

Міністерство освіти і науки України  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»  
Кафедра інформатики факультету інформатики

## РОЗРОБКА ІНТЕЛЕКТУАЛЬНОЇ ГРИ З ВИКОРИСТАННЯМ ІГРОВОГО РУШІЯ UNITY

Текстова частина до курсової роботи  
за спеціальністю «Інженерія програмного забезпечення» - 121

Керівник курсової роботи,

к.т.н., доцент

Жежерун О.П.

*(прізвище та ініціали)*

*(підпис)*

“ \_\_\_\_\_ ” \_\_\_\_\_ 2021 р.

Виконав студент \_\_\_\_\_

Коломєйцев Д.Б.

*(прізвище та ініціали)*

“ \_\_\_\_\_ ” \_\_\_\_\_ 2021 р.

Київ 2021

Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»  
Кафедра інформатики факультету інформатики

ЗАТВЕРДЖУЮ  
Зав.кафедри інформатики,  
к.т.н., доцент  
Жежерун О.П.  
(прізвище та ініціали)  
  
\_\_\_\_\_  
(підпис)  
“ \_\_\_\_ ” \_\_\_\_\_ 2021 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ  
на курсову роботу

студенту Коломейцеву Денису Борисовичу факультету інформатики 4-го курсу

ТЕМА Розробка інтелектуальної гри з використанням ігрового рушія Unity

Зміст ТЧ до курсової роботи:

- Індивідуальне завдання
- Календарний план виконання роботи
- Анотації
- Вступ
- 1 Постановка задачі
- 2 Теоретичні відомості
- 3 Розробка застосунку
- 4 Опис додатку
- Висновки
- Список літератури

Дата видачі “ \_\_\_\_ ” \_\_\_\_\_ 2021 р. Жежерун О.П. \_\_\_\_\_  
(підпис)

Завдання отримав \_\_\_\_\_  
(підпис)

**Тема: Розробка інтелектуальної гри з використанням ігрового рушія Unity**  
**Календарний план виконання роботи:**

№	Назва етапу	Термін виконання	Примітка
1.	Отримання теми курсової роботи	19.10.2020	
2.	Пошук тематичної літератури	17.11.2020	
3.	Ознайомлення з літературою	20.12.2020	
4.	Вивчення аналогів	19.01.2021	
5.	Планування проекту	25.01.2021	
6.	Вивчення можливостей потенційний рушіїв	26.01.2021	
7.	Ознайомлення з рушієм Unity	08.02.2021	
8.	Реалізація інтерфейсу	18.02.2021	
9.	Написання скриптів і налаштування компонентів	27.02.2021	
10.	Написання текстової частини	19.03.2021	
11.	Перегляд змісту роботи керівником	06.04.2021	
12.	Внесення змін до курсової відповідно до зауважень наукового керівника	08.04.2021	
13.	Створення презентації	12.04.2021	

Студент Коломєйцев Д.Б.

Керівник Жежерун О.П.

“            ”

\_\_\_\_\_

## ЗМІСТ

Анотація.....	5
Перелік умовних позначень.....	6
ВСТУП.....	7-8
Завдання роботи.....	9
РОЗДІЛ 1 ОГЛЯД ІГРОВОГО РУШІЯ .....	10-14
1.1 Загальна характеристика .....	10-12
1.2 Головні переваги.....	13
1.3 Основні недоліки.....	14
РОЗДІЛ 2 РОЗРОБКА ПРОЕКТУ.....	15-27
2.1 Планування розробки.....	15
2.2 Реалізація.....	15-16
2.3 Опис GUI та меню.....	16-18
2.4 Початок гри.....	18-20
2.5 Ціль гри.....	21
2.6 Призначення скриптів .....	22-27
2.6.1 Класи вікторини .....	22-23
2.6.1.1 Квест.....	22
2.6.1.2 Додаткові класи .....	23
2.6.2 Загальні класи .....	24
2.6.3 Скрипти налаштування .....	24-27
РОЗДІЛ 3 НАЛАШТУВАННЯ ЕЛЕМЕНТІВ РУШІЯ.....	28-31
ВИСНОВКИ.....	32
Список літератури.....	33

## **Анотація**

Під час виконання даної роботи було розроблено 2D гру в жанрі квесту на основі ігрового рушія Unity. Також, в процесі підготовки, було розглянуто особливості рушія і процесу розробки на ньому. Крім того, описуються компоненти, які були використані для реалізації проекту. Скрипти написані для гри були написані на мові програмування C#.

## **Використані скорочення**

ПК – персональний комп'ютер

C# – об'єктно-орієнтована мова програмування для платформи .NET від компанії Microsoft.

UnityScript – скриптова мова, розроблена на основі JavaScript, має його синтаксис та створена спеціально для ігрового рушія Unity.

Спрайт – графічний об'єкт в комп'ютерній графіці. Частіше за все є растровим зображення яке відображається на екрані (фон(небо), літаки, набої).

2D – (від англ. 2-dimensional) двовимірна розмірність простору.

3D – (від англ. 3-dimensional) тривимірна розмірність простору.

GUI – (від англ. graphical user interface) – графічний інтерфейс користувача.

## ВСТУП

Unity Engine - кросплатформений інструмент для розробки комп'ютерних ігор і застосунків. На сьогодні є одним з найбільш популярних в сфері геймдеву, як для невеликих компаній, так і для інді-розробників, що зумовлено простотою використання і освоєння, безкоштовною моделлю розповсюдження і можливістю зручної монетизації продукту. Має можливість розробки як двомірних, так і 3D додатків на широкий спектр платформ – Windows, PlayStation, Xbox, Android, iOS.

Останні кілька років Unity є одним з основних виборів для розробників на ринку мобільних ігор. За оцінками більше ніж 50% ринку представлені додатками саме на цьому рушії[1]. Але при цьому залишається популярним на ПК і MacOS у студій які віддають перевагу використанню готових рушіїв від сторонніх розробників, а не створенню власних, чи то заради економії коштів і часу, чи то через зручність для створення більш простих ігор. Прикладом перших можна назвати такі студії як китайська miHoYo Limited, яка в 2020 році випустила свою MMORPG «Genshin Impact» одразу на Windows, iOS, Android і PS4, а трохи згодом і на PS5, використавши по максимуму мультиплатформеність рушія і створивши незвичну для сучасного геймдеву кросплатформеність стільникових пристроїв з консолями і ПК, при чому без серйозної втрати в якості графічної складової. В тому числі завдяки цьому їх старт став рекордним за популярністю для ігор китайського виробництва, за кількістю глядачів, перевершивши в той час гегемона платформи Twitch – «Fortnite»[2]. В свою чергу, прикладом другого випадку можна назвати гру «Gwent» від польської студії CD Projekt RED, відомої своєю серією «The Witcher», фактично спін-офом якої і є колекційна карточна гра Гвінт. Не зважаючи на те що він був реалізований як міні-гра на власному рушії CDPR у «The Witcher 3: Wild Hunt», під час розробки окремої версії польська студія вирішила використати переваги Unity. Так, гра

початково вийшла тільки на Windows, але з часом розрослася і на інші плаформи і станом на сьогодні присутня також на PS4, Xbox One, iOS і Android.

У зв'язку з цим, освоєння Unity і використання всіх можливостей цього рушія для розробки різнопланових кросплатформених ігор є дуже актуальним.

Метою цієї роботи є вивчення можливостей рушія для створення 2D розвиваючого квесту в вигляді вікторини для дітей що хотіли б почати вивчення англійської мови.



Завданнями роботи є:

1. Огляд ігрового рушія Unity
2. Дослідження можливостей Unity у створенні ігор з 2D графікою
3. Використання стандартних компонентів Unity для створення зручного 2D інтерфейсу.
4. Розробка прототипу 2D квесту в вигляді вікторини з можливістю безлімітного розширення списку запитань.

