволяє створити якісний звуковий дизайн і покращити загальний звуковий досвід гравців.

### 3.4 Створення анімацій в Unity

Створення анімацій є надзвичайно важливим для гри. Вони надають реалізму, виразності та привабливості геймплею. Анімації впливають на геймплей, забезпечуючи плавність рухів та переходів. Також анімації слугують вказівками, допомагаючи гравцям взаємодіяти з гральним світом. В цілому, анімації піднімають гру на новий рівень, роблять її цікавішою, запам'ятовуванішою та незабутньою для гравців.

Анімація в Unity – це процес створення руху та зміни стану об'єктів у грі. Unity надає потужні інструменти для створення та керування анімаціями. Для створення анімацій в цьому проєкті використовувалися Animator Controller та Animation Window [12].

Animator Controller. Це компонент, що дозволяє керувати анімаціями об'єктів. За допомогою нього можна створювати Animator Controllers, які включають різні анімаційні стани та переходи між ними.

В проєкті був використаний Animator для створення переходу анімації під час натискання гравця на одну з доріг. Об'єкт має підстрибнути трохи в гору та опуститися.

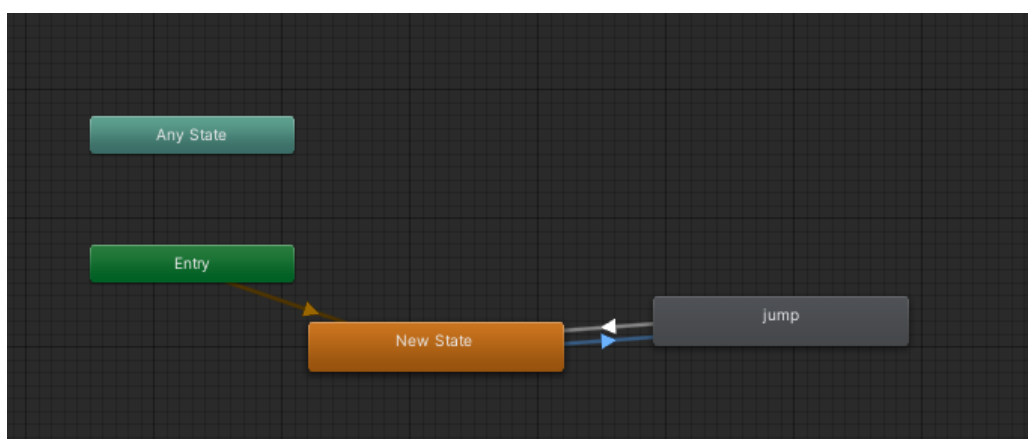


Рисунок 3.22 – Створений аніматор

Animation Window. Це інтерфейс, в якому ви можете створювати та редагувати ключові кадри анімацій. Ви можете налаштовувати позицію, обертання, масштаб та інші параметри об'єктів на різних кадрах.

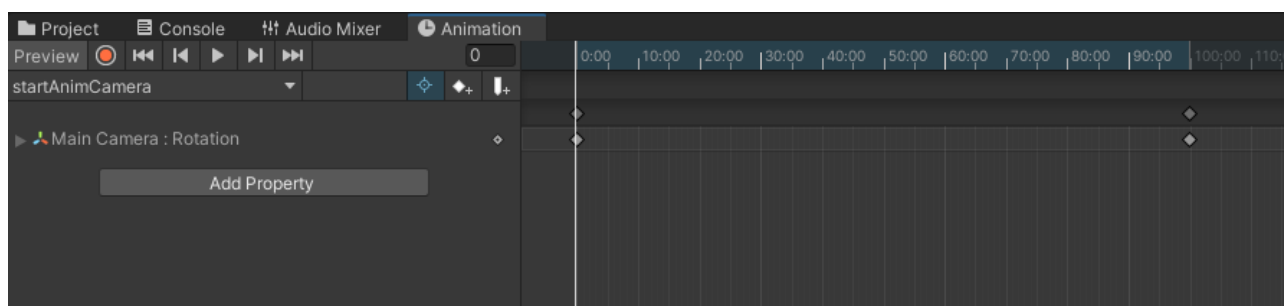


Рисунок 3.23 – Створені ключі анімації для камери

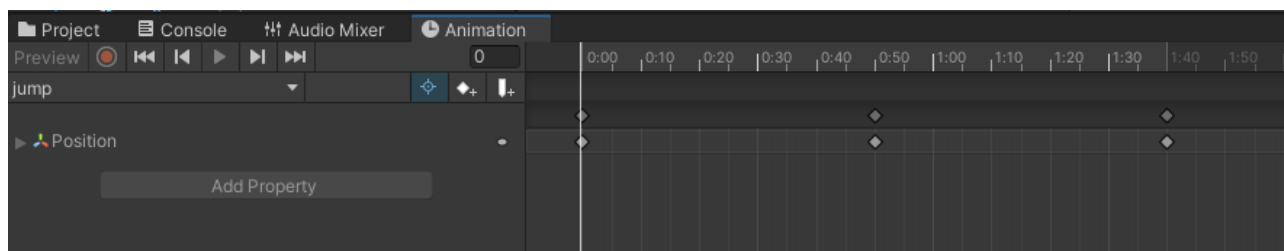


Рисунок 3.24 – Створені ключі анімації для стрибку об'єкта

Створення анімацій є надзвичайно важливим аспектом розробки гри в Unity. Використання Animator Controller та Animation Window дозволяє створювати складні та реалістичні анімації, керувати переходами між ними та визначати параметри анімаційних станів. Використання ключів анімації дозволяє налаштовувати позицію, обертання та інші параметри об'єктів на різних кадрах, створюючи реалістичний рух.

### 3.5 Генерація рівню

При першому запуску гри гравець може розпочати гру тільки з першого рівня. Якщо рівень пройдено, то наступний рівень розблоковано і він може пройти його. Саме для цього і був написаний скрипт LevelManager, який контролює цей процес.

