

Рисунок 3.1 - Вкладка External Tools

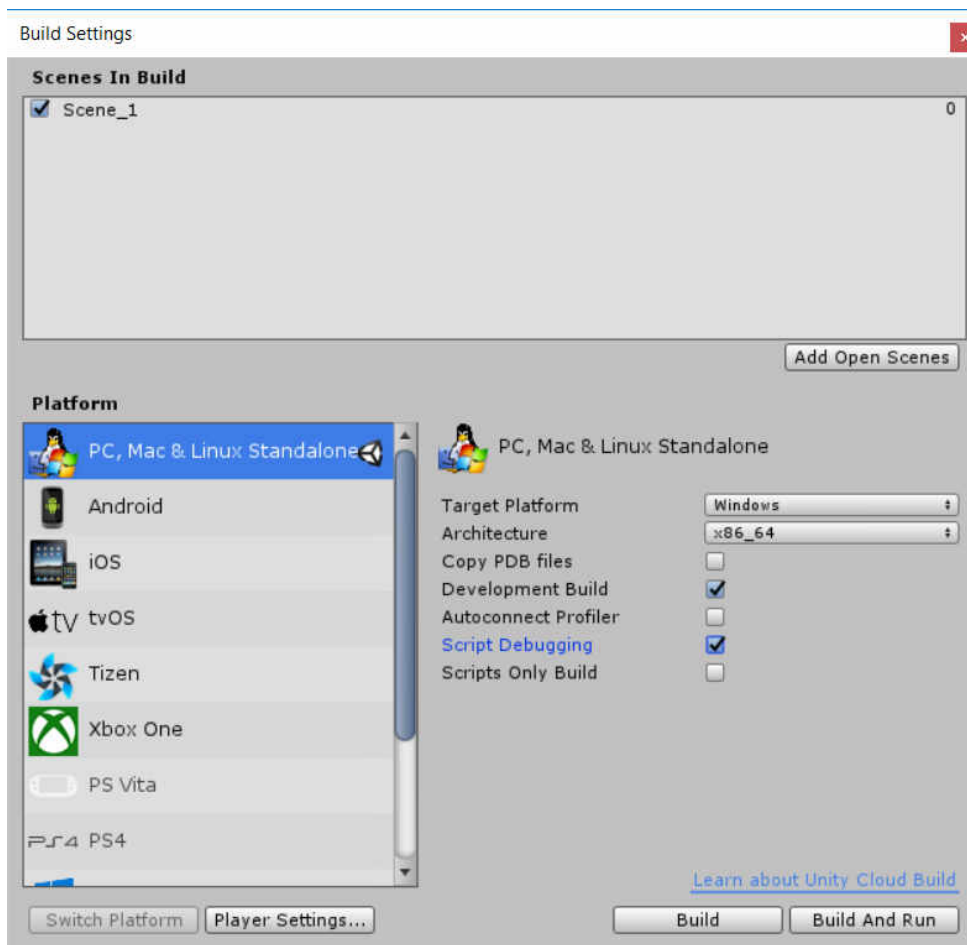


Рисунок 3.2 - Вікно BuildSettings

Огляд можливостей налаштування.

Для налаштування **MonoDevelop** відкриємо вікно **Options** за допомогою меню **Tools**→**Options**.

Розділ **Environment**

Author Information. Заповнюється за бажанням, можна вказати своє ім'я, електронну адресу, назву організації, інформацію про товарні знаки і авторські права (рис. 3.3).

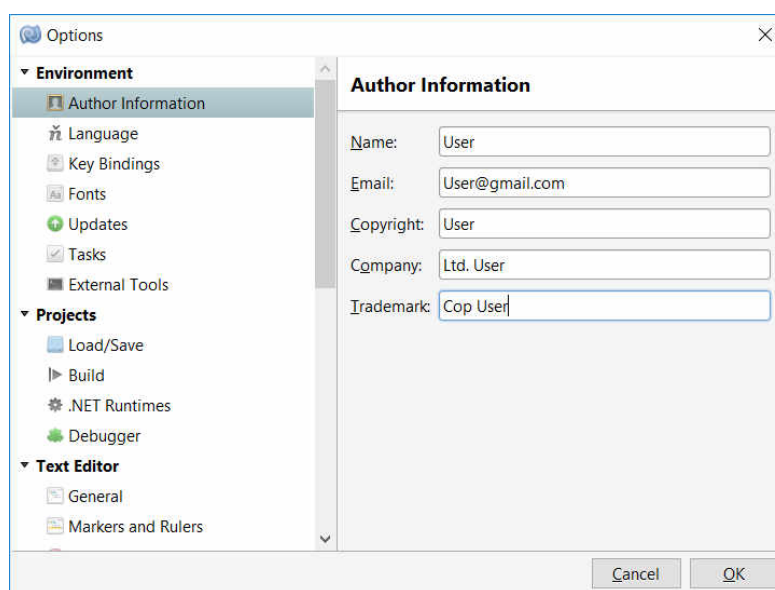


Рисунок 3.3 - Вікно **Options**. Розділ **Environment**

Language. Налаштування мови. Рекомендуємо встановити англійську мову (рис. 3.4).

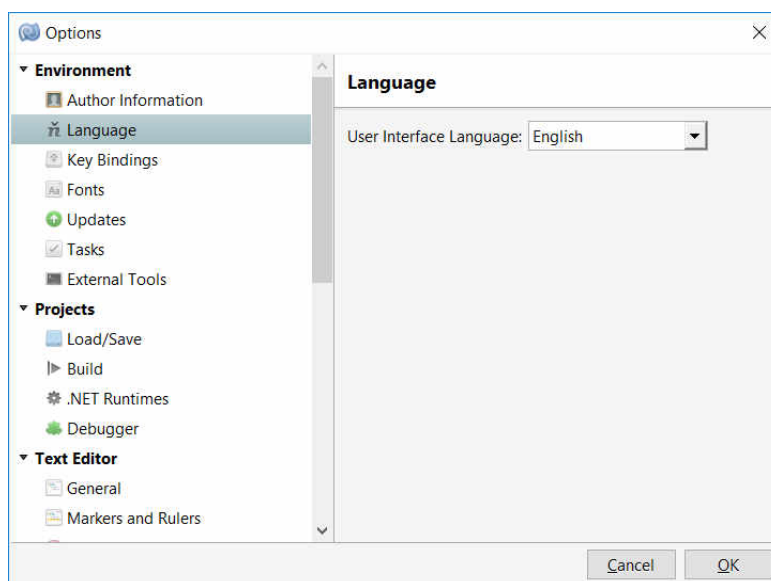


Рисунок 3.4 - Вікно **Options**. Розділ **Language**

Key bindings. Налаштування клавіатурних комбінацій, причому є і така річ як схема клавіатурних комбінацій (рис. 3.5).

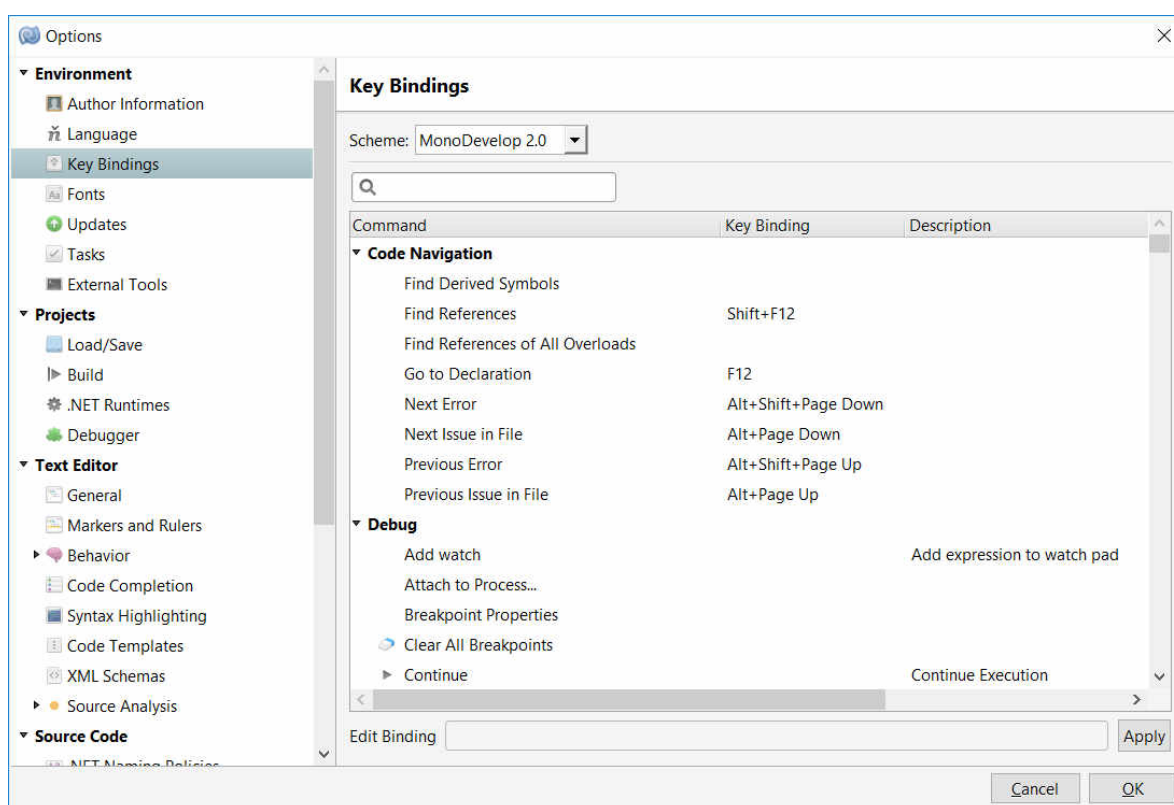


Рисунок 3.5 - Вікно **Options**. Розділ **Key bindings**

Схеми клавіатурних комбінацій дозволяють налаштувати прив'язки клавіш такі як наприклад, в **Visual Studio**. Найцікавіша річ - це пункт, який з'являється в разі наявності однієї і тієї ж дії на різні комбінації, і він

називається **View Conflicts**. Ця опція покаже комбінації з конфліктами і допоможе їх усунути, а корисною вона стає через те, що дуже часто звичайні комбінації в **Monodevelop** починають з незрозумілої причини відмовляти - і, якщо ви спостерігаєте таке в себе, то варто скористатися **View Conflicts** і усунути конфлікти прив'язок.

Fonts. Налаштування шрифтів, як для інтерфейсу, так і для редактора (рис. 3.6).

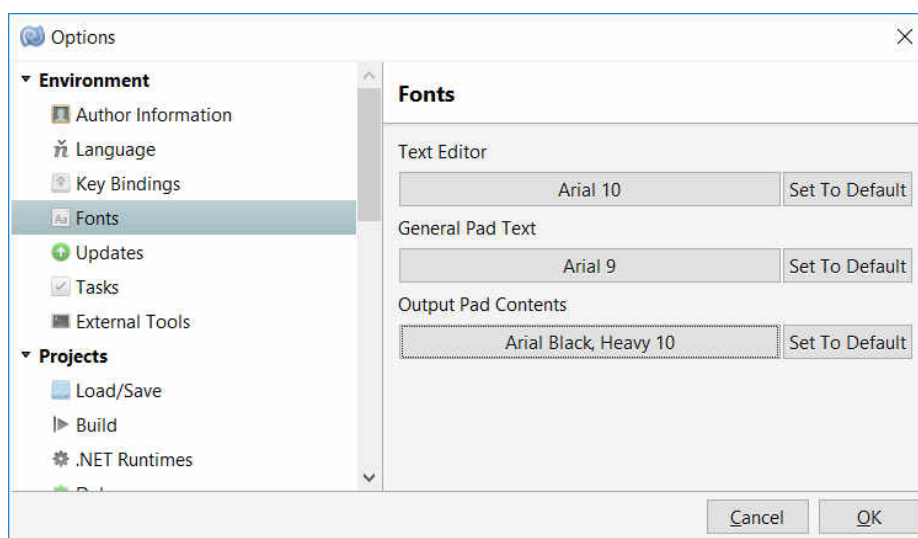


Рисунок 3.6 - Вікно **Options**. Розділ **Fonts**

Розділ **Projects**

Build. Загальні настройки процесу збірки (рис. 3.7).

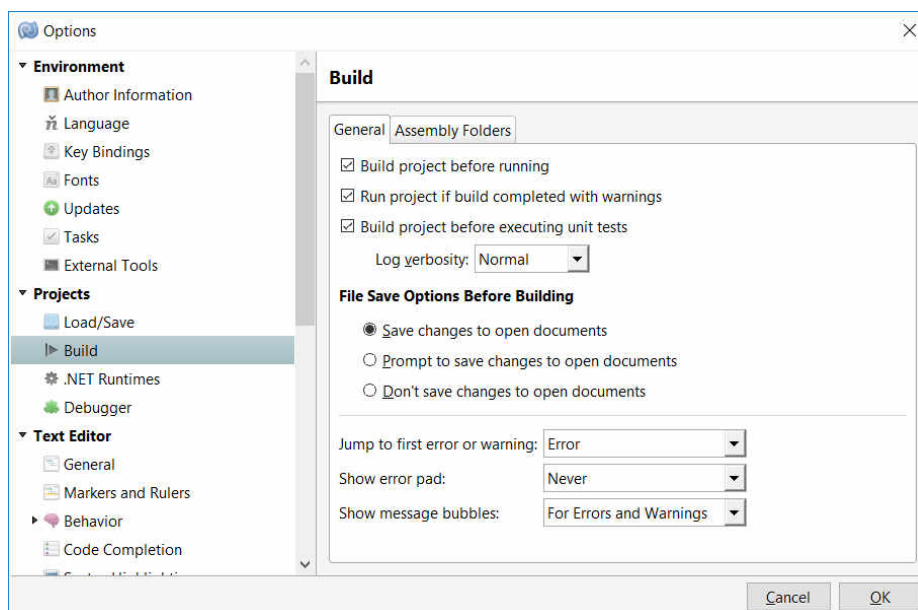


Рисунок 3.7 - Загальні настройки процесу збірки

Після налаштування параметрів в розділі **Build** отримуємо такі можливості: автоматична збірка перед виконанням/запуском юніт-тестів, автоматичне збереження змін під час запуску/збірки, деталізований висновок інформації про помилки, автоматичне підсвічування червоним кольором рядки з помилкою плюс у спливаючому вікні буде виведено інформацію про те, що сталося (іншими словами, підсвічування помилок з інформацією про них з перших).