## РОЗДІЛ 1 ОГЛЯД ІГРОВОГО РУШІЯ

### 1.1 Загальна характеристика

Unity Engine – кросплатформений інструмент для розробки ігор для широкого спектру платформ. Станом на 2021 рік, програма-редактор працює на Windows, Linux і MacOS, а сам рушій може запускатися на 25-ти платформах, серед яких, окрім зазначених вище, варто виділити різноматітні менш популярні консолі, VR-застосунки, інтернет додатки і навіть смарт-телевізори[3].

Дебют Unity відбувся в 2005 році коли він був представлений на конференції розробників софту на платформах Apple - Worldwide Developers Conference. Початково рушій був розрахований винятково для MacOS, проте вже за кілька місяців після презентації вийшло оновлення, що дозволяло працювати з ним на платформі Windows, після чого подібні розширення стали регулярними. На даний момент, останньою версією Unity є 2021.1.2f1, що була випущена 23 березня 2021 року. В основному це оновлення виправляє існуючі баги поточної версії, підвищує стабільність роботи і знижує кількість вилетів.[4] Не зважаючи на це, для розробки в процесі виконання проекту, було використано значно більш стабільну версію 2020.3.3f1, яку рекомендують за замовчуванням при завантаженні актуальної версії із застосунку Unity Hub(рисунок 1.1).

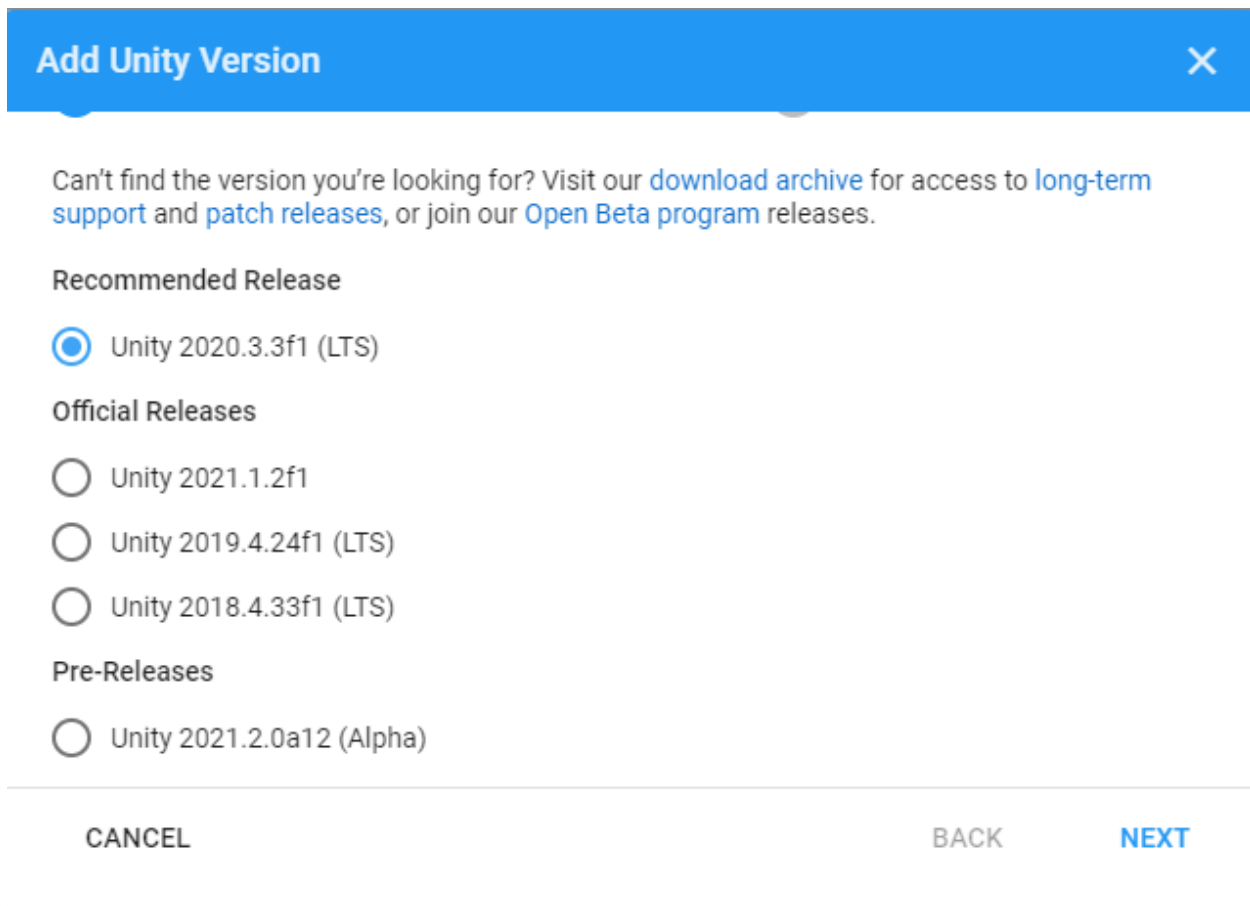


Рис. 1.1 Завантаження клієнту в Unity Hub

Всі вищезазначені причини дозволяють Unity доволі популярним вибором для різноманітних розробників, незалежно від жанру чи бюджету додатку. Для прикладу, у випадковій вибірці з тисячі безкоштовних мобільних ігор, за відсотком використання Unity поступається лише сумарній кількості всіх рушіїв власного виробництва студій (41% проти 34%). Досягнення цього показника легко відслідкувати якщо зважати що найближчий переслідуювач в цьому компоненті – Cocos2D має значення майже вдвічі гірші(18%).[5]

Крім того, за статистикою 2020 року Unity показав надзвичайні результати і має понад 5 мільярдів завантажень додатків, розроблених з його використанням, на місяць, при чому 71% з тисячі найбільш популярних ігор в магазинах AppStore і Google Play і понад 50% ігор на всіх платформах розроблені саме з використанням цього рушія, що є надзвичайно високим результатом[1].

При роботі у актуальній версії Unity можливо використовувати тільки одну мову – C#, хоча раніше (до 2019 року) здійснювалася також підтримка спеціальної мови UnityScript, створеної на основі JavaScript. Проекти за структурою складаються з окремих файлів – сцен, що відрізняються між собою змістом (доданими об’єктами дизайну, музикою, сценаріями, скриптами, тощо)(рисунок 1.2).[6]

Характерними для Unity особливостями є підтримка наслідування об’єктів, а також системи пониження деталізації моделей при збільшенні дистанції зупинка візуалізації об’єктів поза полем зору камер що дозволяє знизити навантаження на систему під час гри, завдяки чому, в тому числі, рушій і заслужив таку велику популярність серед розробників додатків для мобільних платформ, адже вони мають відносно обмежені ресурси, а Unity дозволяє не витратити його на непотрібні елементи, що дозволяє використовувати можливості смартфонів по максимуму[6].

