

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки (ОТ)

Звіт до лабораторної роботи №1

з дисципліни:

«Розробка ігрових застосувань. Unity рішення»

на тему **«Дослідження базового патерну ігрового рушія Unity на прикладі двовимірної технології»**

Перевірив:

доцент каф. ІСТ

Катін П. Ю.

Виконав:

студент 4 курсу, гр. ПІ-93

Говоруха М.А. (варіант 2)

2022 р.

1.1 Завдання

Мета роботи: полягає у набутті знань, умінь та навичок з технології розроблення основ проекту з використанням обраної мови програмування у обраній парадигмі.

Надається досвід створення репозиторію у системі контролю версій.

Також лабораторна робота дає основні навички розробки з використанням IDE ігрового рушія. Дається можливість роботи з іншим типом IDE за вибором студента та поузгодженню з викладачем.

Для підготовки ЛР № 1 може бути використаний будь-який сучасний ПК на основі мікропроцесора AMD64 (Intel® 64) або ARM. Операційна система Windows, Linux або Mac OS (Macintosh Operating System).

Вхідні дані ЛР 1.

Прізвище студента; ім'я студента; шифр навчальної групи; скорочена назва факультету; скорочена назва університету. Порядковий номер у списку, що визначає варіант.

Вхідні дані ЛР 1.

Репозиторій на GitHub з проектом. У окремому файлі вказана вся первинна інформація, що обговорена у вхідних даних. На даному етапі репозиторій не є обов'язковим. Дозволяється тримати проект локально. У проекті реалізовані всі вимоги відповідно до завдання і варіантів. Проект має запускатися на машині студента і викладача. У разі наявності помилок проект не зараховується.

Завдання

Репозиторій у системі контролю версій. Створити проект 2D. Загальні вимоги.

Акаунт на GitHub, на даному етапі за бажанням. Репозиторій на GitHub з проектом.

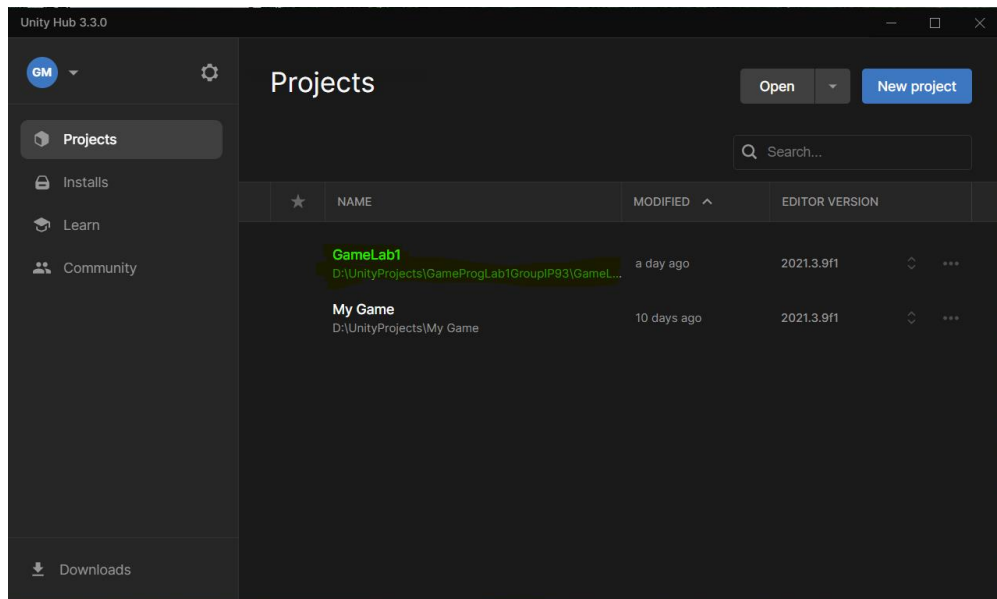
Назва GameProgLab1Group**Num**, де зафарбовано номер групи.