Розділ Text Editor

General. Встановлюємо такі опції (рис. 3.8):

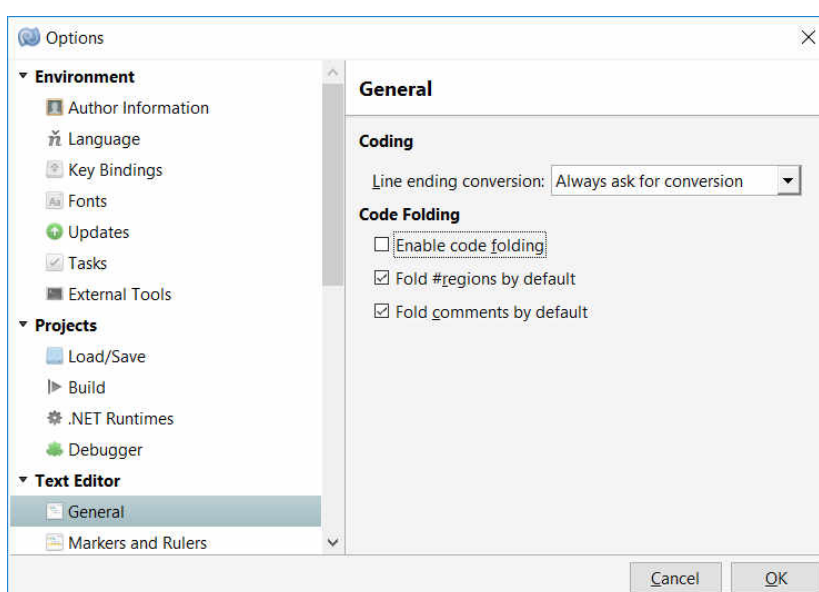


Рисунок 3.8 - Вікно **Options**. Розділ **General**

У підсумку, отримуємо згортку коментарів за замовчуванням, згортку регіонів коду і запити на конвертацію символу рядка (якщо виникне така потреба).

Markers and Rulers. Налаштування всякого роду маркерів, позначок, які допомагають орієнтуватися в коді (рис. 3.9).

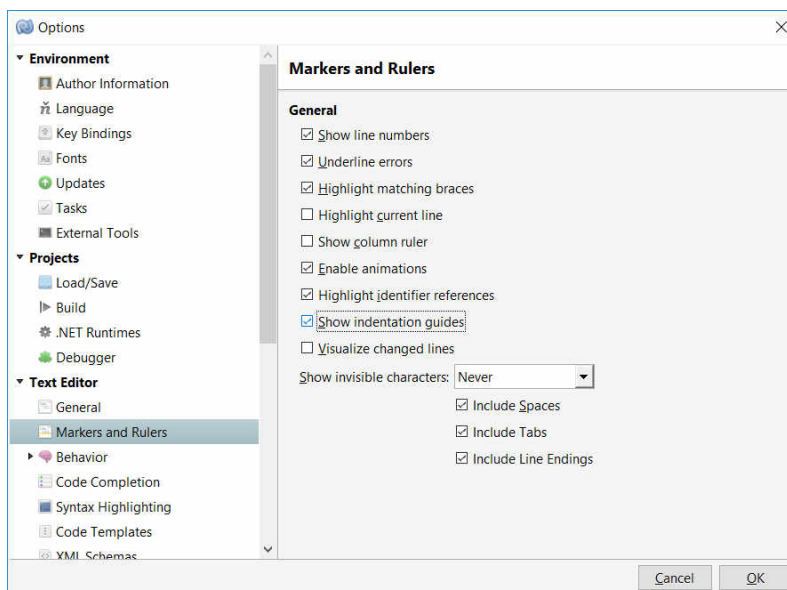


Рисунок 3.9 - Вікно **Options**. Розділ **Markers and Rulers**

Після налаштування параметрів в розділі **Markers and Rulers** отримуємо відображення номерів рядків, підкреслення помилок (відзначаються прямо на ходу: щось помилково набрали на думку аналізатора - і це відразу ж буде підкреслено), підсвічування збігу дужок, посилань на ідентифікатори, а крім того маємо відображення ліній індену (тобто графічне відображення рівня відступу). Нижні опції відключають візуалізацію змінених рядків і відображення невидимих символів (пропуски, переклади рядків, кінці файлів і т.п).

Behavior. Налаштування для форматування коду (рис. 3.10).

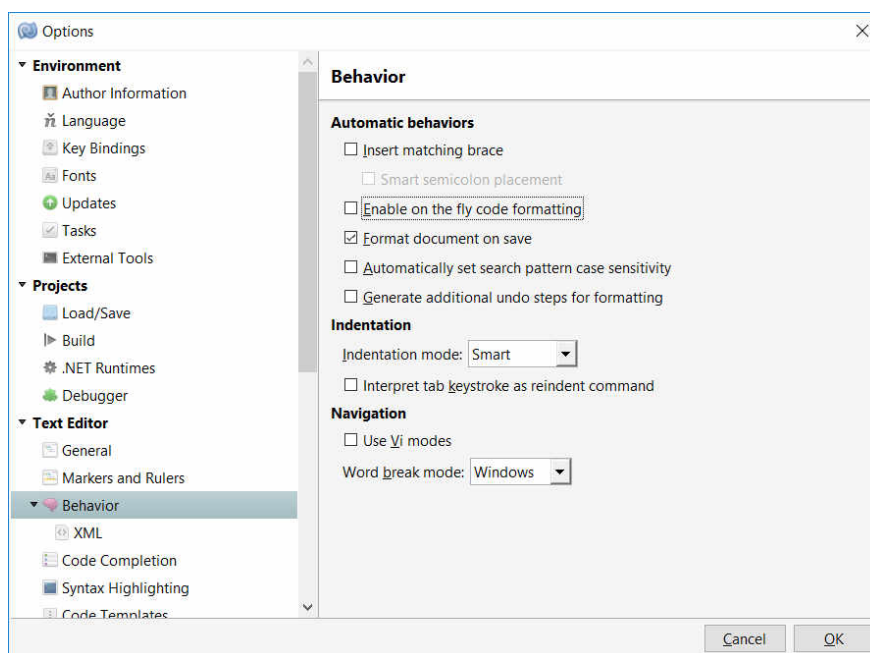


Рисунок 3.10 - Вікно **Options**. Розділ **Behavior**

Code Completion. Налаштування автодоповнень (рис. 3.11).

Після налаштування параметрів в розділі **Code Completion** в автодоповненні будуть показуватися і обсяги імпорту, члени класів будуть показуватися в розширеному вигляді і доповнення по ним також буде розширено.

Крім того, в списку під автодоповненням (список виникає після натискання клавіші **Tab**) буде як мінімум 10 рядків (тобто збільшено обсяг списку).

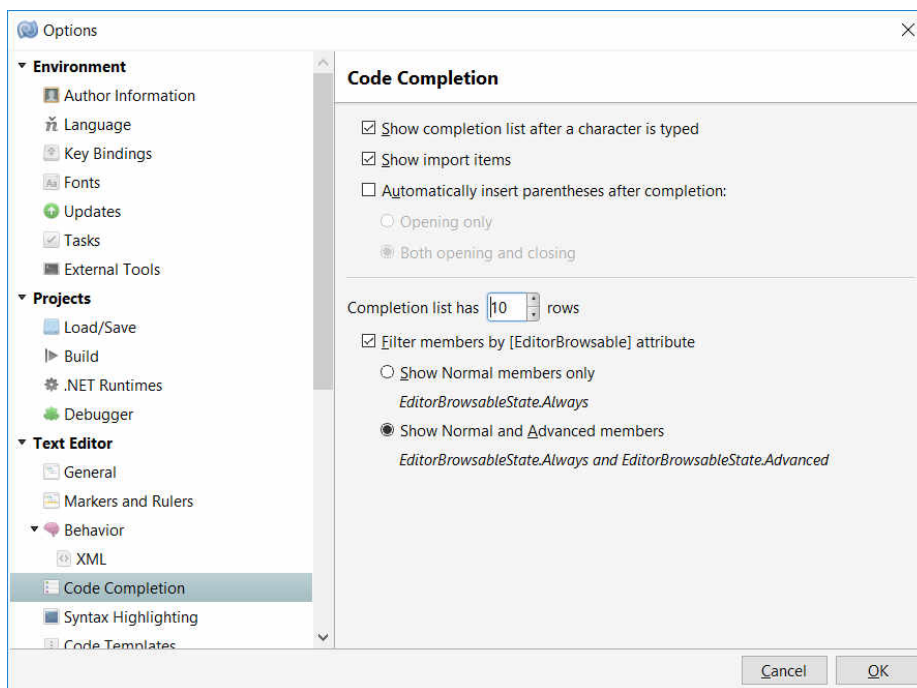


Рисунок 3.11 - Вікно **Options**. Розділ **Code Completion**

Syntax Highlighting. Зміна колірної схеми підсвічування коду (рис. 3.12).

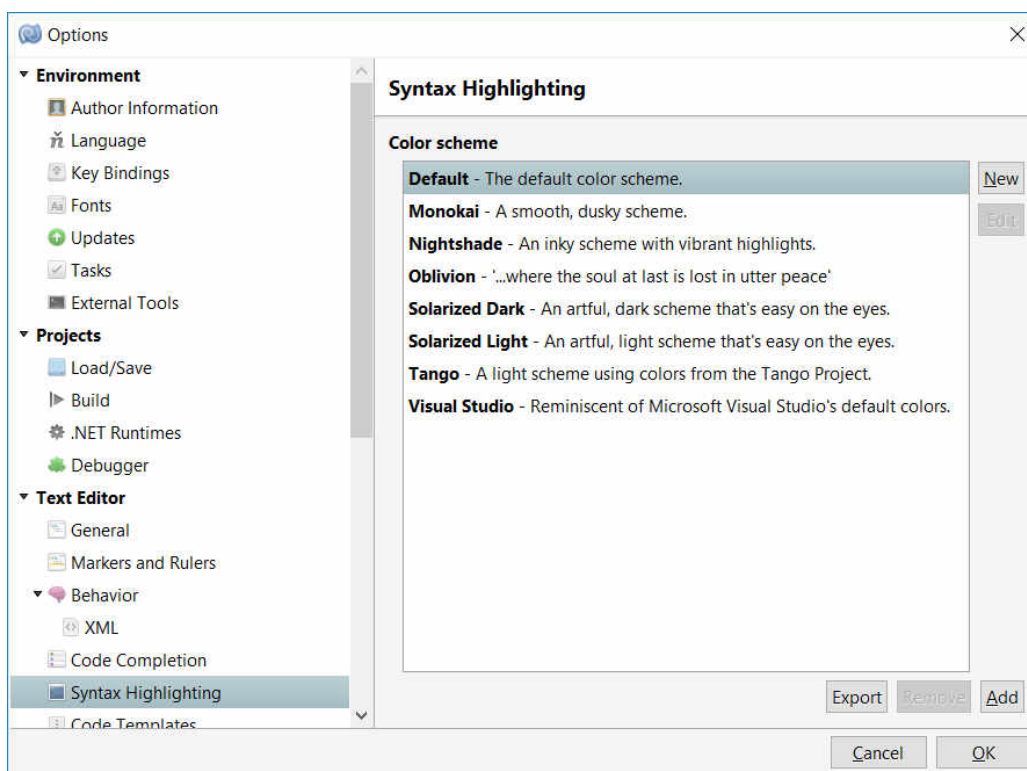


Рисунок 3.12 - Вікно **Options**. Розділ **Syntax Highlighting**

Source Analysis. Включає аналіз вихідного коду (рис. 3.13).

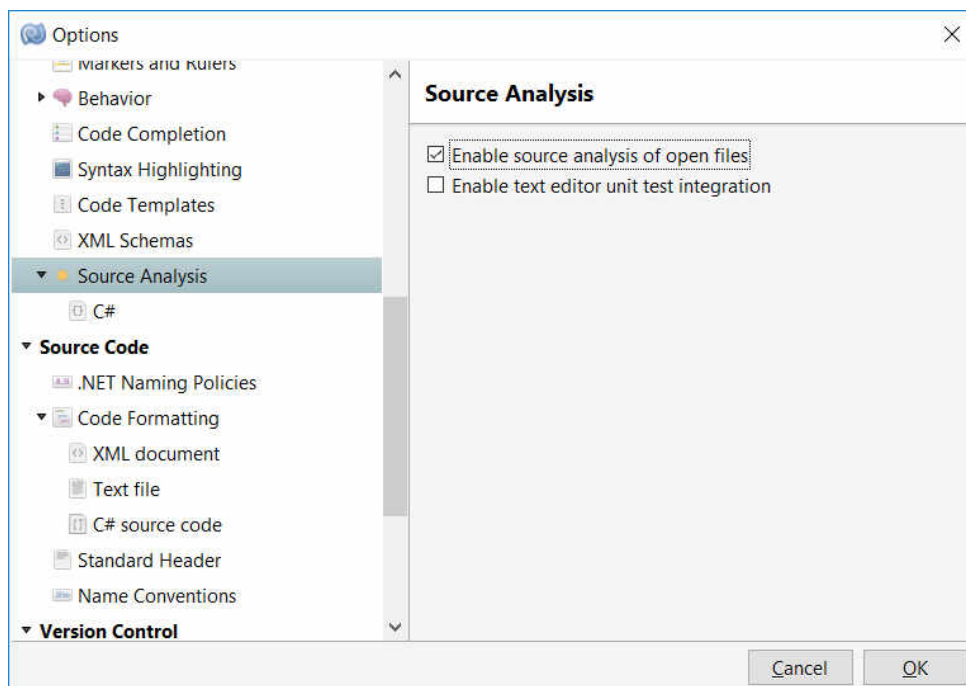


Рисунок 3.13 - Вікно **Options**. Розділ **Source Analysis**

Створення проекту в MonoDevelop

Для створення нового проекту в меню «**File**» вибираємо пункт «**New**→**Workspace**». Відкриється майстер проекту (рис. 3.14).