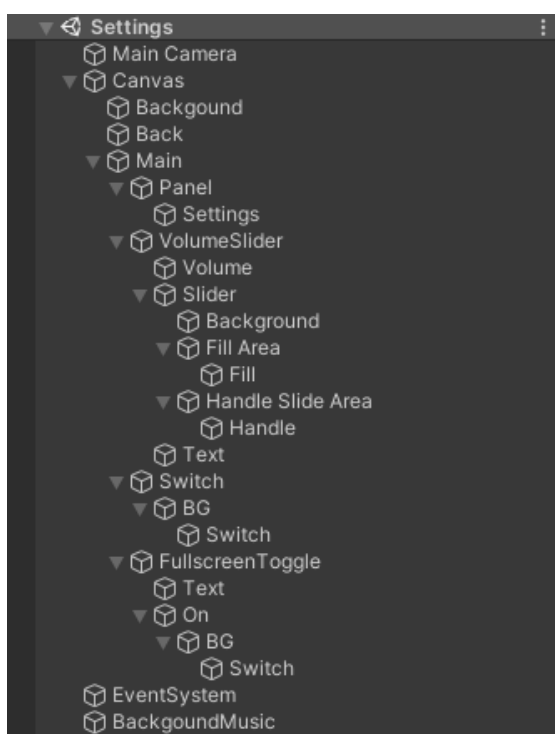


Рис. 1.2 Структура проектів в Unity

## 1.2 Головні переваги

Unity має кілька важливих переваг, які і забезпечують йому таку високу популярність використання. Першою можна назвати зручно виконаний, функціональний та інтуїтивний інтерфейс, завдяки чому використовувати рушій набагато простіше навіть новачкам, що вже сильно розширює базу прихильників серед інді-розробників. Другою є гарно реалізована мультиплатформена підтримка, яка полягає не тільки в широкому спектрі потенційних платформ для розробки, але і можливістю використання рушія на всіх трьох найбільш популярних операційних системах, що дозволяє щороку розширювати базу фанатів як серед гравців, так і серед розробників. До того ж, додатковою перевагою можна вважати наявність влаштованого візуального редактора, що дозволяє полегшити роботу зі сценами шляхом відображення ресурсів та коду проекту в вигляді інтерактивних об'єктів, що ще більше поліпшує поріг входження для новачків.

Також при переліченні переваг Unity що впливають на популярність фреймворку не варто забувати про можливість використання базової версії програми на безкоштовній основі. Не зважаючи на те що вона не надає доступ до всіх функцій інших версій вона дозволяє розробникам-початківцям використовувати усі потужності для створення гри, обмежуючи їх тільки деякими мережевими функціями. Всього існує три версії основного рушія – безкоштовна Personal, а також Pro і Plus.[8] При чому, використовувати персональну ліцензію дозволено тільки для компаній річний оборот яких не перевищує 100 тисяч доларів. Для компаній з оборотом до 200 тисяч рекомендується версія Plus, а у інших випадках нічого крім Pro не допускається. Усі вони дають доступ до використання офіційного магазину як платних, так і вільних у використанні ассетів[7] і їх використання без додаткових виплат.

### 1.3 Основні недоліки

Не зважаючи на доволі вагомий список переваг, Unity однозначно має і свої недоліки. Першим з них можна назвати повільність у роботі. Цей фактор найчастіше впливає з таких додатків які розраховані на масштабні складні сцени. Через це доводиться використовувати сторонні утиліти, які ще більше навантажують систему, що є доволі проблематично, особливо при використанні на пристроях обладнаних HDD-пам'яттю.

Великий об'єм пам'яті який необхідний займає готова гра. Навіть невеликий проект може потребувати відносно велику частину місця на пристрої, що може бути критично під час розробки на мобільні платформи.

Крім того, присутня певна обмеженість, пов'язана з неможливістю управління деякими «вшитими» параметрами, починаючи з недостатньої гнучкості налаштування камери і закінчуючи базою даних, що є критичним при розробці невеликих ігор, але стає проблемою при спробі створити повноцінний AAA-проект.

При роботі відсутня можливість посилань на зовнішні бібліотеки, через що їх доводиться копіювати у проект вручну[6]. А також існує мінус пов'язаний з начебто зручною системою префабів, які з одного боку додають варіативності при роботі з візуальною складовою інтерфейсу, а з іншої їх власне редагування може викликати певні складності при спробі реалізації деяких змін.

## РОЗДІЛ 2 РОЗРОБКА ПРОЕКТУ

### 2.1 Планування розробки

Темою курсової роботи є розробка 2D квесту в форматі вікторини, з можливістю необмеженого розширення кількості запитань і фіксуванням набраних балів в кожній з ігор.

Аналіз ринку ігрових рушіїв і створених на них іграх показав, що найкращим вибором для даного жанру є Unity, у зв'язку з тим що базовий функціонал ідеально підходить для розробки зручного користувацького інтерфейсу, а мова C# дозволяє вдало реалізувати просте редагування і додавання нових питань в середині фреймворку. Крім того, більшість конкурентів, навіть якщо і мають схожі можливості, більше призначенні для створення 3D зразків, що не корелює з встановленою темою роботи.

Головною складовою розробки було створення зручного, на рівні інтуїції простого дизайну, який було б легко опанувати дітям, на яких даний проект і розрахований, а також написання корисного функціоналу, що потенціально, на рівні проекту, дозволив би умовним батькам або вчителям редагувати, змінювати і додавати питання на свій розсуд.

### 2.2 Реалізація

Під час виконання проекту для написання скриптів було використано мову C#, адже починаючи з 2019 року, після видалення UnityScript, вона є єдиною доступною для даного рушія.

У грі було використано кілька візуальних і один музичний ассет з офіційного магазину[7]. Не зважаючи на це, більшість з них були перепрацьовані для того щоб відповідати заданим умовам.

Для забезпечення повноцінного приємного геймплею було використано стандартні можливості рушія в симбіозі з власними скриптами. Яскравим прикладом є налаштування роботи музики на всіх сценах, при тому що при

стандартному додаванні музичного супроводу, аудіозапис стартує заново при завантаженні кожної нової сцени.

### 2.3 Опис GUI та меню

Усі елементи GUI були створені на основі перероблених ассетів, завантажених з офіційного магазину. Після запуску файлу гри користувач бачить стартовий екран-меню, що складається з чотирьох пунктів: «New Game», «Exit», та іконок налаштувань і додаткової інформації («Credits»)(рисунок 2.1). Саме на ній стартує аудіо супровід який є спільним для усіх сцен.

Іконка додаткової інформації веде на сцену з іменами розробників(рисунок 2.2), а налаштувань – відкриває відповідну сторінку, де є можливість визначити чи бажає користувач грати зі звуком, змінити його гучність, а також переключитися між повноекранним і віконним режимом(рисунок 2.3). В перспективі, налаштуванням можна надати більше варіацій, наприклад додати можливість вибору якості графіки, або точного розширення екрану для збільшення бази потенційних користувачів.

Клавіша «Exit» дозволяє миттєво закрити додаток. В свою чергу, «New Game» відправляє користувача в наступну сцену, безпосередньо на сторінку вікторини.