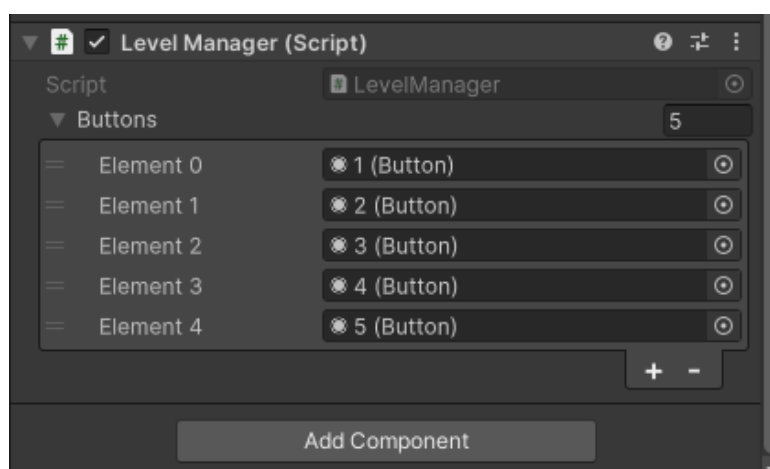


Рисунок 3.25 – Компонент в якому вказується в масиві, яка кнопка відкриває який рівень

```

void Start()
{
    levelUnlock = PlayerPrefs.GetInt("levels", 1);

    for(int i = 0; i < buttons.Length; i++)
    {
        buttons[i].interactable = false;
    }

    for (int i = 0; i < levelUnlock; i++)
    {
        buttons[i].interactable = true;
    }
}

// Update is called once per frame
Ссылка: 0
public void loadLevel(int levelIndex)
{
    SceneManager.LoadScene(levelIndex);
}

```

Рисунок 3.26 – Код завантаження сцени з рівнями

Далі для цієї гри було створено особливу генерацію рівнів. Так як елементи рівня повинні бути статичні, але для кожного ігрового елементу гри генерується випадковим чином кут оберту в 0, 90, 180 та 270 градусів, їх записано в масив.

```

Сообщение Unity | Ссылка: 0
void Start()
{
    totalPipes = PipesHolder.transform.childCount;

    Pipes = new GameObject[totalPipes];

    for(int i = 0; i < Pipes.Length; i++)
    {
        Pipes[i]=PipesHolder.transform.GetChild(i).gameObject;
    }
}

```

Рисунок 3.27 – Масив внесених об'єктів доріг, які будуть згенеровані

```

PossibleRots = correctRotation.Length;
int rand = Random.Range(0, rotations.Length);
transform.eulerAngles=new Vector3(0, rotations[rand], 0);

```

Рисунок 3.28 – Код генерації обертів об'єктів на початку рівня



### 3.6 Розробка логіки гри

Програмна розробка логіки гри є важливим етапом у процесі розробки відеоігор. Це процес створення алгоритмів та правил, які визначають поведінку об'єктів та взаємодію між ними у грі.

Створюється захоплива гра-головоломка, в якій гравець виступає в ролі інженера, який відповідає за побудову системи доріг для забезпечення безперебійного руху транспорту в місті. Головною метою гри це є з'єднання доріг. Гравець повинен з'єднати дорожні елементи таким чином, щоб створити шлях для транспорту.

Кожен елемент дороги має свій контроллер в якому вказується один вірний кут для Т-подібно та L-подібної дороги, та два кути для звичайної прямої дороги.

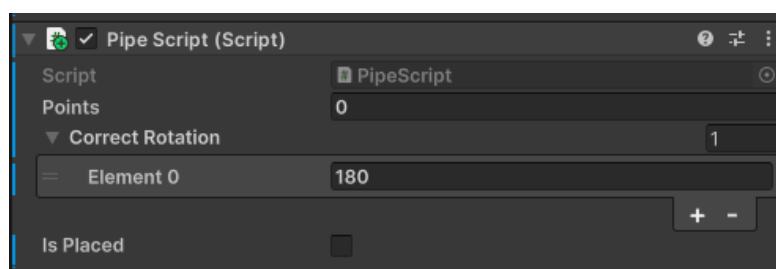


Рисунок 3.29 – Вигляд компоненту для Т-подібно та L-подібної дороги

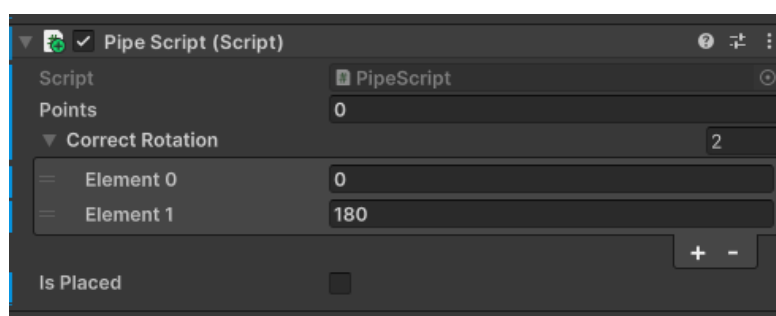


Рисунок 3.30 – Вигляд компонентів для звичайної прямої дороги

Якщо при оберті кут стає на своє місце, то в змінну типу bool записується значення true. Виконується метод правильної позиції, якщо кількість правильних

розстановок дорівнює кількості згенерованих доріг, то рівень вважається пройденим. Також виконується метод, який підраховує кількість ходів.

```
if (PossibleRots > 1)
{
    if (transform.eulerAngles.y == correctRotation[0] || transform.eulerAngles.y == correctRotation[1])
    {
        isPlaced = true;
        gameManager.correctMove();
    }
}
else
{
    if (transform.eulerAngles.y == correctRotation[0])
    {
        isPlaced = true;
        gameManager.correctMove();
    }
}
```

Рисунок 3.31 – Вигляд коду компонента

Щоб відбувалося натискання на об'єкт, то потрібно встановити 3D-коллайдер, який буде трохи вищий за саму фігуру.