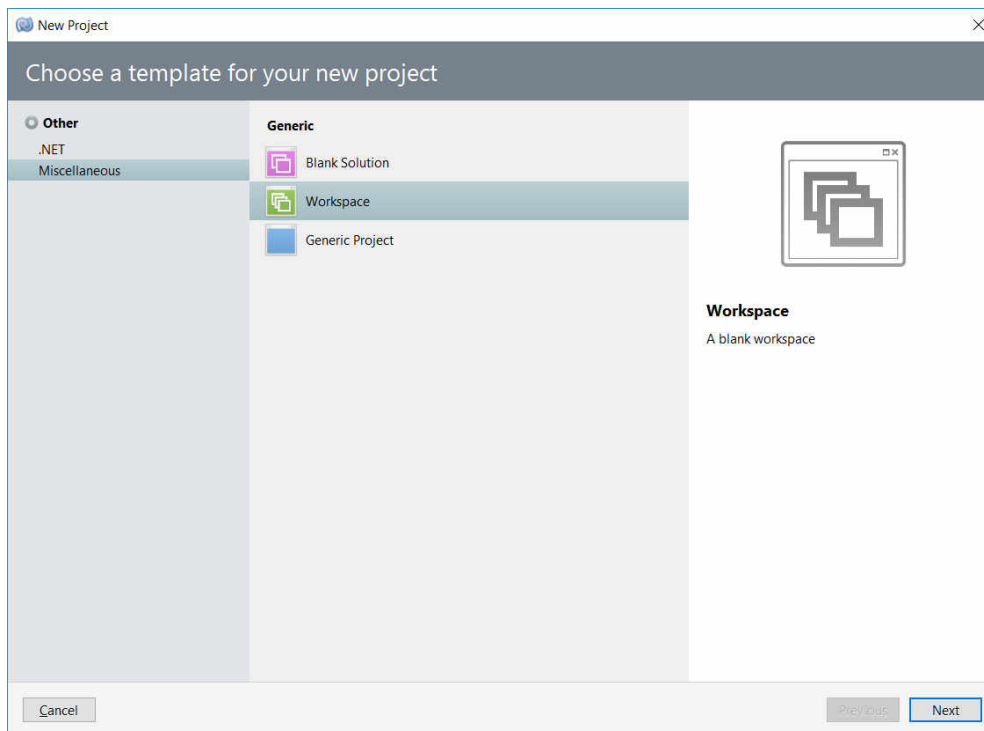


Рисунок 3.14 - Майстер проекту

У текстовому полі **Project Name** вводимо ім'я проекту, в текстове поле **Solution Name** - ім'я рішення (ці два пункти в даному випадку збігаються, а можуть і не збігатися), а в **Location** вказуємо місце розташування проекту.

Галочка під **Location** позначає створення внутрішньої папки з ім'ям рішення, куди будуть поміщені всі оригінали - рекомендується залишити цю позначку як є, але якщо хочеться іншого варіанту, то можете сміливо діяти.

Інші пункти призначені для управління контролем версій (рис. 3.15).

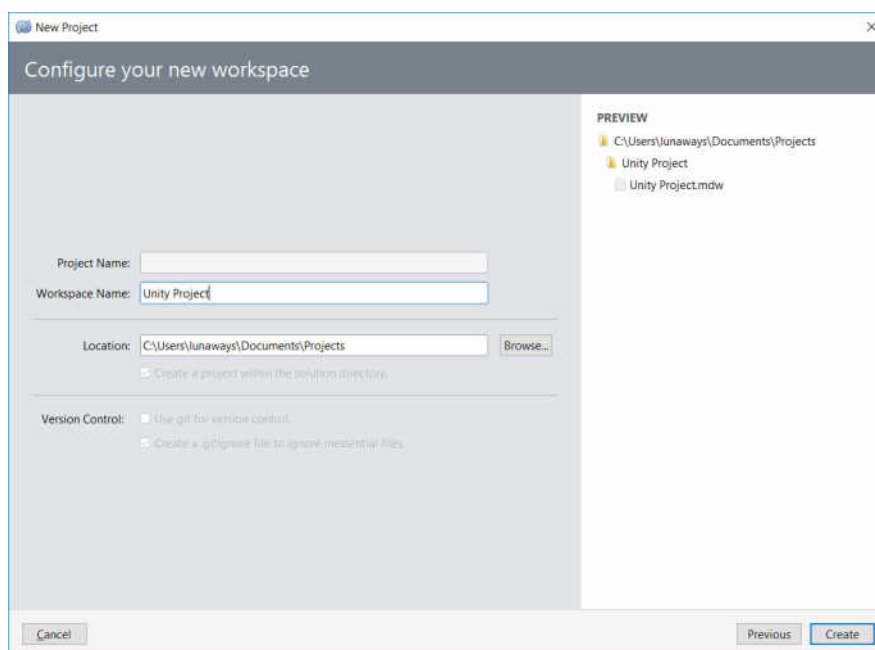


Рисунок 3.15 - Створення проекту

Натискаємо кнопку **Create**, після чого отримуємо порожню папку проекту (рис. 3.16).

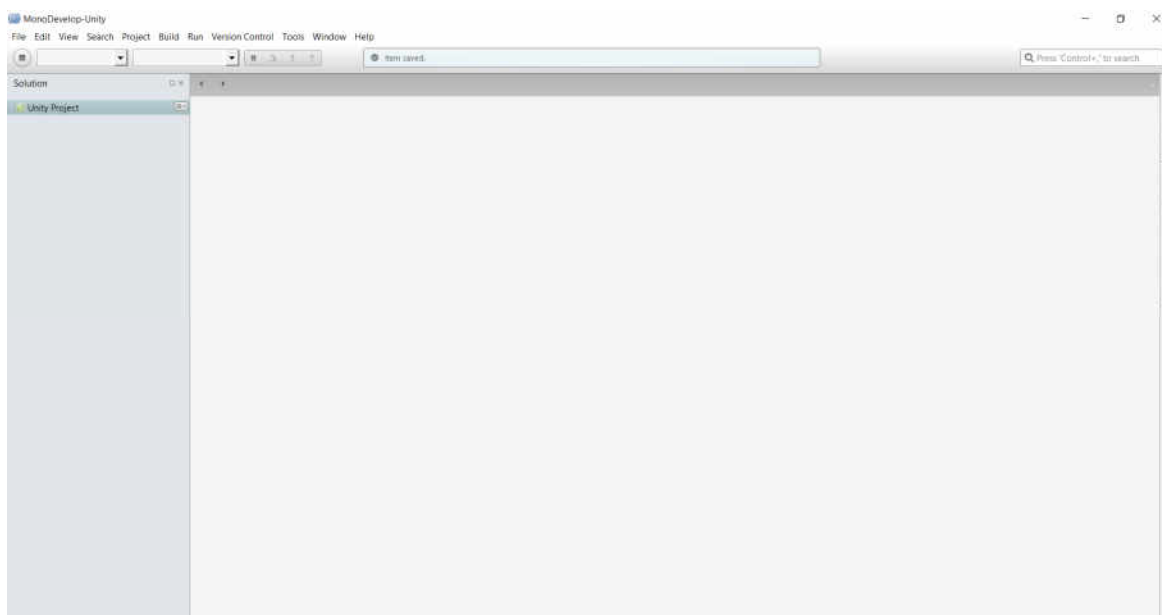


Рисунок 3.16 - Папка проекту

Створимо новий C# - файл за допомогою меню **File**→**New**→**File** (рис. 3.17).

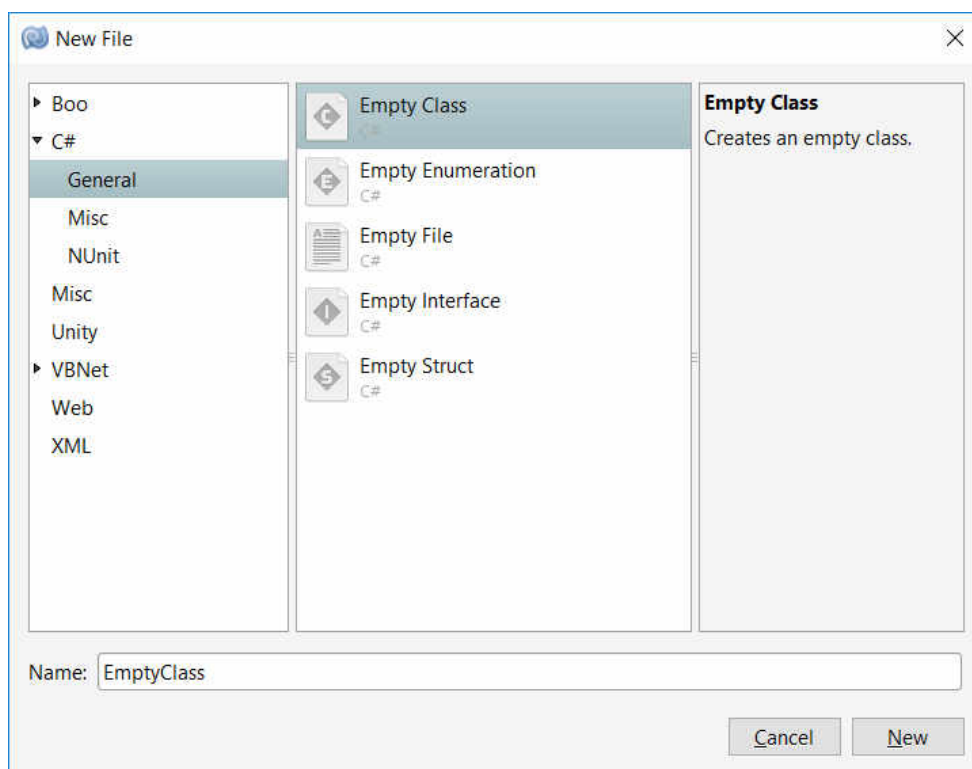


Рисунок 3.17 - Створення нового C# - файлу

Отримуємо новий файл, який є частиною проекту і який вже відкритий на редагування.

Налагодження вихідного коду

Відкриті файли відображаються в **MonoDevelop** як вкладки і можуть бути відредаговані в стандартному текстовому редакторі. Зліва від панелі редактора знаходиться сіра шкала точок останова (**breakpoint bar**). Клік ПКМ на цій шкалі додає маркер точок переривання (**breakpoint**) напроти поточного рядка коду.

Додавання точки переривання на шкалу повідомляє **Unity**, що після досягнення цієї лінії в режимі відтворення слід призупинити виконання сценарію. Коли скрипт так «заморожується», ви можете використовувати **Debugger** для визначення того, що саме робить скрипт (рис. 3.18).

При припиненні виконання на точці переривання, у вкладках в нижній частині вікна **MonoDevelop** відображається інформація про поточний стан виконання. Найбільш важливою з вкладок є **Locals**.

У цій вкладці показані значення локальних змінних в поточній функції (псевдо-локальна змінна з ім'ям **this** автоматично доступна в кожній функції, хоч вона і не оголошена явно; це посилання на поточний екземпляр скрипта, так за допомогою «**this**» можна отримати доступ до всіх змінних, що оголошені в скрипті).

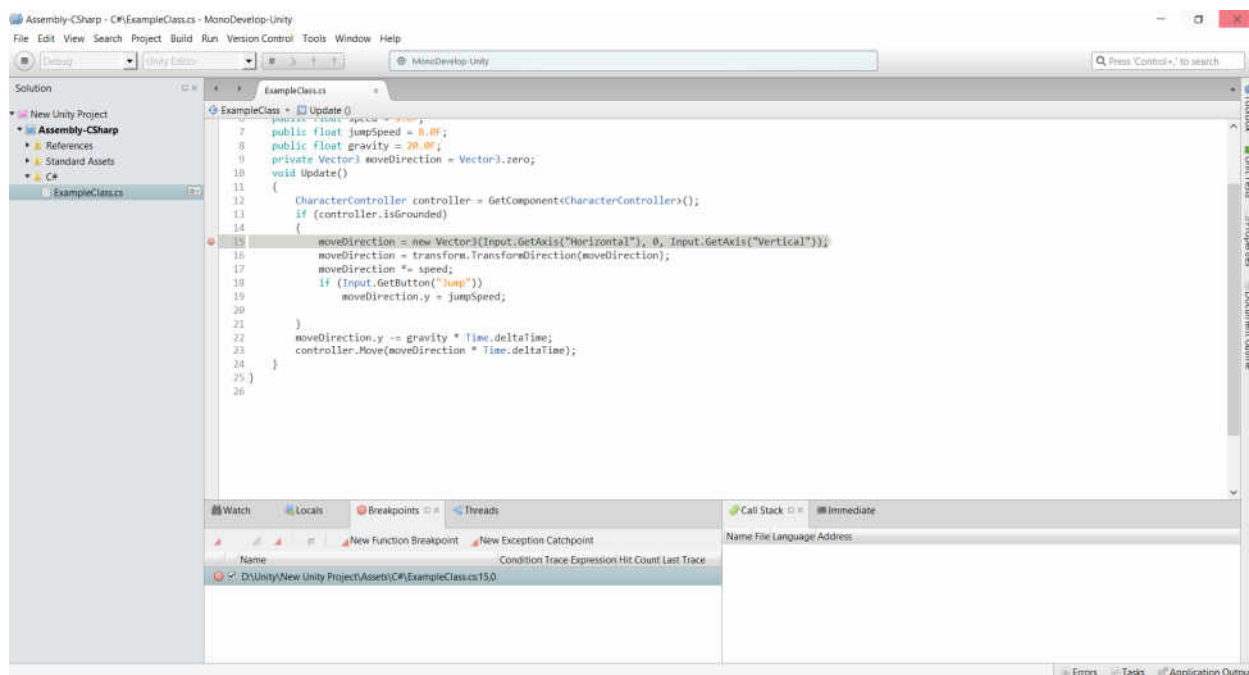


Рисунок 3.18 - Додавання точки переривання

Ви можете використовувати точки переривання разом з вкладкою **Locals** для отримання ефекту, аналогічного додаванню виразів **print** в ваш код - ви можете переглядати значення змінних в будь-який зручний вам момент часу.