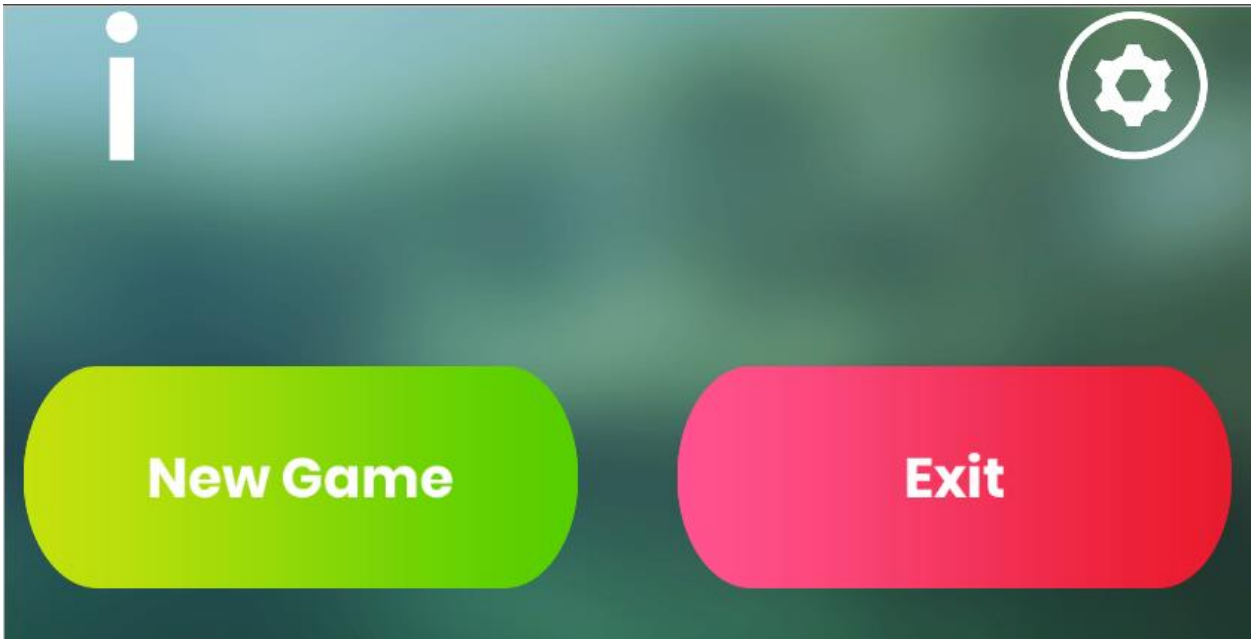


Рис. 2.1 Головне меню

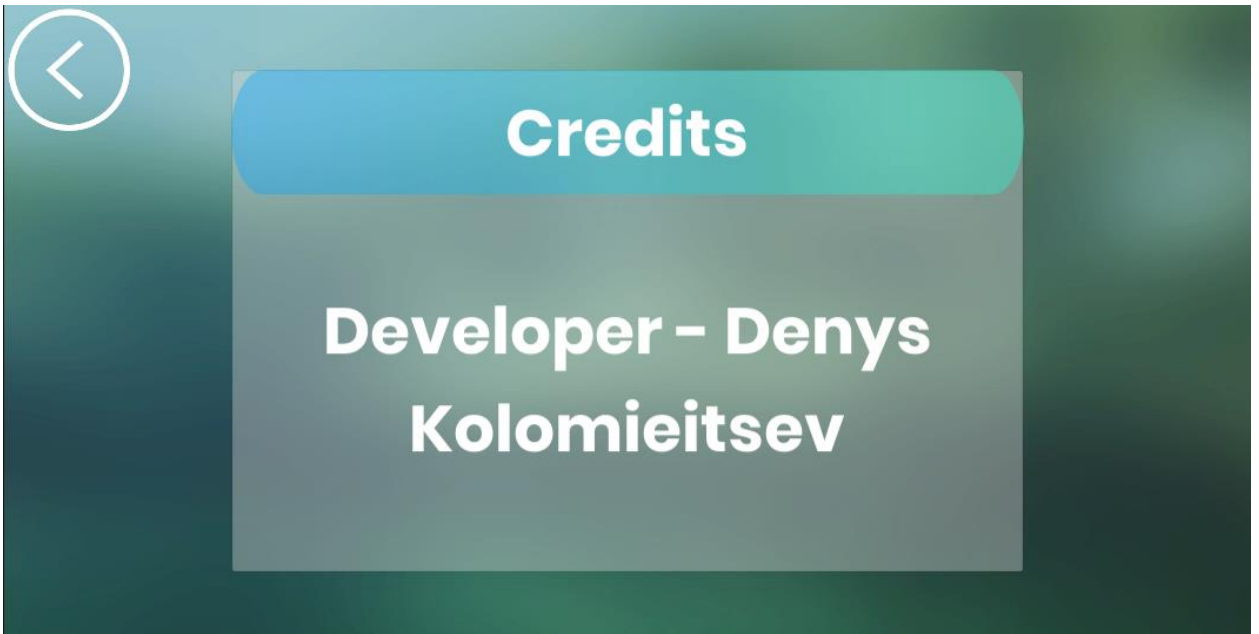


Рис. 2.2 Меню «Про розробників»

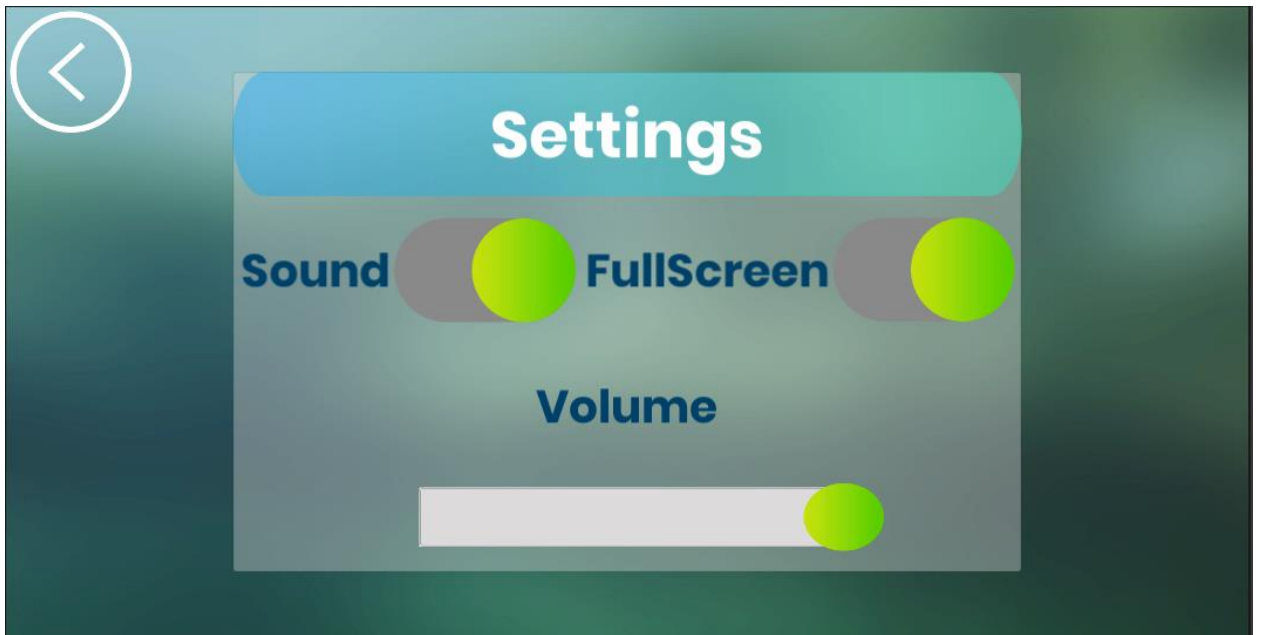


Рис. 2.3 Меню налаштувань

#### 2.4 Початок гри

На сцені квесту(рисунок 2.4) гравець побачить панель з питанням, чотири активних елемента вибору відповіді, «зірку» на якій будуть нараховуватися його правильні відповіді, а також іконку повернення до меню, яку можна використати в будь-який момент.