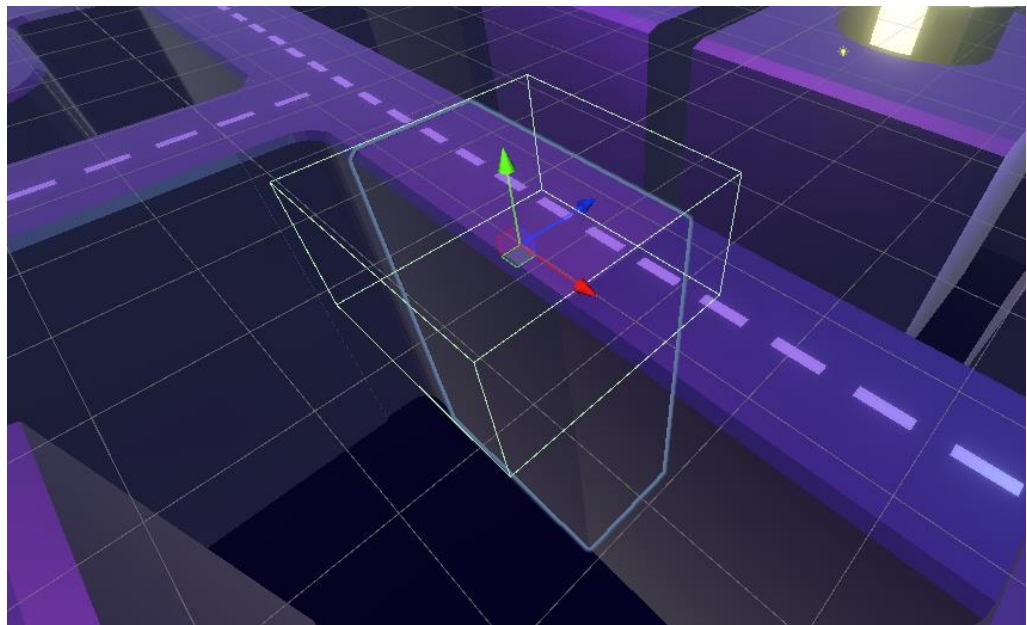


Рисунок 3.32 – Вигляд встановленого коллайдери

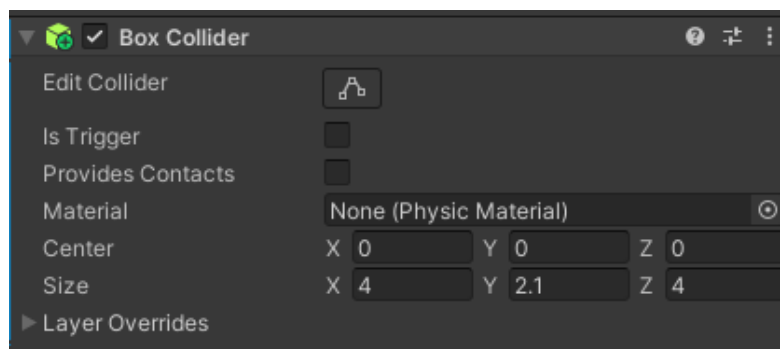


Рисунок 3.33 – Налаштування коллайдери

```
Сообщение Unity | Ссылка: 0
private void OnMouseDown()
{
    gameManager.Move();
    transform.Rotate(new Vector3(0, 90, 0));
    PlayAnim();
}
```

Рисунок 3.34 – Код обертв фігури на її натискання з викликом анімації

```
Ссылка: 4
public void correctMove()
{
    correctedPipes += 1;
    Debug.Log("correct move");
    if (correctedPipes == totalPipes)
    {
        Debug.Log("WIN");
    }
}
```

Рисунок 3.35 – Метод зарахування перемоги

```
Ссылка: 1
public void Move()
{
    totalPoints += 1;
    text.text=totalPoints.ToString();
    result.text=totalPipes.ToString();
}
```

Рисунок 3.36 – Метод, який підраховує та виводить зроблену кількість обертів  
всіх фігур

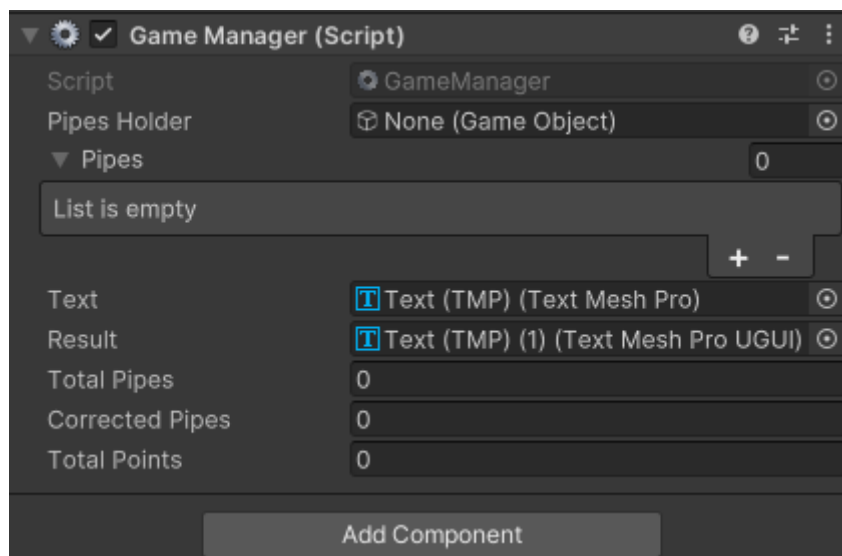


Рисунок 3.37 – Game Manager з вказаними об’єктами для виведення підрахованих очок

```

Ссылка 0
public void restart()
{
    totalPoints = 0;
    text.text = totalPoints.ToString();
    result.text = totalPipes.ToString();
}

```

Рисунок 3.38 – Метод, який перезапускає лічильник ходів при перезапуску рівня

Отже, було запропоновано саме таку реалізацію логіки гри, але її потрібно покращувати, бо чим більша кількість рівнів, тим складніше буде все це контролювати.