Установка ігрового рушія. Створений проект IDE (2D) на основі рушія, що містить 1 сцену, ігровий персонаж. Можуть бути включені інші елементи. Розроблений і налагоджений скрипт для управління ігровим персонажем. Достатньо продемонструвати рух ліворуч, праворуч, стрибки, коректну фізику, зупинку перед перешкодою. **Проект розташовано у репозиторій на GitHub**, основна мета полягає у дослідженні і підтвердженні володіння обраною IDE (2D) і технологією розподіленої системи контролю версій.

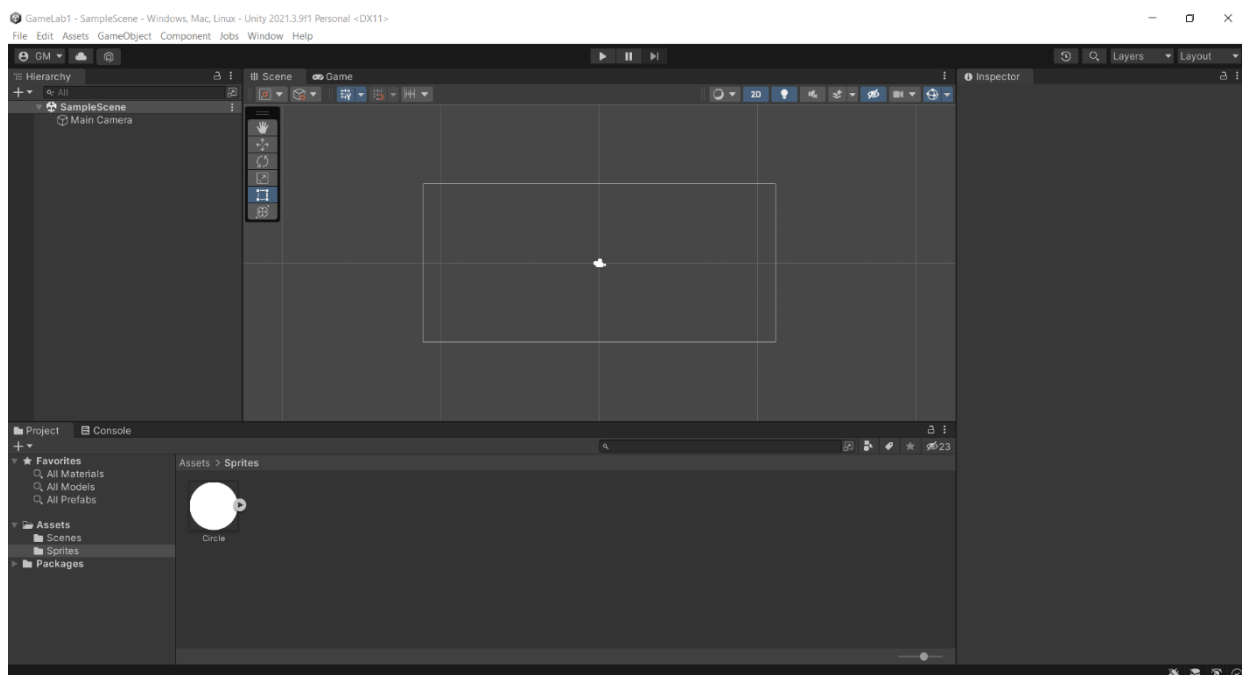
1.2 Хід виконання

Варіант 2: примітив – коло, асет - [Pixel Platformer Art Pack | 2D Environments | Unity Asset Store](#)