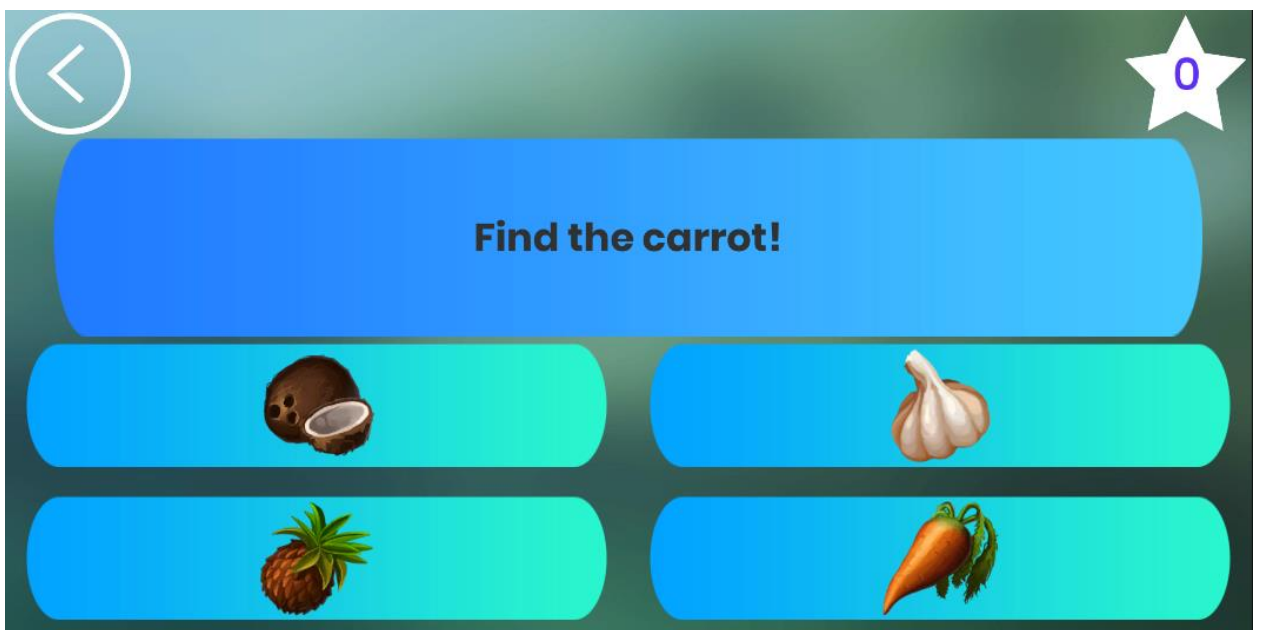


Рис 2.4 GUI вікторини

Завдяки створеному скрипту, порядок поставлених питань і відповідей до них буде випадково обиратися зі списку заданих питань, який можна змінити або доповнити в налаштуванні проекту(рисунок 2.5).

Зміна параметру «Quiz» дозволяє виставити бажану кількість питань на одну гру. При збільшенні чи зменшенні показника додаються нові панелі, в яких наявні наступні пункти:

- «Ask» - текст питання що буде виведене на екран;
- Параметр «Var» дозволяє змінити кількість варіантів для відповіді(рекомендовано 4);
- Елементи 0-3 відповідають за спрайти картинок що будуть відображатися як варіанти відповіді;
- Параметр «Right» - встановлює який з заданих елементів є правильною відповіддю на задане питання.

Не зважаючи на правильність відповіді буде відображено наступне запитання з відповідними варіантами. Якщо ж гравець обрав вірний варіант, число в «зірці» в правому верхньому куті зміниться на одиницю. Максимально набрана кількість очок прямо пропорційна встановленій кількості питань.

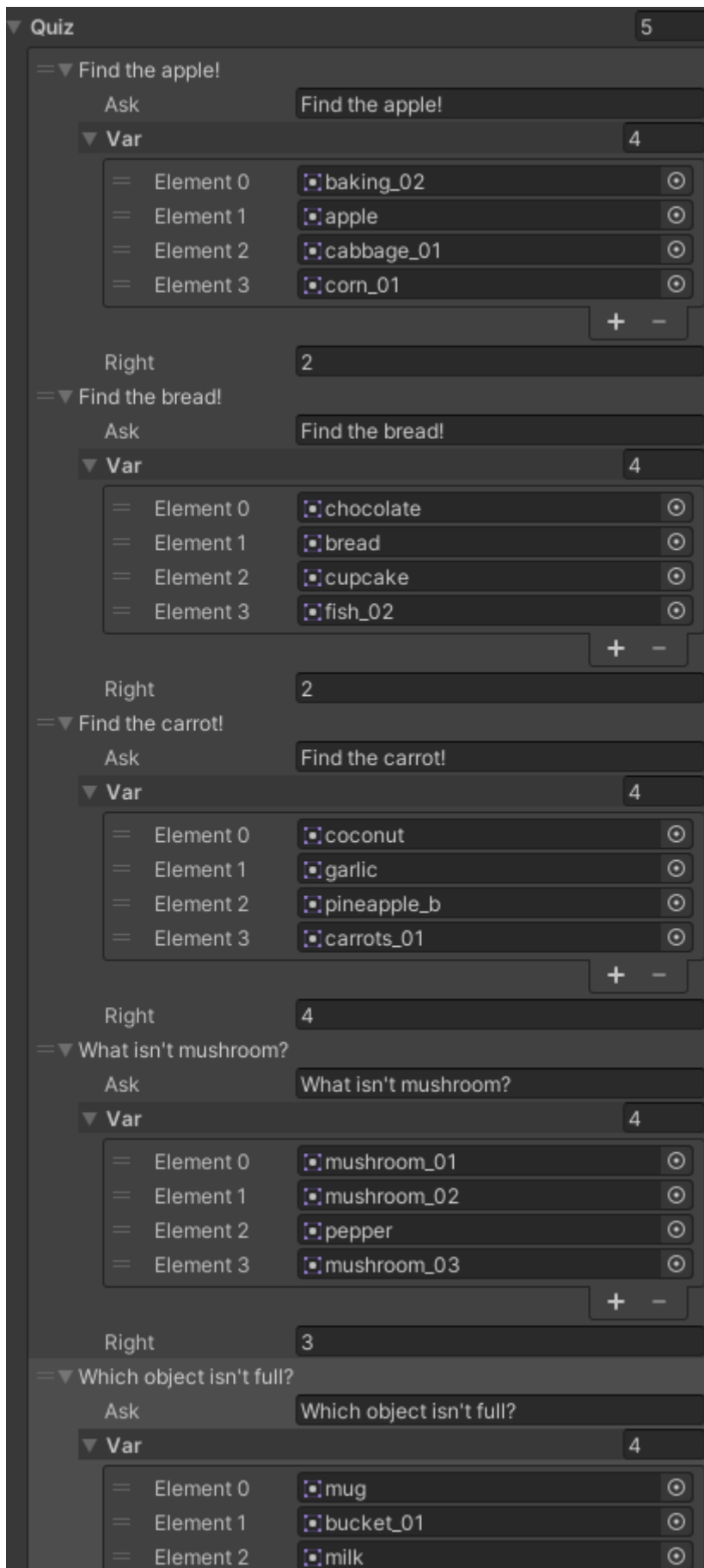


Рис. 2.5 Панель налаштування питань

## 2.5 Ціль гри

Ціллю гри є використання візуального ряду для створення асоціативних аналогій у дітей, шляхом створення пар зображення-назва задля поліпшення запам'ятовування знайомих слів і вивчення нових. Ігровий процес дає можливість проходити створенні питання повторно в разі необхідності, а можливість редагування(рисунки 2.5) дасть можливість вчителям або батькам створювати нові завдання після повного вивчення попередніх.

Технічною ціллю гри є надання максимально можливої кількості правильних відповідей. Після завершення вікторини, гравець побачить перед собою фінальний екран, який сповістить його скільки балів він набрав з n-ної можливої кількості, яка залежить від параметру «Ask»(рисунки 2.6). Також на ньому буде відображена кнопка «Go to menu», яка відправить гравця на екран меню, звідки він зможе повторно почати тест.