У вкладці **Locals** ви також можете редагувати значення змінних. Це може виявитися корисним в разі, якщо ви знайшли невірно встановлену змінну і хочете перевірити, чи пропаде ця проблема при установці правильного значення.

Інша корисна функція MonoDevelop - послідовне виконання. Коли виконання призупиниться на точці з переривання, стане доступна шкала інструментів налаштування у верхній частині вікна **MonoDevelop**.



Рисунок 3.19 - Інструменти покрокового налагодження MonoDevelop

Перші чотири кнопки це **Continue** (продовжити), **Step Over** (крок через), **Step In** (крок всередину) і **Step Out** (крок назовні), які також можуть спрацьовувати по командам з меню **Run** (Праву кнопку, **Detach** може бути використано для закінчення сесії). **Continue** продовжує виконання до чергової точки переривання. **Step Over** і **Step In** виконують один рядок коду в один момент часу. Відмінності між ними в тому, що **Step Over** повністю виконує будь-які функції за одну лінію, а **Step In** дозволяє «зробити крок» всередину функції.

Застосування MonoDevelop при створенні ігрових додатків

Продемонструємо застосування MonoDevelop при виявленні зіткнень (collisions) між об'єктами.

Для цього створимо нову сцену. Додамо об'єкт «Plane», об'єкти «Cube» «Sphere» (рис. 3.20).

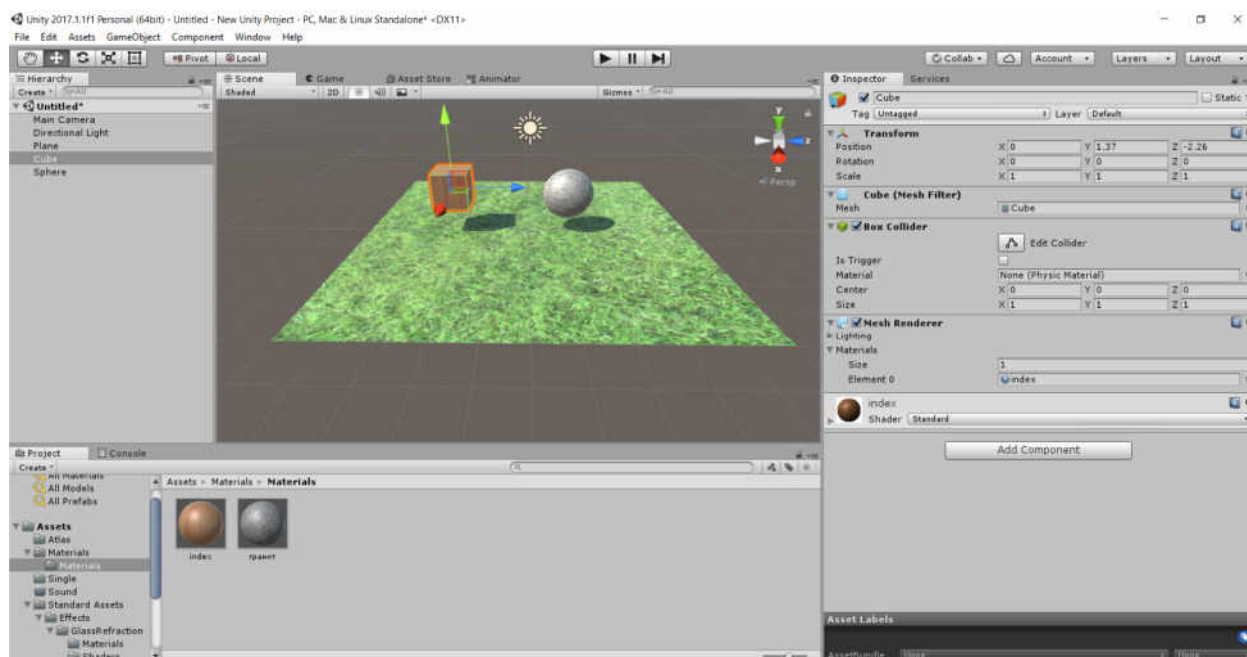


Рисунок 3.20 – Нова сцена з об'єктами «Plane», «Cube» та «Sphere»

У **Project View** за допомогою контекстного меню «Create→ C# Script» створіть новий файл C# - скрипта. Назвіть його «Collisions». Двічі натисніть ПКМ на файлі скрипта і ви відкриєте його в редакторі скриптів **MonoDevelop** (рис. 3.21).

Зверніть увагу, що назва класу завжди має збігатися з назвою скрипта (реєстр букв теж має значення):

```
public class Collisions : MonoBehaviour {  
    // Use this for initialization  
    void Start() {  
    }  
    // Update is called once per frame  
    void Update() {  
  
    }  
}
```

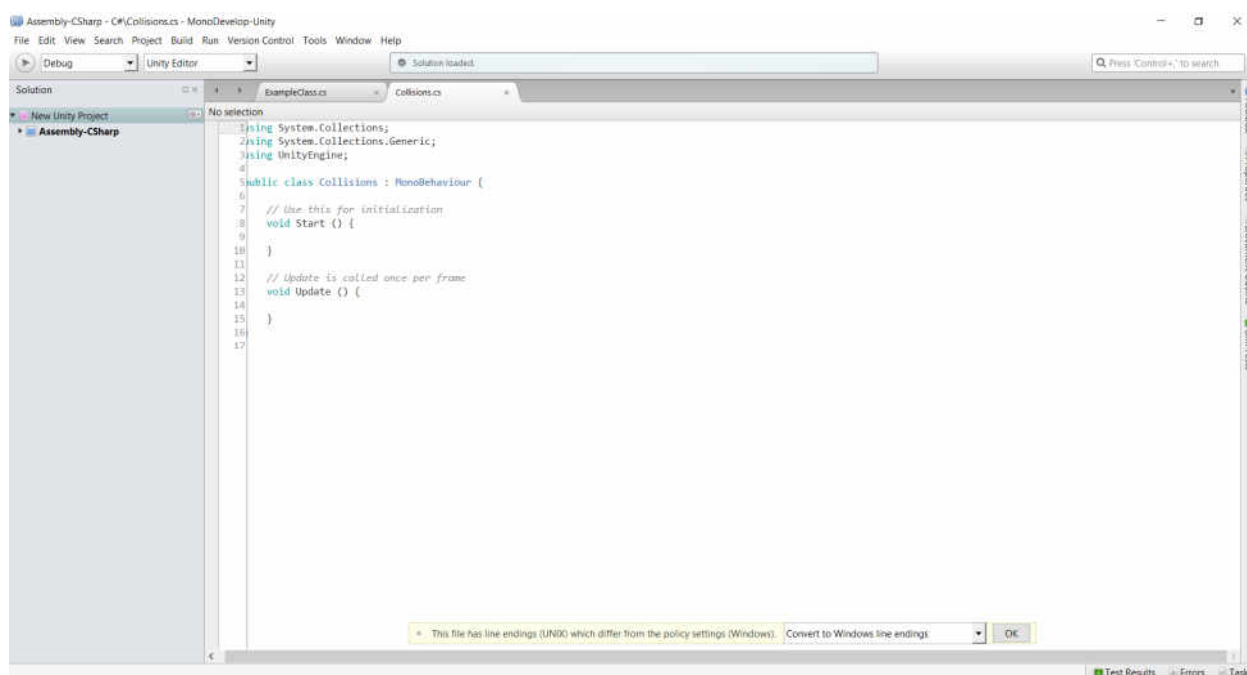


Рисунок 3.21 – C# - скрипт «Collisions» в редакторі скриптів MonoDevelop

За замовчуванням Unity створює каркас для C# -скрипта. Видаліть з тіла класу «Collisions» весь вміст і замініть наступним:

```
// Метод, який викликається при зіткненні об'єкта  
void OnCollisionEnter() {  
    Debug.Log("Hit Something");// Передаємо повідомлення в консоль Unity  
}
```

Функція **OnCollisionEnter** визначає зіткнення об'єкта з іншими об'єктами. А статичний метод **Log** класу **Debug** пише повідомлення в консоль **Unity**. Після збереження скрипта, додайте його як компонент для падаючого куба.

Тепер якщо ви натиснете **Play**, в момент коли куб торкнеться підлоги консоль видасть відповідне повідомлення. Відкрити вікно консолі можна комбінацією **ctrl + shift + c** в Windows. Останнє консольне повідомлення відображається в статус барі проекту (внизу вікна).

Причому повідомлення буде видаватися при кожному зіткненні.

Для того, щоб зрозуміти з яким саме об'єктом зіткнувся куб, ми будемо брати значення параметра класу **Collision**, яке буде приймати метод **OnCollisionEnter**:

```
// Тепер метод приймає об'єкт класу Collision,  
// з яким відбувається зіткнення  
void OnCollisionEnter(Collision myCollision) {  
    // визначення зіткнення з двома об'єктами  
    if (myCollision.gameObject.name == "Floor") {  
        // Звертаємося до імені об'єкта з яким зіткнулися  
        Debug.Log("Hit the floor");  
    }  
    else if (myCollision.gameObject.name == "Wall") {  
        Debug.Log("Hit the wall");  
    }  
}  
}
```

Натисніть кнопку **Play** і після того як ваш куб торкнеться кілька разів сфери зупиніть гру і відкрийте консоль. Тепер при зіткненні буде видаватися відповідні повідомлення.

Завдання до лабораторної роботи №3

1. Завантажити проект Unity.
2. Створити нову сцену. Додати об'єкт «**Plane**», об'єкти «**Cube**» «**Sphere**».
3. У **Project View** створіть новий файл C# - скрипта. Назвіть його «Collisions».
4. Відкрийте C# - скрипт в редакторі скриптів **MonoDevelop**.
5. Відредагуйте C# - скрипт, додавши новий метод **void OnCollisionEnter**.
6. Натисніть кнопку **Play** та отримайте відповідне повідомлення в консолі.

Контрольні запитання

- Назвіть основні етапи процесу установки середовища розробки MonoDevelop.
- Назвіть основні етапи налаштування інтегрованого середовища розробки MonoDevelop в системі Unity 3D.
- Назвіть основні функції середовища розробки MonoDevelop.
- Як здійснюється синхронізація Unity проекту з проектом MonoDevelop?
- Для чого в MonoDevelop використовується шкала Breakpoint Bar?

ЛАБОРАТОРНА РОБОТА № 4

ТЕМА: «ІМПОРТ АУДІОКЛІПІВ В ІГРОВІ ДОДАТКИ. РОБОТА З ДИСПЕТЧЕРОМ AUDIOMANAGER В UNITY 3D»

Анотація

Лабораторна робота орієнтована на оволодіння студентами методами роботи зі звуком за допомогою диспетчера AudioManager в Unity 3D.

Мета лабораторної роботи

Сформувати у студентів знання щодо налаштування центрального диспетчера управління звуком AudioManager в Unity 3D, методів імпорту аудіокліпів в ігрові додатки.

Очікувані результати

У разі успішного виконання лабораторної роботи студент буде вміти створювати ігрові додатки зі звуковим супроводженням за допомогою диспетчера AudioManager.

Імпорт звукових ефектів

Для відтворення звуків потрібно імпортувати аудіофайли в Unity - проект.

Формати аудіофайлів, які підтримує **Unity** подано у табл. 4.1.