### Висновки до розділу 3

У цьому розділі було розглянуто ряд аспектів розробки мобільної гри-головоломки на платформі Unity. Було показано процес створення 3D моделей за допомогою програми ПЗ Blender. Також було продемонстровано створення елементів гри-головоломки, таких як дороги, перехрестя та будинки. Освітлення в грі було створено за допомогою post-processing ефектів в Unity. В розділі було

показано, як додати аудіофайли до проекту та налаштувати їх за допомогою Audio Mixer. Крім того, був створений компонент VolumeController, який дозволяє користувачу контролювати гучність звуку за допомогою слайдера. Також було вирішено проблему з втратою звуку при переході між сценами шляхом створення скрипту DontDestroy. Було представлено використання Animator Controller та Animation Window для створення та керування анімаціями. У даному розділі було розглянуто реалізацію генерації рівнів та розробку логіки гри. Створено скрипт LevelManager, який контролює доступ до рівнів, дозволяючи гравцю прогресувати через рівні, якщо попередній рівень пройдено успішно. Реалізована логіка гри, яка створює захопливий геймплей та вимагає від гравця стратегічного мислення та логічного розуміння для успішного проходження рівнів.

## 4 ТЕСТУВАННЯ ГРИ-ГОЛОВЛОМКИ

Тестування мобільної гри – це процес перевірки якості і функціональності гри на різних мобільних платформах перед її випуском на ринок, але в цьому розділі було проведено перше тестування загалом, перевірка виконаної роботи. Тестування допомагає виявити й усунути помилки, недоліки та неполадки, що можуть вплинути на користувацький досвід і репутацію гри.

Також окрім пошуку помилок в розділі перевіряється, наскільки зручно і привабливо виглядає гра, як гармонійно працюють графіка, звук, управління та інтерфейс. Оцінюється користувацький досвід і зручність гри для гравців. По можливості тестування проводиться на різних пристроях та на різних версіях операційної системи.

Є різні способи тестування застосунків, найголовнішими є:

- Ручне тестування. Це найпоширеніший спосіб тестування мобільних ігор, де тестувальник ручно перевіряє функціональність, геймплей, графіку, звук і всі інші аспекти гри.
- Автоматизоване тестування. З використанням спеціальних інструментів тестування створюються скрипти, які автоматично виконують тестові сценарії. Цей підхід дозволяє швидше виконувати тестування, особливо для повторюваних завдань.
- Тестування користувацького досвіду (UX тестування). Цей вид тестування орієнтований на оцінку вражень користувача від гри. Тестувальники аналізують зручність інтерфейсу, сприйняття графіки, якість звуку, а також загальний користувацький досвід.
- Тестування сумісності. Цей вид тестування перевіряє сумісність гри з різними мобільними платформами, версіями операційних систем, роздільними здатностями екрану тощо.

#### 4.1 Функціональне тестування та огляд інтерфейсу

Тестування було проведено на пристрої Honor 9X 4/128GB з операційною системою Android 10 версії.

Таблиця 4.1 – Тест 1. Завантаження застосунку

<b>№ 1</b> : Завантаження застосунку		
<b>Мета:</b> Перевірити завантаження та відкриття гри		
<b>Пріоритет:</b> High	<b>Час на виконання:</b> 2 хв.	
<b>Дата:</b> 06.06.2022	<b>Власник:</b> Мальцева Владислава	
<b>Передумови:</b>		
<ol style="list-style-type: none"> <li>1. Користувач натиснув на файл для встановлення.</li> <li>2. Смартфон відповідає мінімальним вимогам застосунку.</li> </ol>		
<b>Дії</b>	<b>Очікуваний результат</b>	<b>Наявний результат</b>
Натискання на іконку застосунку.	<ul style="list-style-type: none"> <li>– застосунок успішно встановлений без помилок;</li> <li>– екран старту гри завантажено;</li> <li>– лунає музика.</li> </ul>	Успішно