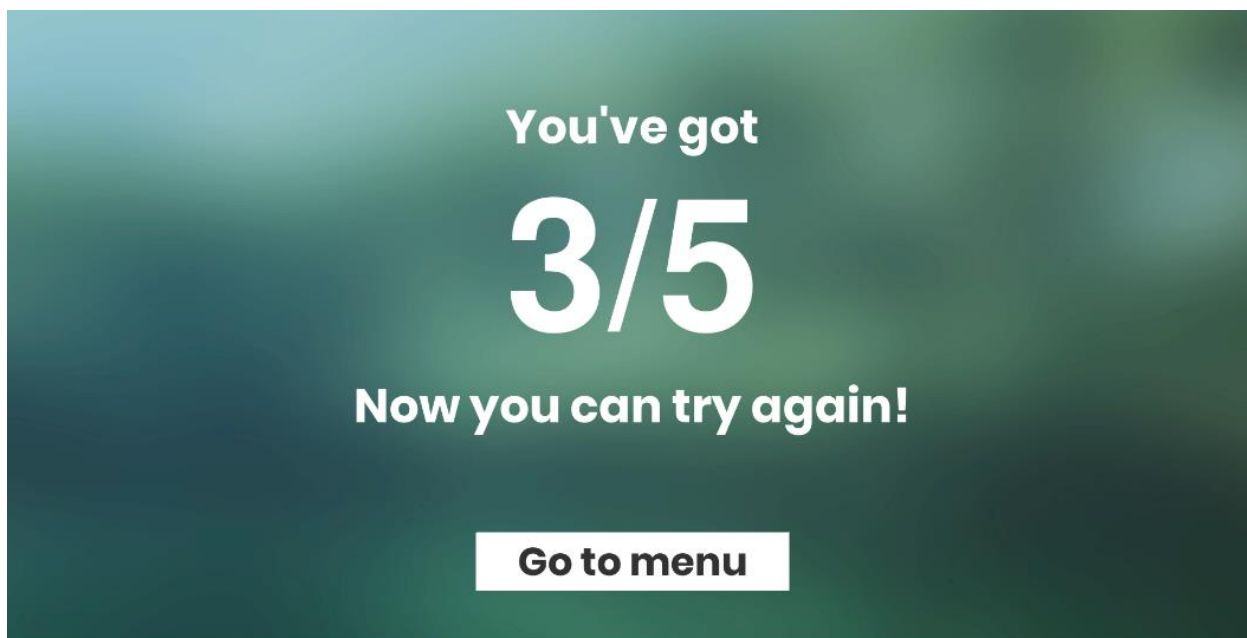


Рис. 2.6 Фінальний екран

## 2.6 Призначення скриптів

### 2.6.1 Класи вікторини

#### 2.6.1.1 Квест

Основним класом гри є клас «Quest» який відповідає за налаштування питань, варіантів відповідей, встановлює для них формат спрайтів, генерує випадковий порядок відображення питань користувачу, а так відповідає за підрахунок і виведення на екран очок і їх загальної кількості, набраної гравцем під час проходження вікторини.

```
public class Quest : MonoBehaviour
{
    public List<QA> quiz;
    public GameObject[] variants;
    public int Question;
    public GameObject OverhaulNum;
    public GameObject ResultNum;
    public Text QLog;
    public Text PointShow;
    public Text StarShow;
    int allCount = 0;
    public int points;

    private void Start()
    {
        allCount = quiz.Count;
        ResultNum.SetActive(false);
        Generation();
    }
    public void Finish()
    {
        OverhaulNum.SetActive(false);
        ResultNum.SetActive(true);
        PointShow.text = points + "/" + allCount;
    }
    public void yes()
    {
        points += 1;
        quiz.RemoveAt(Question);
        Generation();
    }
    public void no()
    {
        quiz.RemoveAt(Question);
        Generation();
    }
}
```

```

void Ans()
{
    for (int i = 0; i < variants.Length; i++)
    {
        variants[i].GetComponent<Options>().RightVar = false;
        variants[i].transform.GetChild(0).GetComponent<Image>().sprite = quiz[Question].Var[i];

        if (quiz[Question].Right == i+1)
        {
            variants[i].GetComponent<Options>().RightVar = true;
        }
    }
}

void Generation()
{
    if(quiz.Count >0)
    {
        Question = Random.Range(0, quiz.Count);

        QLog.text = quiz[Question].Ask;
        Ans();
        StarShow.text = points.ToString();
    }
    else
    {
        Debug.Log("Level Passed!");
        Finish();
    }
}
}
}

```

### 2.6.1.2 Додаткові класи

Крім того, клас «Quest» використовує два додаткових класи. Клас «QA» оголошує необхідні дані, а клас «Options», в свою чергу, служить для перевірки правильності обраного варіанту (опції) відповіді користувачем.

```

public class Options : MonoBehaviour
{
    public bool RightVar = false;
    public Quest quest;

    public void Var()
    {
        if(RightVar)
        {
            quest.yes();
        }
        else
        {
            quest.no();
        }
    }
}

[System.Serializable]
public class QA
{
    public string Ask;
    public Sprite[] Var;
    public int Right;
}

```

## 2.6.2 Загальні класи

В додатку присутні певні класи які впливають на всю гру комплексно. До таких класів можна віднести клас «Buttons» який забезпечує перехід на інші сцени, задані розробником в налаштуванні або закриття додатку і клас «AllMusic» який дозволяє налаштувати рівномірну спільну роботу саундтреку без повторювання з початку при зміні сцени. Цей результат був досягнутий завдяки тому, що під час запуску гри створюється додаткова тимчасова сцена, яка і відповідає за постійне програвання саундтреку.

```
public class Buttons : MonoBehaviour
{
    public void LoadScene (string sceneName)
    {
        SceneManager.LoadScene(sceneName);
    }
    public void CloseApp()
    {
        Application.Quit();
    }
}

public class AllMusic : MonoBehaviour
{
    void Awake()
    {
        GameObject[] musicObj = GameObject.FindGameObjectsWithTag("Song");
        if(musicObj.Length > 1)
        {
            Destroy(this.gameObject);
        }
        DontDestroyOnLoad(this.gameObject);
    }
}
```

## 2.6.3 Скрипти налаштування

На панелі налаштувань(рисунок 2.3) використовується декілька скриптів. Це скрипти «Mute», «Fullscreen», «Switch» та «Settings».



Скрипт «Switch» відповідає за роботу зроблених в ручну перемикачів звуку і переходу на повноекранний або віконний режим. Він переміщує об'єкту «Switch» у сцені в момент його активації(рисунок 2.8-2.9)

```
public class Switch : MonoBehaviour
{
    [SerializeField] RectTransform moveSwitch;

    Toggle toggle;

    Vector2 changePosition;

    void Start()
    {
        toggle = GetComponent <Toggle>();

        changePosition = moveSwitch.anchoredPosition;

        toggle.onValueChanged.AddListener(On);

        if(toggle.isOn)
            On (true);
    }

    void On(bool on)
    {
        if (on)
            moveSwitch.anchoredPosition = changePosition * -1;
        else
            moveSwitch.anchoredPosition = changePosition;
    }

    void Off()
    {
        toggle.onValueChanged.RemoveListener(On);
    }
}
```