Таблиця 4.1 - Формати аудіофайлів, які підтримує Unity

Тип файла	Переваги та недоліки
WAVE, WAV	Waveform Audio File Format (WAVE, WAV, від англ. Waveform - «в формі хвилі») - формат файлу-контейнера для зберігання запису оцифрованого аудіопотоку, підвид RIFF. Цей контейнер, як правило, використовується для зберігання нестислого звуку в імпульсно-кодової модуляції. Однак контейнер не накладає жодних обмежень на використання алгоритму кодування.
AIFF	Audio Interchange File Format (AIFF) - формат аудіофайлів, який застосовується для зберігання звукових даних. AIFF був розроблений компанією Apple Computer на основі формату IFF компанії Electronic Arts і найчастіше використовується в комп'ютерах Apple Macintosh. Звукові дані в стандартному файлі формату AIFF представляють собою нестиснуту імпульсно-кодову

	<p>модуляцію. Також існує і стисла версія формату AIFF, яку називають AIFC (іноді AIFC-C), в якій для стиснення можуть бути використані різні кодеки.</p> <p>AIFF, поряд з CDA і WAV, є одним з форматів, що використовуються в професійних аудіо- та відеододатках, так як на відміну від більш популярного формату MP3, звук в AIFF кодується без втрат в якості. Як і будь-які стиснені файли, файли AIFF займають набагато більше дискового простору, ніж їх стислі аналоги.</p>
MP3	<p>MP3 - кодек третього рівня, розроблений командою MPEG, формат файлу для зберігання аудіоінформації. Формат був ліцензований, але 23 квітня 2017 року термін дії всіх патентів закінчився і ліцензійні збори припинилися.</p> <p>Формат MP3 використовує спектральні відсікання, згідно психоакустичної моделі. Звуковий сигнал розбивається на рівні по тривалості відрізки, кожен з яких після обробки упаковується в свій фрейм (кадр). Розкладання в спектр вимагає безперервності вхідного сигналу, в зв'язку з цим для розрахунків використовується також попередній і наступний фрейм. У звуковому сигналі є гармоніки з меншою амплітудою і гармоніки, що лежать поблизу більш інтенсивних - такі гармоніки відсікаються, так як середньостатистичне людське вухо не завжди зможе визначити наявність або відсутність таких гармонік. Така особливість слуху називається ефектом маскуванню. Також можлива заміна двох і більше сусідніх піків одним усередненим (що, як правило, і призводить до спотворення звуку).</p> <p>Критерій відсікання визначається вимогою до вихідного потоку. Оскільки весь спектр актуальний, високочастотні гармоніки не відсікаються, а тільки вибірково видаляються, щоб зменшити потік інформації за рахунок розрідження спектра. Після спектральної «зачистки» застосовуються математичні методи стиснення і упаковка у фрейми. Кожен фрейм може мати кілька контейнерів, що дозволяє зберігати інформацію про декілька потоків (лівий і правий канал або центральний канал і різниця каналів). Ступінь стиснення можна варіювати, в тому числі в межах одного фрейму. Інтервал можливих значень бітрейту складає 8-320 кбіт/с.</p> <p>Режими кодування і опції:</p> <p>Існує три версії MP3 формату для різних потреб: MPEG-1, MPEG-2 і MPEG-2.5. Відрізняються вони можливими діапазонами бітрейту і частоти дискретизації:</p>

	<p>32—320 кбіт/с при частотах дискретизації 32000 Гц, 44100 Гц і 48000 Гц для MPEG-1 Layer 3;</p> <p>16—160 кбіт/с при частотах дискретизації 16000 Гц, 22050 Гц і 24000 Гц для MPEG-2 Layer 3;</p> <p>8—160 кбіт/с при частотах дискретизації 8000 Гц і 11025 Гц для MPEG-2.5 Layer 3.</p>
OGG	<p>Ogg - відкритий стандарт формату мультимедіаконтейнера, який є основним файловим і потоковим форматом для мультимедіакодеків фонду Xiph.Org, а також назва проекту, що займається розробкою цього формату і кодеків для нього. Як і всі технології, що розробляються під егідою Xiph.Org, формат Ogg є відкритим і вільним стандартом, які не мають патентних чи ліцензійних обмежень.</p> <p>Ogg є всього лише контейнером. Звук або відео стискаються кодеками, а результат обробки зберігається в подібних контейнерах. Контейнери Ogg можуть зберігати потоки, закодовані декількома кодеками. Наприклад, файл з відео і звуком може містити дані, закодовані аудіо- та відеокодеками.</p> <p>У контейнері Ogg можна зберігати звук і відео в різних форматах (таких як MPEG-4, Dirac, MP3 та інші), але зазвичай Ogg використовується з наступними аудіокодеками:</p> <p>з втратами:</p> <p>Opus (раніше Harmony) - з низькою затримкою кодування (від 2,5 мс до 60 мс, налаштовується) і більш високою компресією аудіо, так само бітрейт від 6 до 510 кбіт/с;</p> <p>Speex - для стиснення мовного сигналу на низьких бітрейтах (~ 8-32 (кбіт / с)/канал);</p> <p>Vorbis - для стиснення звуку на середніх і високих бітрейтах (~ 16-500 (кбіт / с)/канал).</p> <p>без втрат.</p> <p>FLAC - для обробки звукових архівів та інших аудіо, що вимагають високої якості відтворення.</p>
MOD	<p>MOD - формат файлів, розроблений для створення, зберігання та відтворення музичних композицій на ПК Amiga. Свою назву отримав від того, що став першим форматом, що зберігає свої фрагменти (наприклад, семпли) в інших файлах (принцип модульності). Файли цього формату мають, як правило, розширення .mod.</p> <p>Кожен файл формату MOD містить в собі оцифровані записи реального звучання інструментів, так звані семпли.</p>

	<p>Композитор, який пише в форматі MOD, використовує програму, яка називається трекером, в якій вказує, який саме інструмент, в який час, якою нотою і якою з октав повинен прозвучати. Послідовність нот записується в список - трек, а кілька паралельно відтворюваних треків утворюють блок, який називається патерном. Патерни, які створюються композитором отримують номери, після чого композитор може в довільній формі вказувати який патерн і коли повинен прозвучати. Сукупність патернів і утворює модуль - файл у форматі MOD.</p>
XM	<p>Формат XM – це розширений модуль MOD, тип аудіофайлу FastTracker 2, який було введено розробником - демогрупою Triton.</p> <p>XM-файл є мультисемплінгом за допомогою доступних інструментів з об'ємним сигналом в панорамній оболонці, а також стислою структурою. Формат розширив список доступних команд ефектів і каналів, додав 16-бітну підтримку і запропонував альтернативну таблицю частот для портаменто (спосіб виконання, при якому наступна нота не відразу береться точно (в звуко-висотному відношенні), а використовується плавний перехід до потрібної висоти від попередньої ноти).</p> <p>XM є основним форматом для більшості трекерної музики.</p>

Колекцію аудіофайлів необхідно імпортувати в Unity.

Для імпорту необхідних ресурсів (аудіофайлів) на вкладці **Project** в розділі **Assets** створюємо папку для зберігання аудіофайлів «**Sound**». Переходимо в папку, вибираємо меню «**Assets → Import NewAssets**» та імпортуємо необхідні аудіофайли (рис. 4.1). В нашому випадку це файл формату .mp3.

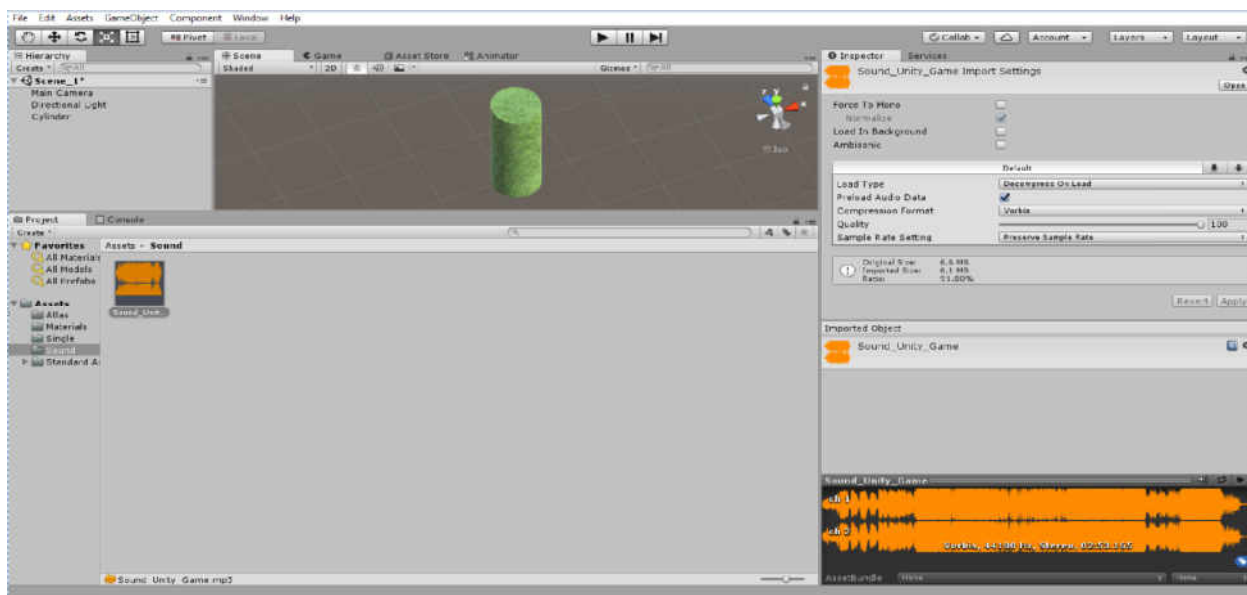


Рисунок 4.1 – Імпорт аудіофайлу

На панелі «**Inspector**» можна задати параметри імпорту аудіофайлів (рис. 4.2). Прапорець «**Force To Mono**» встановлювати не потрібно. Він дозволяє вибрати між моно- і стереозвуком. Найчастіше звук записується в стерео, тобто в файлі присутні одночасно дві хвилі, одна для лівого, інша для правого навушника або колонки. Щоб зменшити розмір файлу, можна прибрати половину звукової інформації.

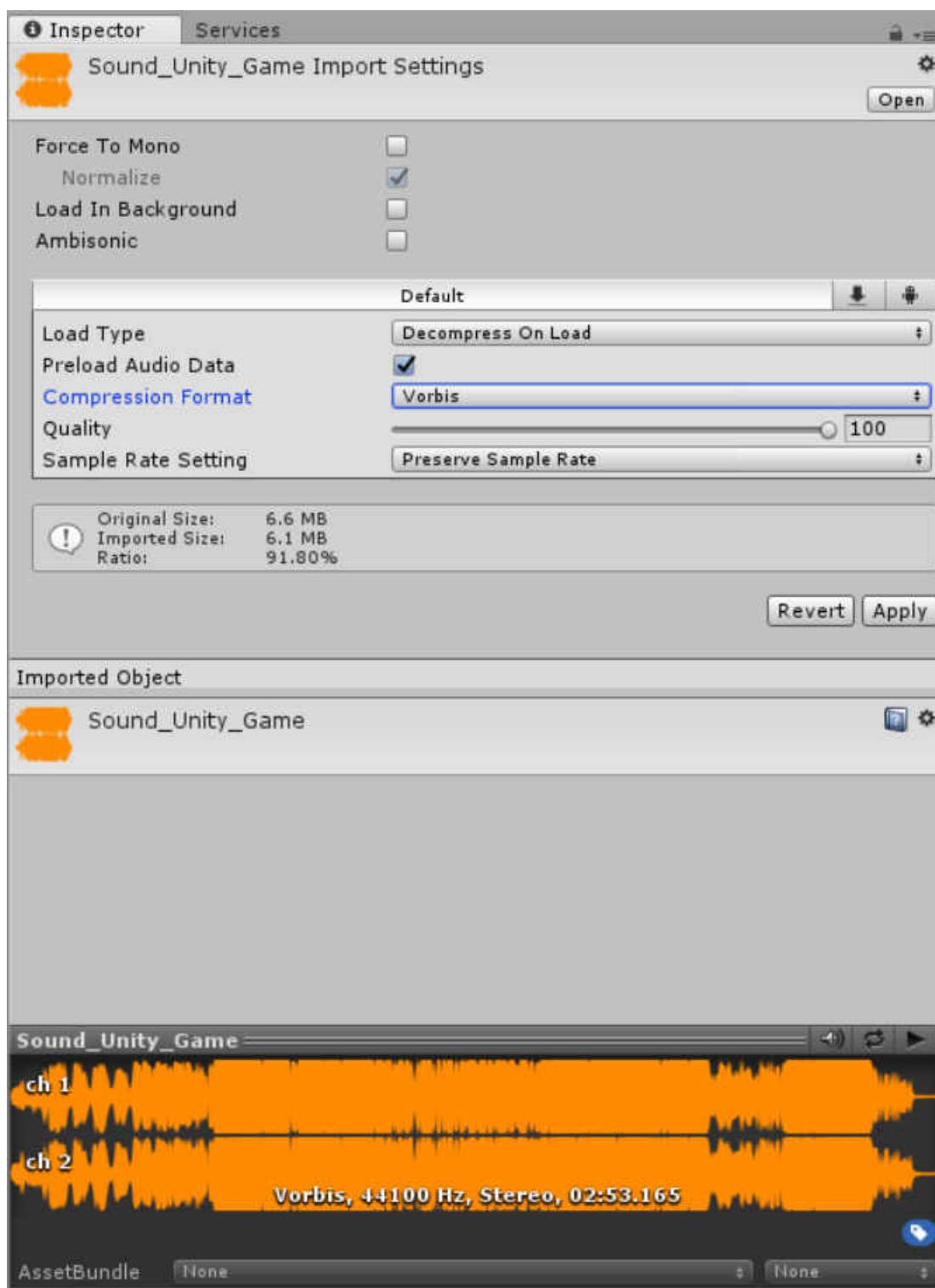


Рисунок 4.2 – Параметри імпорту аудіофайлу

Після цього на обидві колонки буде надсилатися одна хвиля замість двох.

Прапорець «**LoadInBackground**» дозволяє завантаження аудіофайлів у фоновому режимі.

Завантаження звуку у фоновому режимі дозволяє програмі продовжити свою роботу. Цей варіант в загальному випадку підходить для довгих музичних кліпів, але виключає можливість негайного відтворення звуку. Для коротких звукових кліпів прапорець «**LoadInBackground**» зазвичай не встановлюється, що гарантує їх повне завантаження перед відтворенням.

Прапорець «**Ambisonic**» вказує на те, що аудіофайл створено за технологією **Ambisonics**.

Технологія **Ambisonics** призначена перш за все для запису зі збереженням реальної звукової картини. При використанні багатоканального запису можлива імітація кругової (три канали) і сферичної (чотири канали) панорами. Існує матрична система UHJ, що дозволяє кодування зробленої за технологією **Ambisonics** багатоканального запису в два канали. Результат кодування сумісний зі стерео і моно. Поряд з двоканальним, можливо і багатоканальне відтворення записів, зроблених за технологією **Ambisonics**.

При використанні чотирьох каналів відтворення (тобто чотирьох незалежних акустичних систем) можна отримати кругову панораму. Для сферичної панорами необхідно шість каналів, можливе використання і більшої кількості акустичних систем.