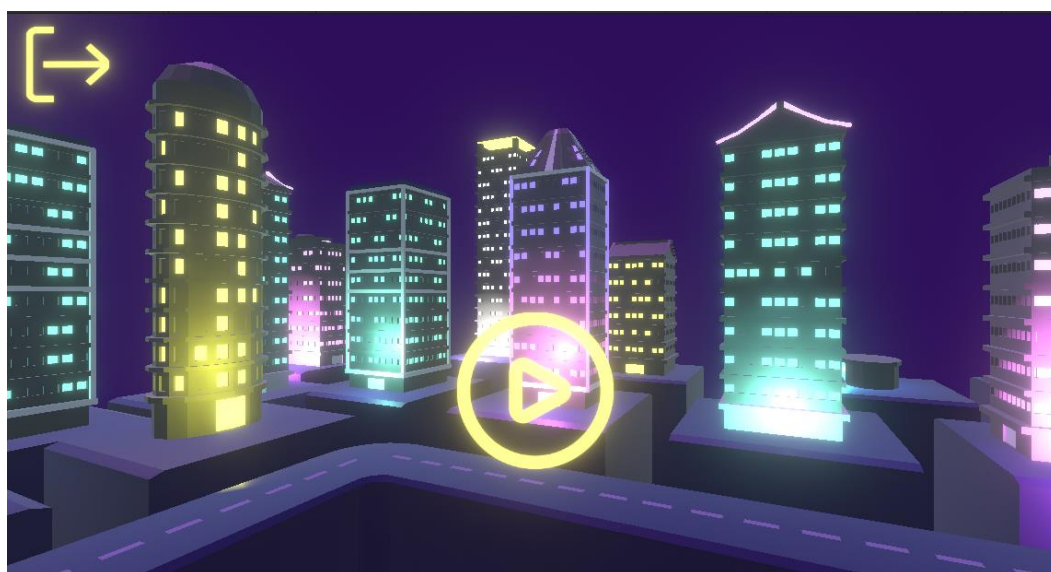


Рисунок 4.1 – Вигляд головного екрану при запуску гри

Таблиця 4.2 – Тест 2. Перехід на мапу рівнів

<b>№ 2</b> : Перехід на мапу рівнів		
<b>Мета:</b> Перевірити перехід на мапу рівнів		
<b>Пріоритет:</b> High	<b>Час на виконання:</b> 1 хв.	
<b>Дата:</b> 06.06.2022	<b>Власник:</b> Мальцева Владислава	
<b>Передумови:</b> Користувач відкрив гру, головний екран було завантажено.		
<b>Дії</b>	<b>Очікуваний результат</b>	<b>Наявний результат</b>
Натискання на кнопку початку гри.	<ul style="list-style-type: none"> <li>– мапа рівнів успішно завантажується</li> <li>– музика продовжує грати</li> <li>– музика не перезапускається</li> </ul>	Успішно



Рисунок 4.2 – Вигляд мапи рівнів



Таблиця 4.3 – Тест 3. Заблоковані рівні

<b>№ 3 : Заблоковані рівні</b>		
<b>Мета:</b> Перевірити заблокованість рівнів		
<b>Пріоритет:</b> High	<b>Час на виконання:</b> 1 хв.	
<b>Дата:</b> 06.06.2022	<b>Власник:</b> Мальцева Владислава	
<b>Передумови:</b>		
<ol style="list-style-type: none"> <li>1. Було відкрито мапу рівнів.</li> <li>2. Гравець грає вперше.</li> </ol>		
<b>Дії</b>	<b>Очікуваний результат</b>	<b>Наявний результат</b>
Натискання на рівні окрім першого	<ul style="list-style-type: none"> <li>– сцени з рівнями не повинні завантажуватися;</li> <li>– кнопки позначаються заблокованими</li> </ul>	Успішно



Рисунок 4.3 – Перевірка заблокованих рівнів

Таблиця 4.4 – Тест 4. Перехід на перший рівень

<b>№ 4 : Перехід на перший рівень</b>		
<b>Мета:</b> Перевірити завантаження пешого рівня		
<b>Пріоритет:</b> High	<b>Час на виконання:</b> 1 хв.	
<b>Дата:</b> 06.06.2022	<b>Власник:</b> Мальцева Владислава	
<b>Передумови:</b> Було обрано рівень для проходження.		
<b>Дії</b>	<b>Очікуваний результат</b>	<b>Наявний результат</b>
Було натиснуто на кнопку з першим рівнем	<ul style="list-style-type: none"> <li>– завантаження сцени з першим рівнем;</li> <li>– вдала генерація;</li> <li>– лічильник має бути зі значенням 0.</li> </ul>	Успішно



Рисунок 4.4 – Вигляд завантаженого рівню

Таблиця 4.5 – Тест 5. Проходження рівня

<b>№ 5 : Проходження рівня</b>		
<b>Мета:</b> Перевірити проходження рівня		
<b>Пріоритет:</b> High	<b>Час на виконання:</b> 1 хв.	
<b>Дата:</b> 06.06.2022	<b>Власник:</b> Мальцева Владислава	
<b>Передумови:</b> Було завантажено рівень.		
<b>Дії</b>	<b>Очікуваний результат</b>	<b>Наявний результат</b>
Було натиснуто на дороги для обертю	<ul style="list-style-type: none"> <li>– фігури обертаються;</li> <li>– лічильник з кожним обертю збільшується на 1;</li> <li>– вдала анімація кожного об'єкту.</li> </ul>	Успішно