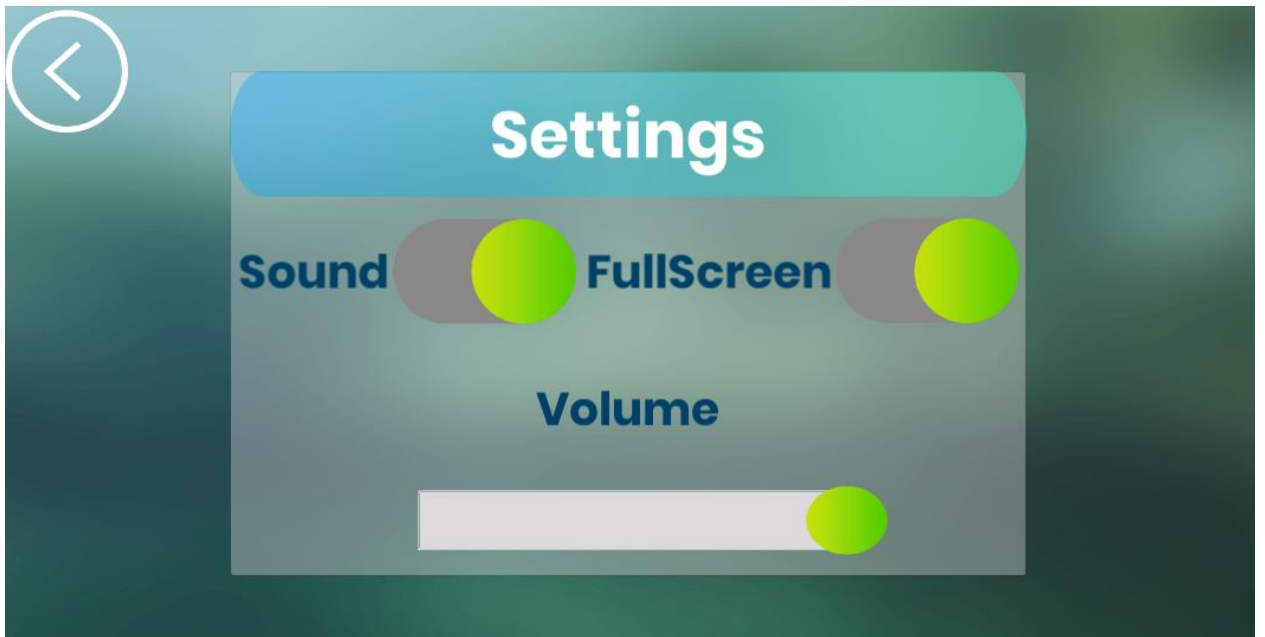


Рис. 2.8 Базове положення вимикачів

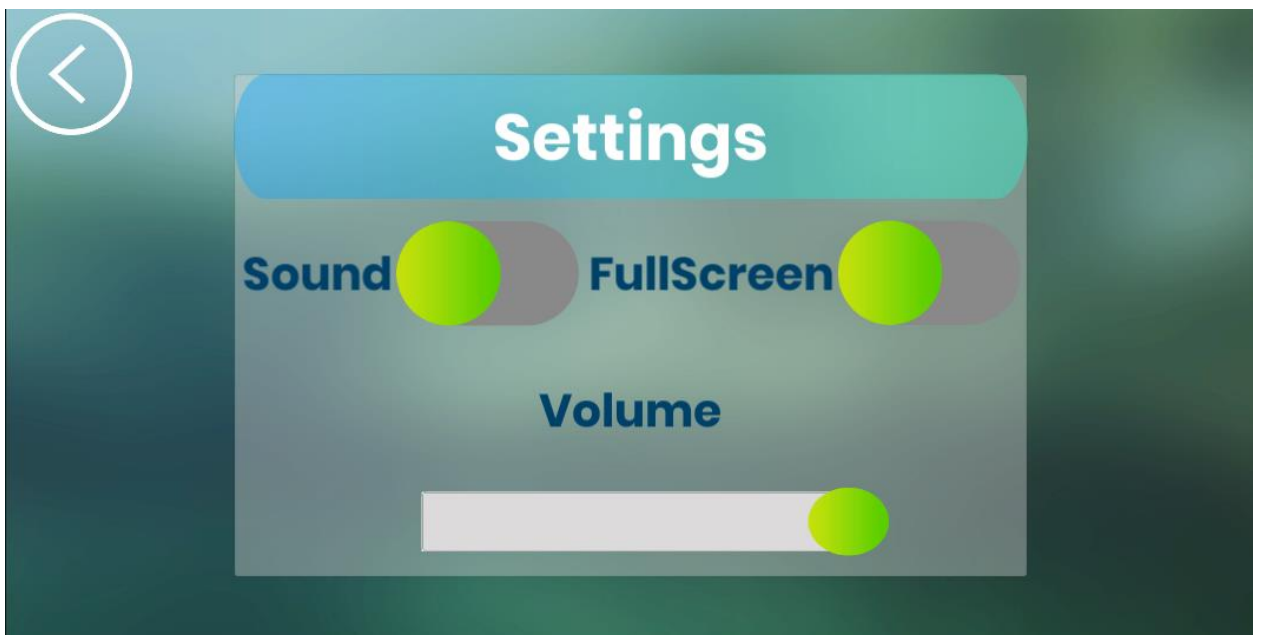


Рис. 2.9 Альтернативне положення вимикачів після активації

Скрипт «Mute» прив'язаний до вимикача «Sound» і виконує функцію деактивації або активації саундтреку у грі.

```

public class Mute : MonoBehaviour
{
    private bool SwitchOff;

    void Start()
    {
        SwitchOff = false;
    }

    public void Muted()
    {
        SwitchOff = !SwitchOff;

        AudioListener.pause = SwitchOff;
    }
}

```

За перехід від повноекранного режиму до віконного і навпаки відповідає скрипт «FullScreen».

```

public class FullScreen : MonoBehaviour
{
    public void FS(bool is_fullscreen)
    {
        Screen.fullScreen = is_fullscreen;
    }
}

```

За підв'язування рівня звуку до слайдеру «Volume» відповідає скрипт «Settings». Він передає дані слайдеру стандартному інструменту Unity «Audio Mixer», в результаті чого змінюється гучність в межах -80 – 0 децибел.

```

public class Settings : MonoBehaviour
{
    private bool VolumeToggle;
    public AudioManager audioMixer;

    void start()
    {
        VolumeToggle = false;
    }

    public void Muted()
    {
        VolumeToggle = !VolumeToggle;
        AudioListener.pause = VolumeToggle;
    }

    public void VolumePer (float MusicVolume)
    {
        audioMixer.SetFloat("MusicVolume", MusicVolume);
    }
}

```

## РОЗДІЛ 3 НАЛАШТУВАННЯ ЕЛЕМЕНТІВ РУШІЯ

### 3.1 Елементи екрану налаштування

Налаштування слайдери гучності на сцені «Settings» відповідають за його коректну роботу і відображення, а також за відповідність значенням встановленим стандартним компонентом «Audio Mixer»(рисунок 3.1).

За допомогою компонента «Toggle» було налаштовано параметри елементів перемикача присутнього в елементах «Sound» і «Fullscreen», завдяки яким присутня кастомна анімація під час виконання «OnClickEvent»(рисунок 3.2).

### 3.2 Елементи екрану меню

На сцені меню було створено допоміжний об'єкт «Song», який має стандартний компонент «Audio Source»(рисунок 3.3) який використовується для програвання музики на усіх сценах за допомогою скрипта «AllMusic».

### 3.3 Елементи екрану гри

На основній сцені гри скрипт «Quest» дозволяє налаштувати об'єкти, які будуть отримувати необхідні дані. Кожен з пунктів відповідає за:

- Overhaul Num – основна панель на якій знаходяться питання і відповіді;
- Result Num – панель фіналу на яку виводяться фінальні результати;
- Q Log – елемент який буде отримувати тексти питань зі скрипта;
- Point Show – елемент який отримує дані результату гравця для фінального екрану;
- Star Show – елемент який отримує дані результату гравця для відображення в «зірці».