Найбільш важливими параметрами є «**Load Type**» (тип, що завантажується) та «**Compression Format**» (формат стиснення). Останній відповідає за формат збережених звукової інформації.

Список «**Load Type**» дозволяє вказати, яким чином комп'ютер буде завантажувати дані з файлу. Параметр «**Vorbis**», вказує на те, що буде завантажено аудіоформат зі стисненням. Короткі звукові файли в стисненні не потребують, тому для них необхідно обрати варіант PCM (Pulse Code Modulation - імпульсно-кодова модуляція). Параметр ADPCM – є варіацією PCM та надає більш якісний звук.

5. Відтворення звукових ефектів

Для відтворення звуку в Unity необхідно додати три компоненти **AudioClip**, **AudioSource** та **AudioListener**.

Компонент **Audio Clip** містить дані аудіо, які використовуються в джерелах аудіо. Unity підтримує моно, стерео і мультіканальні звукові ресурси (до восьми каналів). Unity може імпортувати такі типи файлів: .aif, .wav, .mp3, .ogg. Також, Unity може імпортувати трекерні модулі з наступних типів файлів: .xm, .mod, .it, .s3m. Ресурси з трекерними модулями працюють так само, як і будь-які інші аудіо ресурси в Unity, щоправда для них недоступний перегляд форми сигналу в інспекторі імпорту.

Компонент **Audio Source** відтворює **Audio Clip** в сцені. Якщо **Audio Clip** є 3D кліпом, джерело програється в заданому положенні в просторі і буде приглушатися в залежності від відстані. Аудіо може бути розподілене по

колонках за допомогою властивості **Spread** і трансформуватися між 3D і 2D за допомогою властивості **PanLevel**. Можна контролювати залежність цих ефектів від відстані за допомогою кривих загасання. Для збагачення аудіо ряду, до джерела можна застосовувати окремі аудіо фільтри.

Властивості компоненту **Audio Source** подано у табл. 4.2.

Таблиця 4.2 - Властивості компоненту **Audio Source**

Властивість	Функція
Audio Clip	Посилання на аудіо кліп для програвання цим джерелом.
Output	Звук можна виводити через Audio Listener або звуковий мікшер.
Mute	Якщо обрана ця властивість, звук все ще буде програватися, але не буде звучати.
Bypass Effects	Застосовується для швидкого пропуску всіх звукових ефектів, застосованих до джерела звуку. Простий шлях включення/відключення всіх звукових ефектів.
Bypass Listener Effects	Застосовується для швидкого включення/відключення всіх звукових ефектів, застосованих до Audio Listener.
Bypass Reverb Zones	Застосовується для швидкого включення/відключення всіх зон реверберації.
Play On Awake	Якщо обрана ця властивість, звук почне відтворюватися одразу після завантаження сцени. Якщо властивість відключено, вам потрібно буде запустити звук програмно, за допомогою методу <code>Play ()</code> .
Loop	Застосовується для нескінченного повтору Audio Clip після його закінчення.
Priority	Визначає пріоритет даного джерела звуку серед всіх інших джерел у сцені (0 = найбільш важливий, 256 = найменш важливий, 128 за замовчуванням). Використовуйте 0 для музики, щоб уникнути її випадкового перемикання.
Volume	Визначає наскільки гучний звук на відстані однієї світової одиниці виміру (одного метра) від Audio Listener.
Pitch	Ступінь зміни висоти тону при уповільненні/прискоренні Audio Clip. Величина 1 означає нормальну швидкість відтворення.
Stereo Pan	Встановлює положення в стерео області 2D звуків.
Spatial Blend	Встановлює ступінь впливу 3D движка на джерело звуку.

Doppler Level	Визначає кількість ефекту Доплера, що застосовується до даного джерела (при значенні 0 ефект застосовуватися не буде).
Spread	Встановлює кут поширення для 3d стерео або мультिकанального звуку в просторі динаміків.
Min Distance	Гучність звуку буде максимальною, наскільки це можливо, протягом MinDistance. Поза MinDistance вона буде поступово знижуватися. Збільште MinDistance звуку, щоб зробити його «голосніше» в тривимірному світі, знизьте, щоб зробити звук «тихіше» в тривимірному світі.
Max Distance	Відстань, на якому звук перестає затухати. За межами цієї точки його гучність залишиться на рівні, на якому вона була б на відстані MaxDistance одиниць від слухача і більше не буде затухати.
Rolloff Mode	<p>Параметр визначає швидкість загасання звуку. Чим вище значення, там ближче повинен бути слухач до звуку перш, ніж його можна буде почути (визначається за графіком).</p> <ul style="list-style-type: none"> - Logarithmic Rolloff <p>Гучність звуку є досить високою, коли ви близько до джерела, але у разі віддалення від об'єкта вона досить швидко падає.</p> <ul style="list-style-type: none"> - Linear Rolloff <p>Чим далі ви від джерела звуку, тим гірше ви його чуєте.</p> <ul style="list-style-type: none"> - Custom Rolloff <p>Звук джерела аудіо поводиться відповідно до графіка загасання.</p>

Компонент **Audio Listener** поводитья як мікрофон. Він отримує вхідні дані з будь-якого джерела звуку (**Audio Source**) в сцені і програє звуки через динаміки. Для більшості додатків має сенс додавати **Audio Listener** до головної камери - об'єкту **Main Camera**. Більш того, до **Audio Listener** можна додавати аудіо ефекти, щоб застосувати їх до всіх чутних в сцені звуків.

У компонента **Audio Listener** немає властивостей. Його досить просто додати, щоб він запрацював. За замовчуванням, він завжди додається до об'єкту **Main Camera**.

Audio Listener працює в зв'язці з джерелами звуку (компонент **Audio Source**), дозволяючи вам створювати акустичне оточення в іграх. Якщо **Audio Listener** додано до **GameObject** в сцені, то будь-які досить близькі до слухача джерела будуть чутні в динаміках. У кожній сцені може бути тільки один **Audio Listener** для коректної роботи системи.

Якщо джерела у форматі 3D, тоді слухач буде імітувати положення, швидкість і орієнтацію звуку в 3D просторі. В 2D режимі буде ігноруватися будь-яка 3D обробка.

Відтворення аудіокліпу

Налаштуємо звук в Unity. **AudioClip** вже імпортовано, до камери додано компонент **Audio Listener** за замовчуванням. Залишилося додати тільки компонент **Audio Source**.

Audio Source виступає в ролі контролера, який запускає і зупиняє відтворення аудіокліпу.

Для створення нового джерела звуку **Audio Source** необхідно вибрати пункт меню «**GameObject**→**Create Empty**».

На панелі «**Hierarchy**» виділіть щойно створений **GameObject**. На панелі «**Inspector**» за допомогою кнопки «**Add Component**→**Audio**→**Audio Source**» додайте компонент **Audio Source** (рис. 4.3-4.4).

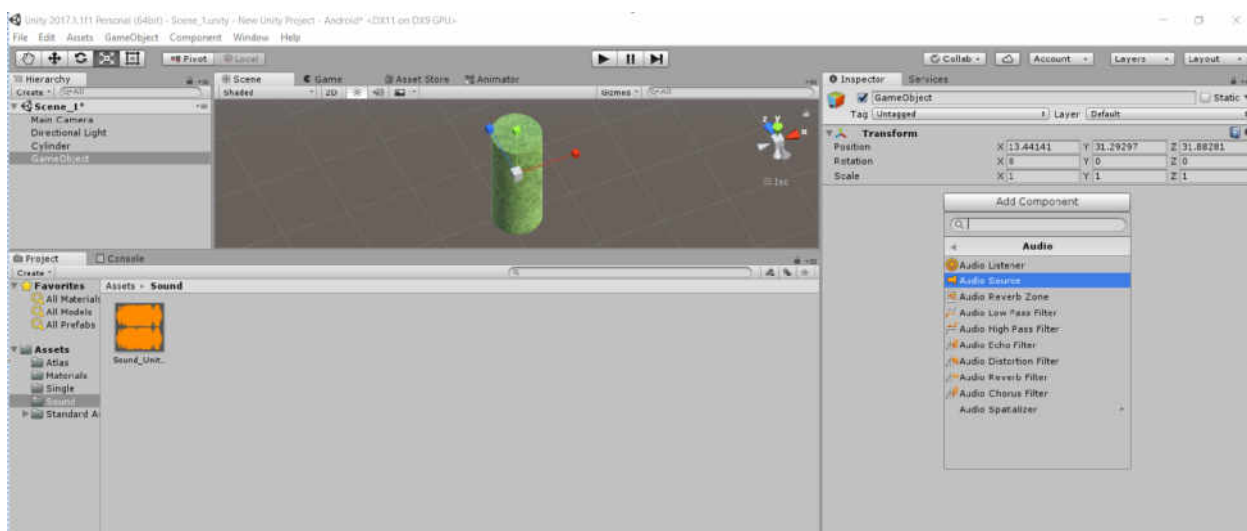


Рисунок 4.3 – Створення компоненту **Audio Source**

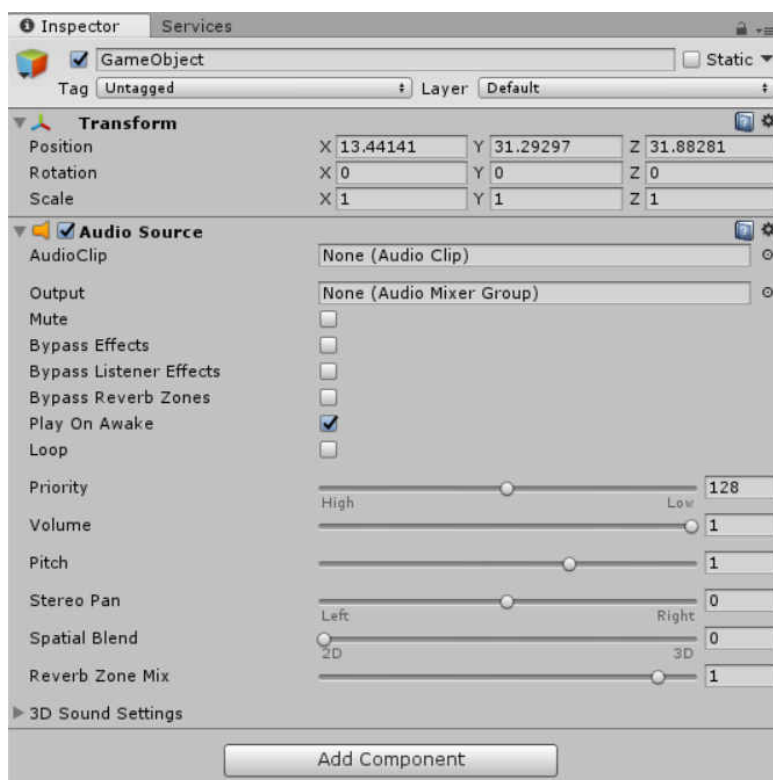


Рисунок 4.4 – Властивості компонента **Audio Source**

На панелі «Inspector» вкажіть властивість **AudioClip** компонента **Audio Source** (аудіокліп, який треба прослухати). Для цього необхідно перетягнути файл зі звуком зі вкладки **Project** на чарунку **AudioClip** панелі «Inspector».

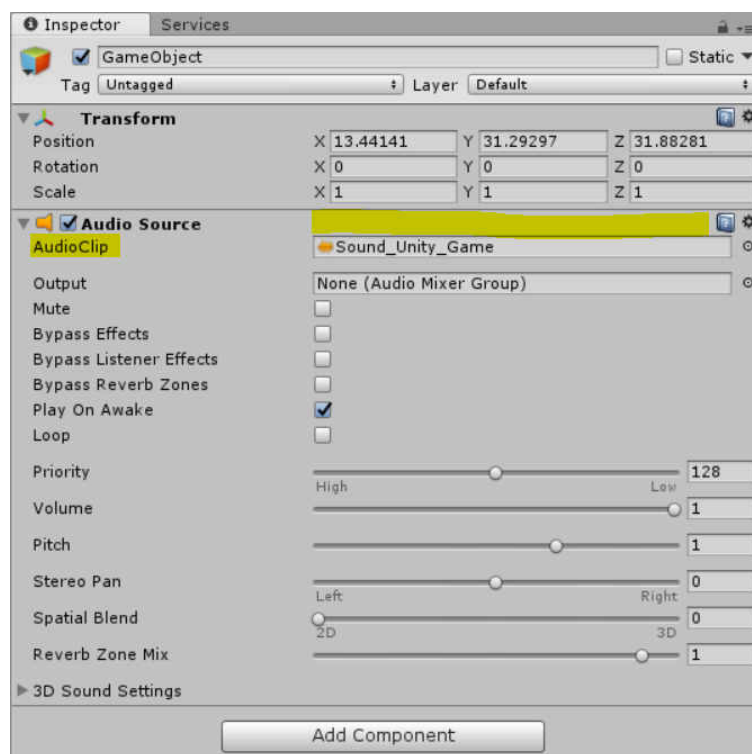


Рисунок 4.5 – Властивість **AudioClip** компонента **Audio Source**

Встановіть прапорці **Play On Awake** та **Loop**, обов'язково переконайтеся, що прапорець **Mute** знято.

Слід зауважити, що якщо ви хочете створити **Audio Source** тільки для одного компонента **Audio Clip**, який знаходиться в папці **Assets** вашого проекту, то ви можете просто перетягнути кліп у вікно сцени - **GameObject** з компонентом **Audio Source** для вашого кліпу буде створений автоматично. Перетягування кліпу на існуючий **GameObject** додасть до нього кліп разом з новим **Audio Source**, якщо цього компонента там ще немає. Якщо на об'єкті вже є компонент **Audio Source**, тоді новий кліп, який ви перетягнули, замінить той, який вже використовується джерелом звуку.