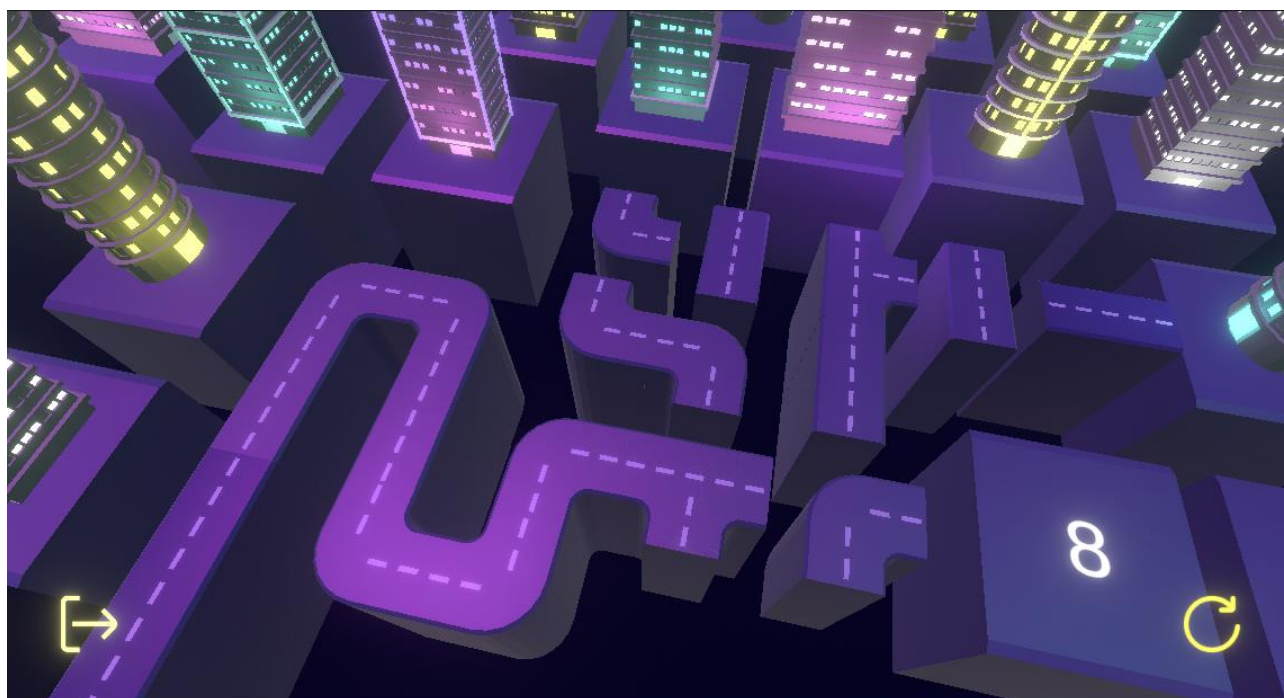


Рисунок 4.5 – Вигляд завантаженого рівню з демонстрацією роботи лічильником

Таблиця 4.6 – Тест 6. Зарахування проходження рівня

<b>№ 6 : Зарахування проходження рівня</b>		
<b>Мета:</b> Перевірити зарахування проходження рівня		
<b>Пріоритет:</b> High	<b>Час на виконання:</b> 1 хв.	
<b>Дата:</b> 06.06.2022	<b>Власник:</b> Мальцева Владислава	
<b>Передумови:</b>		
<ol style="list-style-type: none"> <li>1. Було завантажено рівень.</li> <li>2. Користувач вирішував головоломку.</li> </ol>		
<b>Дії</b>	<b>Очікуваний результат</b>	<b>Наявний результат</b>
Всі фігури були правильно повернуті	Буде показано вікно з результатом, який було нараховано на сцені з рівнем.	Успішно



Рисунок 4.6 – Вигляд вікна з результатом



Таблиця 4.7 – Тест 7. Зарахування проходження рівня

<b>№ 7 : Перепроходження рівня</b>		
<b>Мета:</b> Перевірити перепроходження рівня		
<b>Пріоритет:</b> High	<b>Час на виконання:</b> 1 хв.	
<b>Дата:</b> 06.06.2022	<b>Власник:</b> Мальцева Владислава	
<b>Передумови:</b> Було пройдено рівень.		
<b>Дії</b>	<b>Очікуваний результат</b>	<b>Наявний результат</b>
Було натиснуто на кнопку «Перепройти»	<ul style="list-style-type: none"> <li>– рівень буде регенеровано;</li> <li>– лічильник знову стане зі значенням 0.</li> </ul>	Успішно

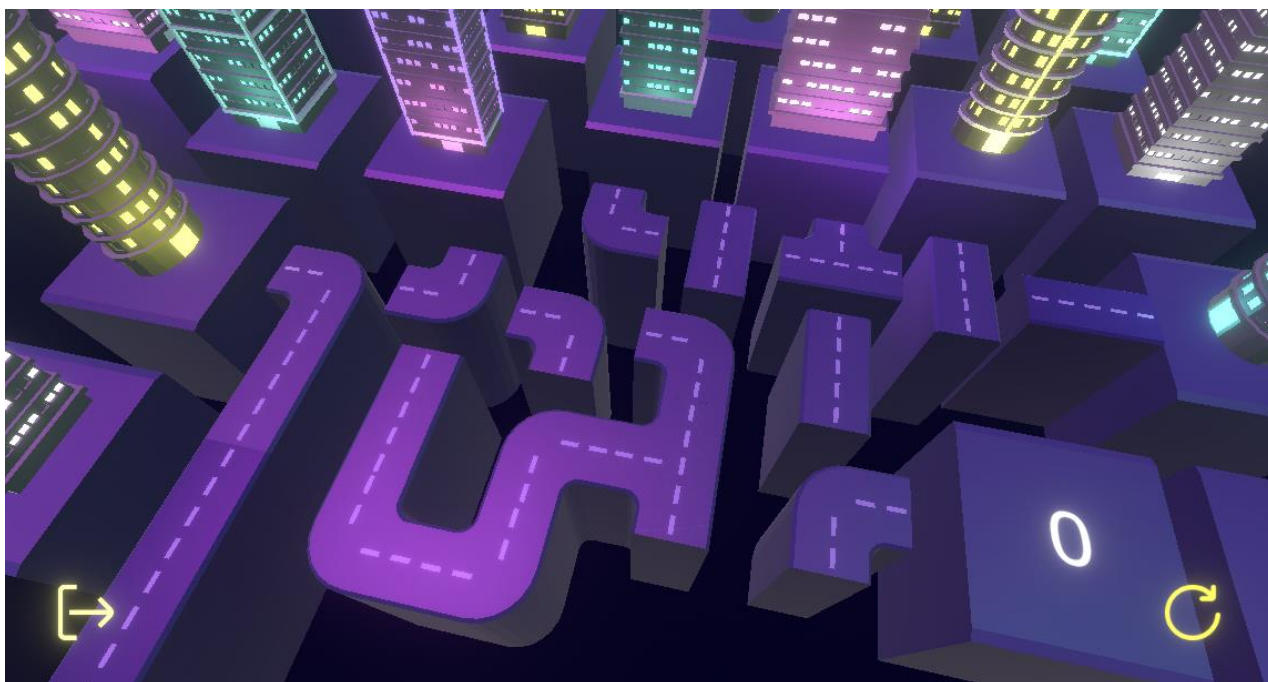


Рисунок 4.7 – Вигляд вікна з результатом

Таблиця 4.8 – Тест 8. Відкриття та закриття налаштувань

<b>№ 8 : Відкриття та закриття налаштувань</b>		
<b>Мета:</b> Перевірити відкриття та закриття налаштувань		
<b>Пріоритет:</b> High	<b>Час на виконання:</b> 1 хв.	
<b>Дата:</b> 06.06.2022	<b>Власник:</b> Мальцева Владислава	
<b>Передумови:</b> Було перейдено на мапу рівнів.		
<b>Дії</b>	<b>Очікуваний результат</b>	<b>Наявний результат</b>
Було натиснуто на кнопку «Налаштування»	– вікно налаштувань буде завантажено; – вікно буде вдало закрито.	Успішно