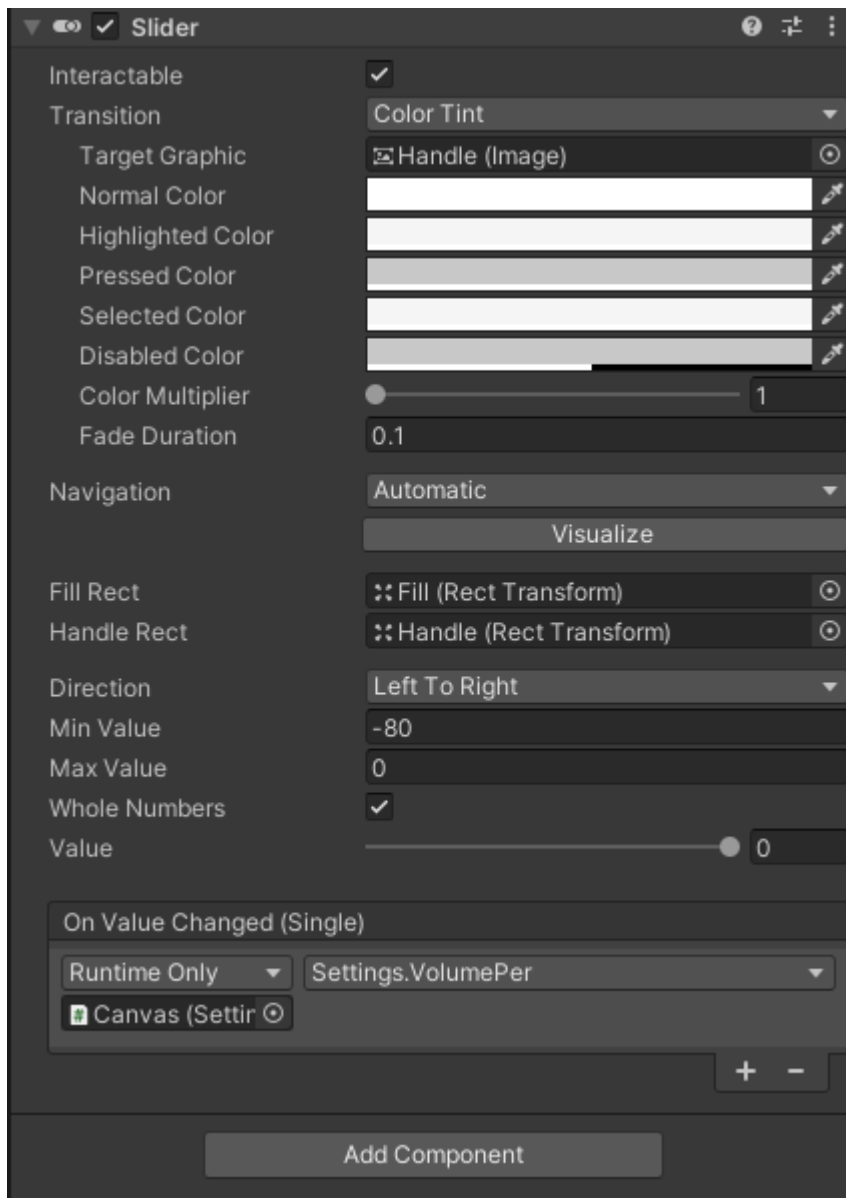


Рис. 3.1 Налаштування компоненту «Slider»

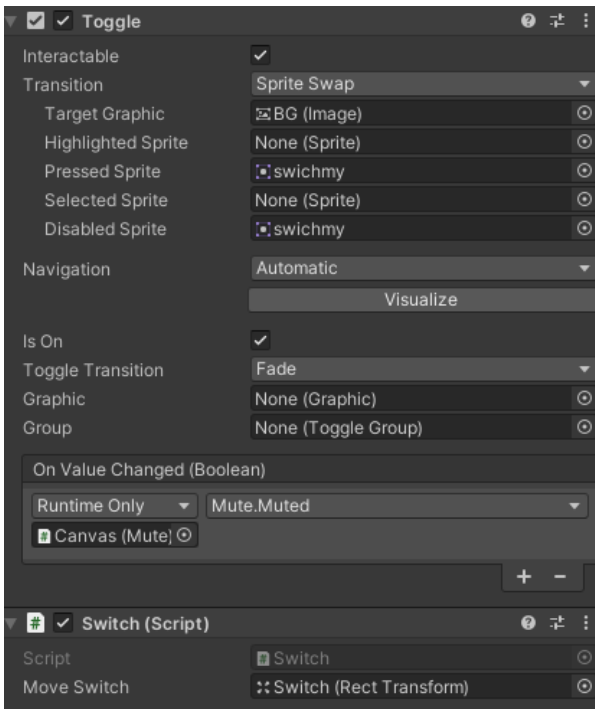


Рис. 3.2 Налаштування компоненту «Toggle»



Рис. 3.3 Налаштування елемента «Audio Source»

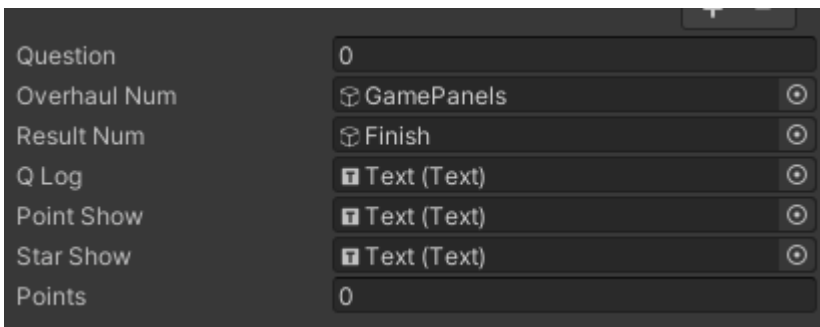


Рис. 3.4 Налаштування скрита «Quest»

## ВИСНОВКИ

В процесі виконання даного проекту було досліджено особливості створення відеоігор з двовимірною графікою за допомогою ігрового рушія Unity для операційної системи Linux і розроблено прототип розвиваючого квесту для допомоги дітям в вивченні англійської мови на його основі.

Під час підготовки до проекту було зазначено що даний рушіє є надзвичайно зручним для роботи з 2D інтерфейсом, що серйозно спрощує задачу в порівнянні з аналогами. Це, а також інші переваги, такі як можливість безкоштовного користування, легкість в використанні і якісна функціональність, можна вважати причиною настільки великої користувацької бази Unity Engine.

Важливою можливістю даного рушія для розробки 2D застосувань варто зазначити можливість редагування спрайтів, велику кількість корисних вбудованих компонентів і легкість підключення необхідних власних скриптів. С# повністю відповідає всім вимогам для зручного використання у даному фреймворку.

Якісним доповненням є власний магазин ассетів, який дозволяє розробникам не витрачати зайвий час на малювання власних елементів інтерфейсу і спрайтів, а використовувати вже існуючі, виправляючи і змінюючи їх за необхідністю



## СПИСОК ЛІТЕРАТУРИ

1. Unity Growth Facts [Електронний ресурс] – Режим доступу до ресурсу: <https://unity3d.com/ru/public-relations>
2. Genshin Impact works its magic to become biggest global launch of a Chinese game ever [Електронний ресурс] – Режим доступу до ресурсу: <https://www.scmp.com/tech/apps-social/article/3103522/genshin-impact-works-its-magic-become-biggest-global-launch>
3. Unity User Manual 2020.3 (LTS) [Електронний ресурс] – Режим доступу до ресурсу: [docs.unity3d.com/Manual/index.html?\\_ga=2.147061943.1679198015.1618231226-792086137.1617460014](https://docs.unity3d.com/Manual/index.html?_ga=2.147061943.1679198015.1618231226-792086137.1617460014)
4. What's new in Unity 2021.1.2 [Електронний ресурс] – Режим доступу до ресурсу: <https://unity3d.com/unity/whats-new/2021.1.2>
5. How to deliver machine learning in a large company? A conversation with Danny Lange, VP of AI and ML at Unity [Інтернет-стаття] - Режим доступу до ресурсу: <https://www.linkedin.com/pulse/how-deliver-machine-learning-large-company-danny-lange-azeem-azhar/>
6. Introduction to Game Design, Prototyping, and Development: From Concept to Playable Game with Unity and C# / Jeremy Gibson Bond., 2019 – (Addison-Wesley).
7. Unity Asset Store [Електронний ресурс] - <https://assetstore.unity.com/>
8. Unity Store [Електронний ресурс] – Режим доступу до ресурсу: <https://store.unity3d.com/>.