6. Робота з диспетчером **AudioManager**

Audio Manager (менеджер звуків) дозволяє налаштовувати максимальну гучність всіх звуків, які буде відтворено в сцені. Щоб відкрити цей менеджер, виберіть пункт меню «**Edit → Project Settings → Audio**» (рис. 4.6).

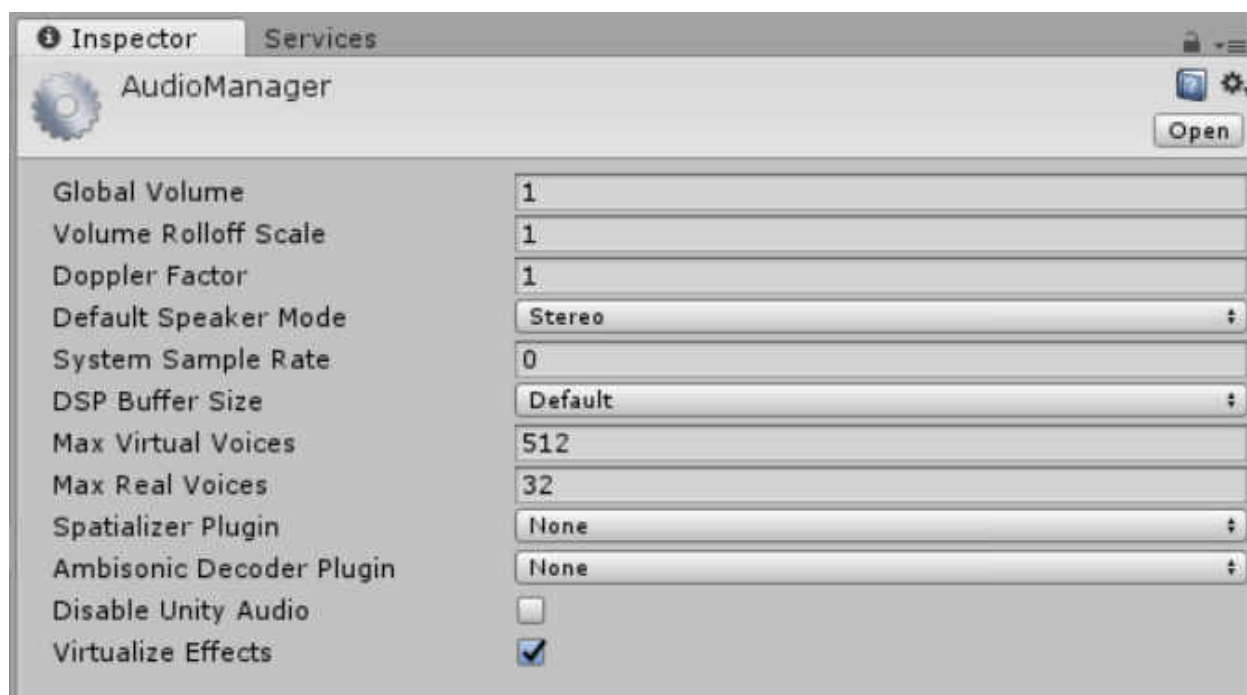


Рисунок 4.6 – Диспетчер **AudioManager**

Властивості диспетчера **AudioManager** подано у табл. 4.3.

Таблиця 4.3 - Властивості диспетчеру **AudioManager**

Властивість	Функція
Global Volume	Гучність всіх звуків.
Volume Rolloff Scale	Встановлює глобальний фактор ступеня загасання для логарифмічних згасаючих джерел звуку. Чим вище значення, тим швидше гучність буде затухати, і навпаки, чим менше значення, тим повільніше вона буде затухати (при значенні 1 буде імітовано «реальний світ»).
Doppler Factor	Ступінь чутності ефекту Доплера. Ефект відключений при значенні 0. 1 означає, що ефект буде досить добре чути для об'єктів з високою швидкістю пересування.
Default Speaker Mode	Визначає режим динаміків для вашого проекту. За замовчуванням має значення 2 - стерео колонки.
System Sample Rate	Вихідна частота вибірки. Якщо встановлено значення 0, буде використовуватися коефіцієнт вибірки системи. Також зауважте, що лише деякі платформи дозволяють змінити цей параметр, наприклад, iOS або Android.
DSP Buffer Size	Розмір DSP буфера можна встановити для оптимізації між латентністю і продуктивністю. Default Розмір буфера за замовчуванням Best Latency Краща латентність за рахунок зниження продуктивності Good Latency Баланс між латентністю і продуктивністю Best Performance Краща продуктивність за рахунок погіршення латентності.
Max Virtual Voices	Кількість віртуальних голосів, якими керує аудіосистема. Ця величина завжди повинна бути більшою, ніж кількість голосів у грі. Якщо ні, в консолі з'явиться попередження.
Max Real Voices	Кількість реальних голосів, які можна програвати одночасно. У кожному кадрі будуть вибрані найгучніші голоси.
Spatializer Plugin	Тільки для версії Unity Pro
Ambisonic Decoder Plugin	Тільки для версії Unity Pro

Disable Audio	Відключає аудіосистему в standalone збірках. Врахуйте, що це також впливає на звуки MovieTexture. У редакторі аудіо система все ще буде працювати і буде підтримувати попередній аудіо кліпів, але виклики AudioSource.Play і playOnAwake не будуть оброблятися, щоб імітувати поведінку аудіо системи standalone збірки.
Disable Unity Audio	Відключення звуку в Unity
Virtualize Effects	Віртуалізація звукових ефектів

Завдання до лабораторної роботи №4

1. Завантажити проект Unity.
2. Обрати необхідні аудіофайли.
3. Здійснити імпорт колекції аудіофайлів в проект Unity та задати параметри імпорту аудіофайлів.
4. Додати до проекту необхідні компоненти для роботи з аудіокліпами AudioClip, AudioSource та AudioListener.
5. Здійснити налаштування компоненту AudioSource.
6. Здійснити відтворення аудіокліпу.
7. Здійснити налаштування максимальної гучності всіх звуків, які буде відтворено в сцені за допомогою диспетчеру **AudioManager**.

Контрольні запитання

- Назвіть основні функції центрального диспетчера управління звуком AudioManager.
- Назвіть основні етапи імпорту аудіо-кліпів в ігрові додатки.
- В чому полягає відмінність 2D і 3D звуку?
- За допомогою якого компонента можна легко регулювати на глобальному рівні гучність звукових ефектів?
- Для чого в Unity 3D використовуються файли формату WAV?
- Назвіть необхідні компоненти для роботи з аудіокліпами.

ЛАБОРАТОРНА РОБОТА № 5

ТЕМА: «АНІМАЦІЯ СПРАЙТАМИ. РОБОТА З СИСТЕМОЮ АНІМАЦІЇ МЕСАНІМ»

Анотація

Лабораторна робота орієнтована на оволодіння студентами методами анімації за допомогою інструменту Mecanim, методами анімації спрайтами (окремі спрайти, атлас спрайтів).

Мета лабораторної роботи

Сформувати у студентів знання щодо методів розробки анімованих персонажів, технології анімації за допомогою інструменту Mecanim, методів анімації спрайтами (окремі спрайти, атлас спрайтів) в Unity 3D.

Очікувані результати

У разі успішного виконання лабораторної роботи студент буде вміти створювати ігрові додатки з анімованими персонажами, застосовувати методи анімації спрайтами при розробці ігрового контенту.

Анімація спрайтами

Імпорт та налаштування спрайтів. Спрайт (**sprite**) - це спеціальний 2D-об'єкт Unity. Він дозволяє відображати в сцені імпортовану плоску, регулярну текстуру. В іграх з горизонтальним слайдингом спрайти використовуються і для анімованих персонажів, і для реквізитів.

Застосування 2D-спрайтів можна проілюструвати на прикладі книги з рухомими картинками (flip-book), де на кожній сторінці зображений ескіз одного ключового кадру і читач, швидко гортаючи сторінки, спостерігає ефект анімації.

Є два способи імпортування та налаштування анімованих спрайтів:

Окремі спрайти

Якщо ви створили анімаційний спрайт, використовуючи окремі файли зображень для кожного кадру анімації, то для їх імпорту ви можете просто перетягнути їх всі відразу в панель проекту Unity.



Anim_01.png



Anim_02.png



Anim_03.png



Anim_04.png

Рисунок 5.1 - Анімаційні слайди з окремих файлів

При цьому кожен файл буде імпортовано як звичайну текстуру (рис. 5.2).

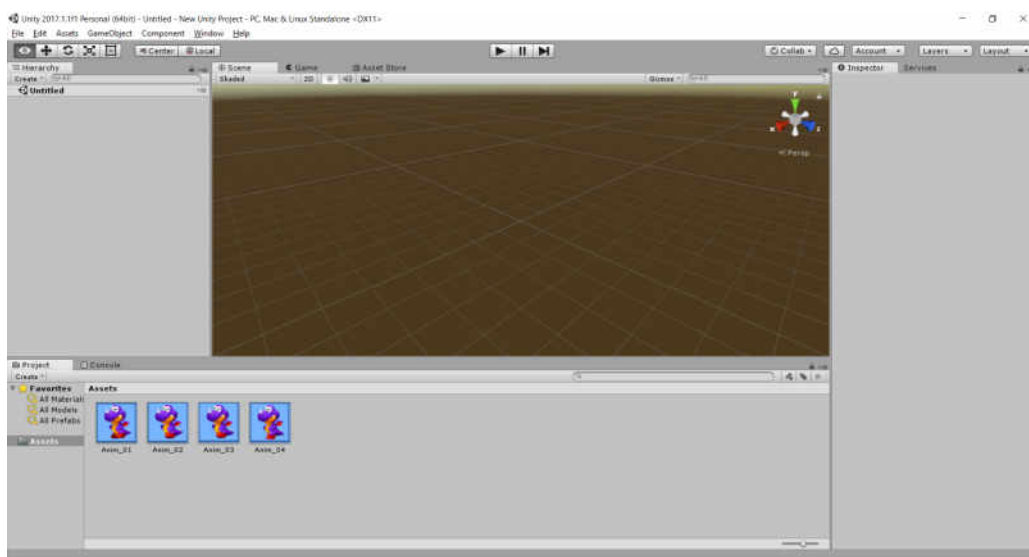


Рисунок 5.2 - Імпорт анімаційних слайдів з окремих файлів в панель проекту Unity

Для перетворення їх в спрайти виберіть всі імпортовані текстури в панелі проекту. У інспекторові об'єктів виберіть опцію **Sprite** для параметра **Texture Type**. Переконайтеся, що режим спрайтів **Sprite Mode** встановлений в **Single**, так як кожне окреме зображення є одним кадром для загального спрайту (одного і того ж персонажа).

Опція **Pivot** для спрайту повинна бути встановлена в **Bottom**. Такий вибір дозволить правильно розташувати спрайт в межах сцени. Після цього натисніть на кнопку **Apply** для підтвердження (рис. 5.3).

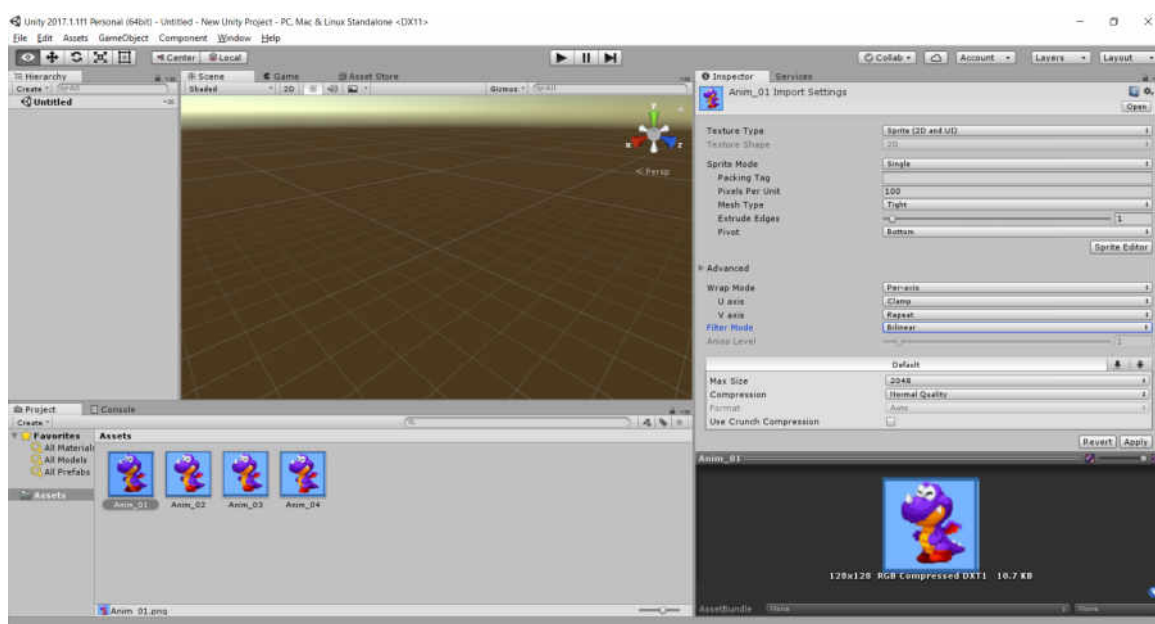


Рисунок 5.3 - Налаштування імпортованих текстур спрайтів

Імпортовані кадри тепер налаштовані і готові до анімації спрайтами в сцені.

Атлас спрайтів

Спрайти можуть бути зібрані разом на одному аркуші спрайтів, який називається текстурою атласу (Texture atlas). Іншими словами, всі кадри в анімації можуть бути поміщені в рядки і стовпці одного файлу текстури, а не розміщуватися в окремих файлах (рис.5.4).



Рисунок 5.4 - Текстура атласу спрайтів

Цей вид листів спрайтів також підтримується Unity, але він вимагає деяких додаткових налаштувань, в тому числі і на етапі імпорту.