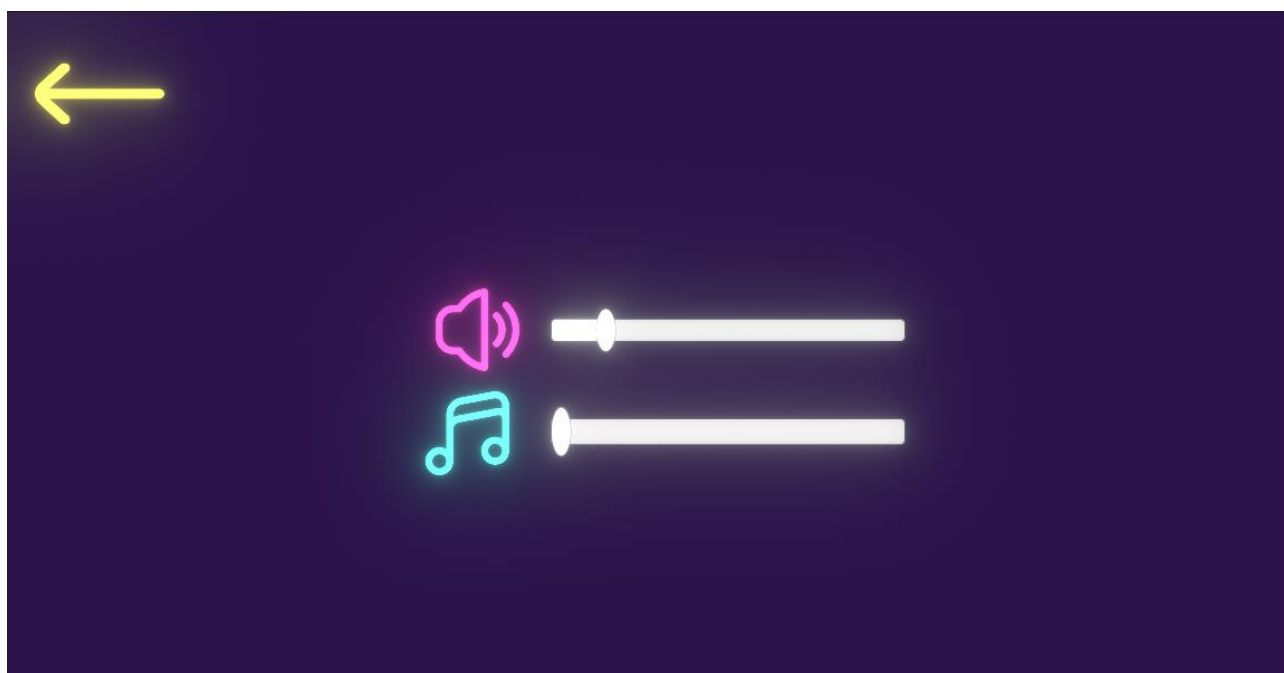


Рисунок 4.7 – Вигляд вікна налаштувань

Таблиця 4.9 – Тест 9. Налаштування музики

<b>№ 8</b> : Налаштування музики		
<b>Мета:</b> Перевірити налаштування музики		
<b>Пріоритет:</b> High	<b>Час на виконання:</b> 1 хв.	
<b>Дата:</b> 06.06.2022	<b>Власник:</b> Мальцева Владислава	
<b>Передумови:</b> Було перейдено в вікно налаштувань.		
<b>Дії</b>	<b>Очікуваний результат</b>	<b>Наявний результат</b>
Було використано слайдери	<ul style="list-style-type: none"> <li>– гучність музики змінюється пропорційно слайдеру;</li> <li>– гучність звуку змінюється пропорційно слайдеру;</li> <li>– після закриття вікна налаштування повинні зберегтися.</li> </ul>	Успішно

#### 4.2 Подальший розвиток гри

Після здачі проекту ця гра продовжить існувати, але її потрібно покращувати. Навіщо? В світі ігор конкуренція дуже висока, тому покращення гри дозволяє зберегти або зайняти своє місце на ринку. Якщо гра пропонує незадовільний досвід, має помилки або недоліки, гравці швидко втратять інтерес і перейдуть до інших альтернатив. Покращення гри дозволяє забезпечити більш насичений, захоплюючий та задовільний геймплей. Покращення гри може сприяти збільшенню активності користувачів. У випадку платних або фрі-тумплей ігор, покращення гри може позитивно вплинути на монетизацію.

Ось такі пропозиції було сформовано під час тестування та користування:

- змінити швидкість обертання фігур (занадто швидко);
- додати більше рівнів;
- урізноманітнити рівні з їх проходженням;

- додати більше анімацій (оживити місто, додати авто на дороги при складенні шляху);
- змінити логіку перемоги на більш автоматизовану. Використовувати алгоритм Дейкстри, для знаходження шляхів рішення рівнів;
- додати мінімальну монетизацію;
- додати нові режими гри;
- додати мультиплеєрний аспект, проходити рівні співпрацюючи або змагаючися;
- додати події в грі, гра стане більш живою та динамічною;

#### **Висновки до розділу 4**

У розділі було проведено перше тестування мобільної гри, яке має на меті перевірити якість та функціональність гри перед її випуском на ринок. Тестування допомагає виявити та виправити помилки, недоліки та неполадки, які можуть вплинути на користувацький досвід та репутацію гри.

У даному розділі були використані різні методи тестування, зокрема ручне тестування, тестування користувацького досвіду та тестування сумісності. Кожен метод має свої переваги та дозволяє виявити різні аспекти якості гри.

Результати тестування показали, що гра успішно пройшла перевірку на функціональність та огляд інтерфейсу. Всі очікувані результати були досягнуті, анімація та звук працюють належним чином. Також було перевірено різні функції гри, такі як перехід на мапу рівнів, заблоковані рівні, перехід на конкретний рівень, обертання фігур, зарахування проходження рівня та налаштування гри. У всіх випадках результати були відповідними очікуванням.

Згідно з отриманими пропозиціями під час тестування та користування, запропоновано подальший розвиток гри. Він включає зміну швидкості обертання фігур, додавання більше рівнів, урізноманітнення рівнів, додавання більше анімацій, зміну логіки гри та покращення графіки. Ці пропозиції спрямовані на поліпшення геймплею та збільшення активності користувачі.



## ВИСНОВКИ

В ході виконання кваліфікаційній роботі бакалавра були розглянуті різні аспекти розвитку логічного мислення, уваги, стратегічних навичок за рахунок розробки казуальної гри головоломки з використанням тривимірного моделювання.

Для досягнення визначеної мети вирішено поставлені завдання:

- досліджено категорії ігор;
- був зроблений аналіз ігрового ринку;
- сформовано головні вимоги майбутнього застосунку;
- створено блок-схеми алгоритму роботи гри та необхідні UML-діаграми;
- були створені необхідні 3д моделі;
- створено анімації;
- створено елементи дизайну інтерфейсу;
- реалізовано програмну частину застосунку;
- проведено тестування.

Було проаналізовано ігровий ринок застосунків, розглянуто жанри ігор та проведено аналіз аналогів ігор-головоломок казуального жанру. Було виявлено переваги та недоліки програм для моделювання 3D об'єктів і сформовано функціонал та особливості створюваного застосунку.

Також було продемонстровано використання діаграм взаємодії, стану та діяльності для моделювання процесів та систем гри. Ці діаграми допомогли у зрозумінні роботи системи та виявленні проблем на ранніх етапах розробки.

Було показано процес створення 3D моделей за допомогою програми Blender, створення елементів гри, освітлення та налаштування звуків. Також було розглянуто реалізацію генерації рівнів та логіки гри. Всі ці етапи допомогли створити захоплюючий геймплей та вимагають від гравця стратегічного мислення та логічного розуміння для успішного проходження рівнів.

Також було проведено тестування мобільної гри для перевірки якості та функціональності. Різні методи тестування, включаючи ручне тестування, тестування користувацького досвіду та тестування сумісності, були використані для виявлення помилок і неполадок.

Результати тестування показали, що гра успішно пройшла перевірку на функціональність та відповідає очікуванням. Пропозиції тестування та користувачів спрямовані на подальший розвиток гри, такі як зміна швидкості обертання фігур, додавання більше рівнів, урізноманітнення геймплею та поліпшення графіки.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. 187 Mobile Gaming Statistics for 2023: The Ultimate List | Udonis. Udonis Mobile Marketing Agency. URL: <https://www.blog.udonis.co/mobile-marketing/mobile-games/mobile-gaming-statistics> (date of access: 04.05.2023).
2. 56+ Mobile Gaming Statistics (2023): Revenue, Market Share, Demographics. HeadphonesAddict. URL: <https://headphonesaddict.com/mobile-gaming-statistics/> (date of access: 04.05.2023).
3. Mobiloids. Water Pipes – Apps on Google Play. Android Apps on GooglePlay. URL: [https://play.google.com/store/apps/details?id=com.mobiloids.waterpipespuzzle&hl=en\\_US&gl=US](https://play.google.com/store/apps/details?id=com.mobiloids.waterpipespuzzle&hl=en_US&gl=US) (date of access: 25.01.2023).
4. App2Eleven. Pipe – logic puzzles – Apps on Google Play. Android Apps on Google Play. URL: [https://play.google.com/store/apps/details?id=air.App2Eleven.LogicGamePuzzlesPipe&hl=en\\_US&gl=US](https://play.google.com/store/apps/details?id=air.App2Eleven.LogicGamePuzzlesPipe&hl=en_US&gl=US) (date of access: 25.01.2023).
5. Dingo D. Aquavias – Water Flow Puzzle – Apps on Google Play. Android Apps on Google Play. URL: [https://play.google.com/store/apps/details?id=ua.krou.aqueducts&hl=en\\_US&gl=US](https://play.google.com/store/apps/details?id=ua.krou.aqueducts&hl=en_US&gl=US) (date of access: 25.01.2023).
6. Grady Booch, James Rumbaugh, Ivan Jacobson. The Unified Modeling Language User Guide. Publisher: Addison Wesley, 2007, 512 p.
7. Мова UML. Діаграма використання. Все для навчання. URL: <http://p4ilka.blogspot.com/2018/12/uml.html> (дата звернення: 06.04.2023).
8. What is Component Diagram?. Ideal Modeling & Diagramming Tool for Agile Team Collaboration. URL: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-component-diagram/> (date of access: 06.04.2023).

9. UML – Class Diagram. Online Tutorials Library. URL: [https://www.tutorialspoint.com/uml/uml\\_class\\_diagram.htm](https://www.tutorialspoint.com/uml/uml_class_diagram.htm) (date of access: 06.04.2023).
10. UML diagrams guide. Nulab. URL: <https://nulab.com/learn/software-development/uml-diagrams-guide/> (date of access: 18.02.2023).
11. Gordon Fisher. Blender 3D Basics Beginner's Guide. Second Edition. Publisher: Packt Publishing , 2014, 526 p.
12. J. Sharp. Microsoft Visual C#. Detailed guide. Publisher: Peter Language, 2017, 817 p.
13. Jeremy Gibson Bond. Unity and C#. Gamedev from idea to implementation. Pitter Press 2020, 928 p.
14. Lapin Andrey. Unity 3D tutorial. Internet publishing house, 2013, 77 p.
15. Goldstone Will. The Basics of Making Games in Unity Packt Publishing, 2019, 266 p.
16. Cem Kaner, Jack Falk, Hung Quoc Nguyen. Testing Computer Software. Second Edition. Publisher: International Thomson Publishing Press, 2021, p 538.