Щоб імпортувати атлас спрайтів, просто перетягніть текстуру в панель проекту Unity. Атлас буде імпортовано як звичайний файл текстури. Після імпорту він з'явиться в панелі проекту (рис. 5.5).

Після імпорту ви повинні будете налаштувати текстуру для роботи в якості спрайту. Щоб зробити це, виберіть текстуру в панелі проекту. У інспекторі об'єктів виберіть **Sprite** для опції типу текстури **Texture Type**. Потім виберіть **Multiple** для опції режимів спрайту **Sprite Mode**, тому, що файл текстури містить кілька кадрів, а не один. Наприкінці натисніть на кнопку **Apply**.

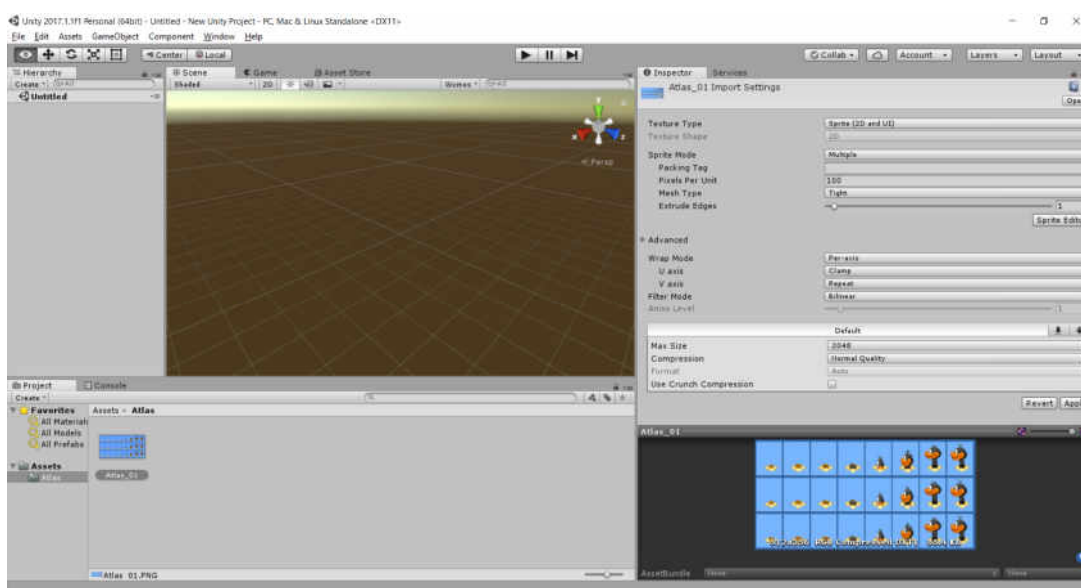


Рисунок 5.4 - Налаштування імпортованої текстури атласу спрайтів

Виберіть **Multiple** для опції **Sprite Mode**, давши зрозуміти Unity, що в текстурі присутні кілька кадрів, але Unity ще не знає, де саме вони знаходяться в текстурі. Щоб визначити це, можна використовувати редактор спрайтів (**Sprite Editor**). Цей інструмент доступний за допомогою кнопки **Sprite Editor** в інспекторі об'єктів, коли текстура обрана в панелі проекту (рис. 5.5 – 5.6).

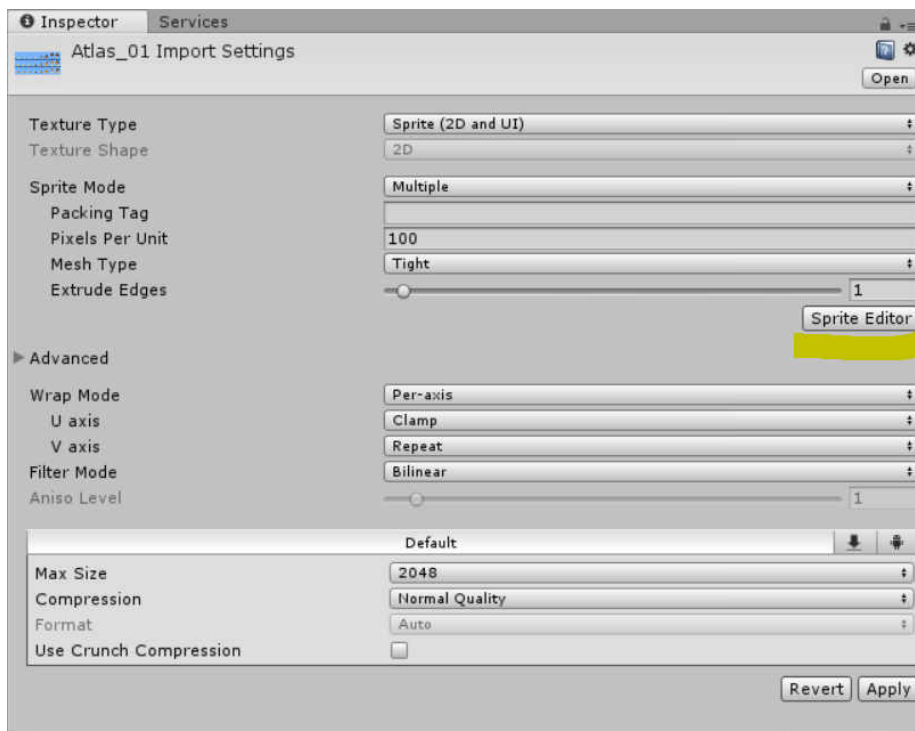


Рисунок 5.5 - Інспектор об'єктів з кнопкою **Sprite Editor**



Рисунок 5.6 - Редактор **Sprite Editor**

У діалоговому вікні редактора спрайтів **Sprite Editor** ви зможете вручну вказати кордони для кожного спрайту, просто перемагаючи рамку навколо кожного з них в зображенні.

Ви можете скористатися альтернативним підходом, автоматично згенерувати фрагменти спрайту, задавши розміри плитки. У текстурі з файлів на рис.5.6, кожен спрайт має розміри 62×80 . Для їх поділу натисніть на кнопку **Slice** в верхньому лівому кутку вікна редактора спрайтів **Sprite Editor** (рис. 5.7).

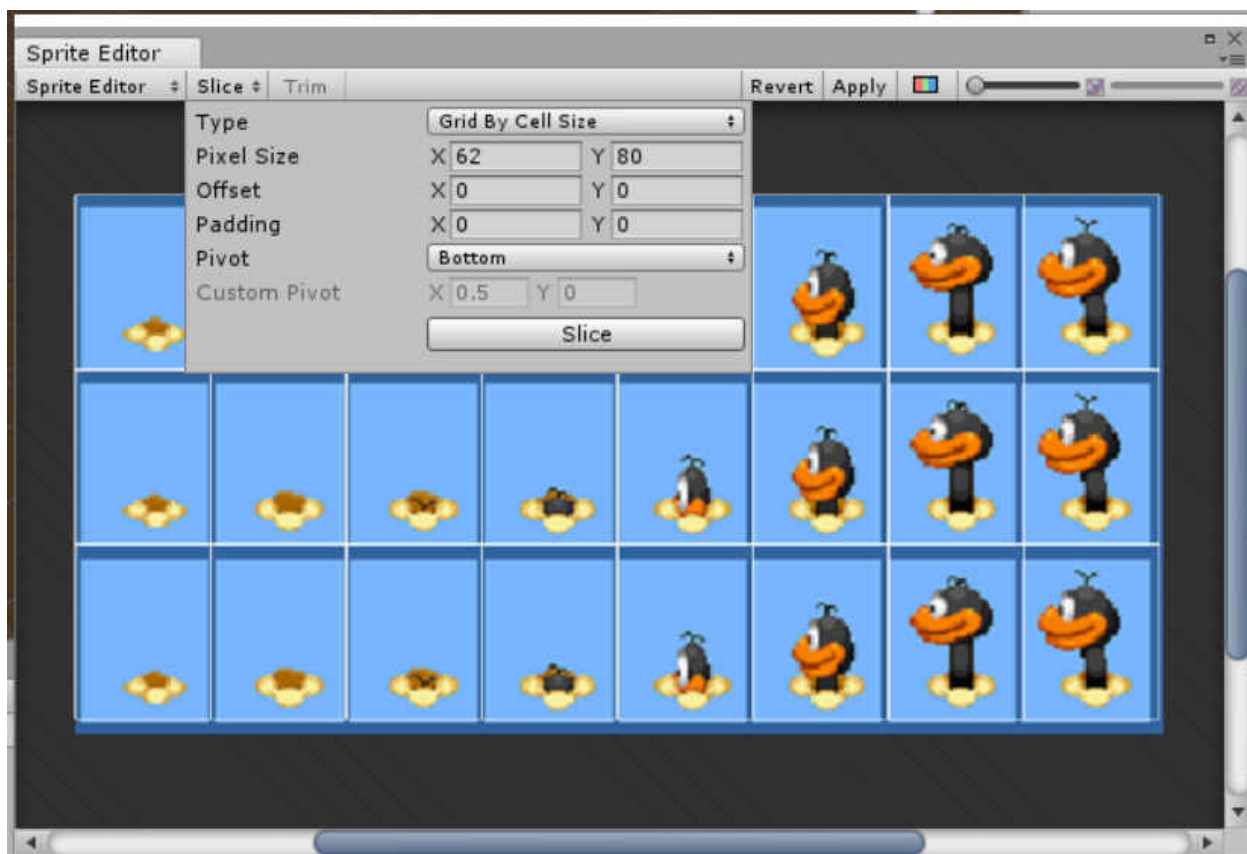


Рисунок 5.7 - Редактор **Sprite Editor**. Генерація слайдів

Натискання кнопки **Apply** в вікні редактора спрайтів призведе до генерації послідовності об'єктів спрайтів, які з'являться в панелі проекту в якості незалежних суб'єктів, готових до анімації (рис. 5.8).

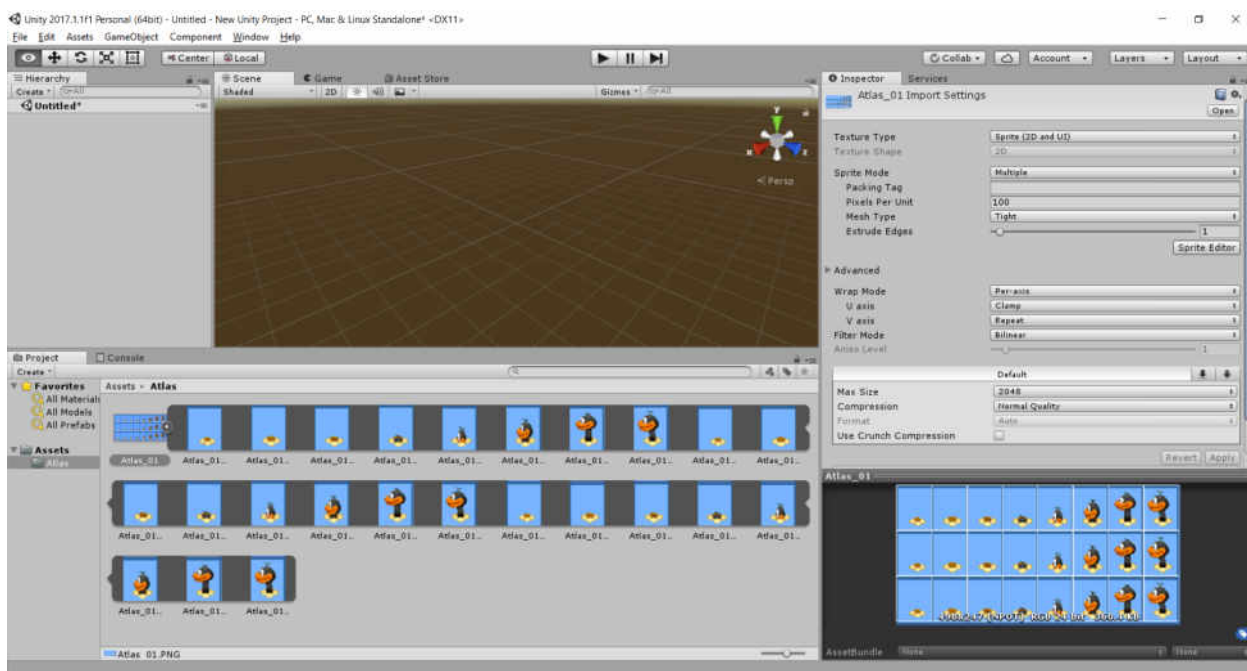


Рисунок 5.8 - Згенерована послідовність об'єктів спрайтів

Анімація за допомогою спрайтів

Після того як ви імпортували набір об'єктів спрайтів (не має значення, як окремі зображення або як текстуру атласу), ви готові запуснути анімацію з їх участю.

На панелі проекту Unity виберіть всі спрайти, що входять в повну послідовність анімації, а потім перетягніть їх всі разом на панель ієрархії сцени. Якщо ви зробите це, з'явиться діалогове вікно збереження **Save**, пропонуючи вам створити нову послідовність анімації (файл з розширенням **.anim**). Задайте ім'я файлу анімації (наприклад, **PlayerRun.anim**) і виберіть кнопку **Save** (рис. 5.9).

У сцені буде створено об'єкт спрайту, він повинен стати видимим на вкладках сцени і гри. Якщо ви протестуєте вашу сцену, то ваш персонаж буде анімаційним, ви повинні побачити всі зображення послідовності. Зазначимо, що Unity створило новий актив **Animation Clip (.anim)**, який визначає ключову послідовність кадрів. Крім того, Unity створило контролер **Mecanim** для запуску відтворення анімації при завантаженні рівня і для управління швидкістю відтворення, додало компонент аніматора до об'єкта спрайту в сцені, щоб зв'язати об'єкт з його даними по анімації (рис. 5.10).