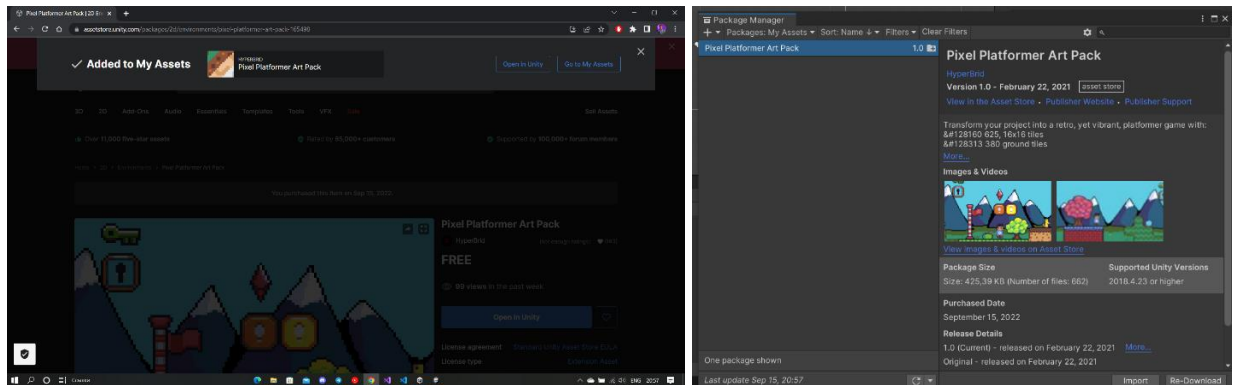
- Було створено 2D проект за допомогою UnityHub з назвою GameLab1, котрий поміщено у репозиторій GameProgLab1Group**IP93**:



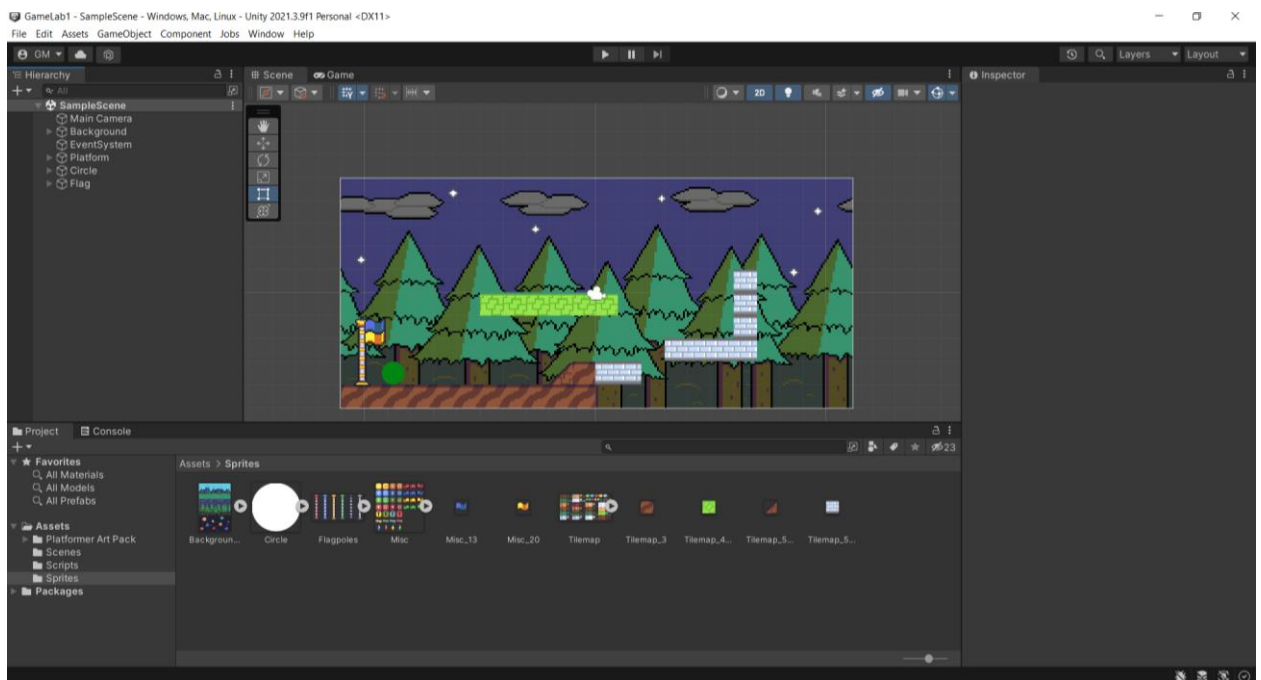
- Далі було створено каталог для спрайтів, і в ньому відповідно створено спрайт з примітивів згідно з варіантом завдання:



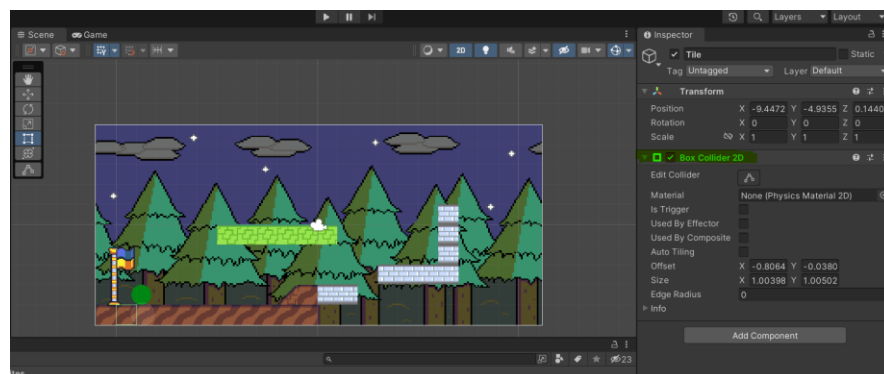
- Наступним кроком було обрано та завантажено відповідний до варіанту набір асетів:

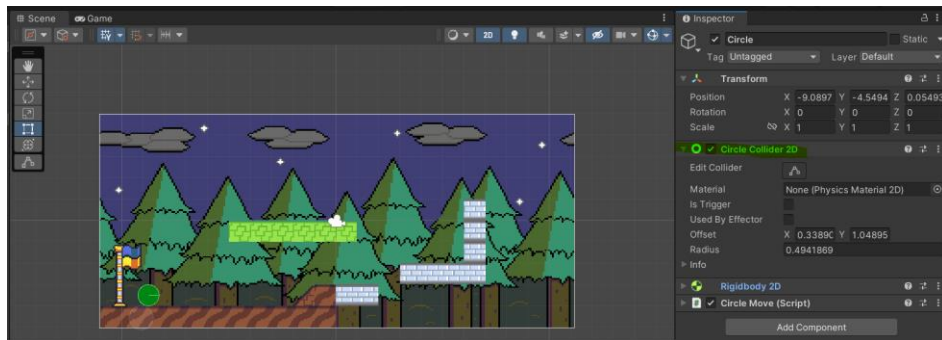


- Після здійснених вищенаведених підготовчих дій, було імпортовано набір асетів, та обрано з них спрайти для конструювання платформи гри, також було додано примітив у якості ігрового об'єкту та змінено його колір, результат наведено нижче:

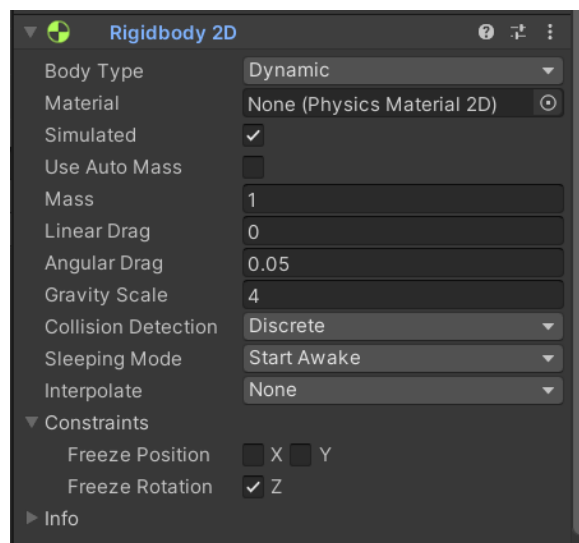


- До всіх об'єктів платформи було додано Box колайдер, а до ігрового об'єкту у результаті проведених порівнянь було прийнято рішення додати Circle колайдер, до декорацій (прапор) колайдерів додавати не потрібно, так як вони не взаємодіють з ігровим об'єктом:





- Також до ігрового об'єкту було додано клас Rigidbody2D, щоб надати об'єкту функціональність і поставити його під фізичний контроль, також я змінив деякі налаштування для більш реальної фізики (коеф. гравітації, заморозити рухи по осі z):



- Наступним кроком було розроблено базовий скрипт, який відповідає за рухи, стрибки, та умову коли ігровий об'єкт покидає ігрову зону, вихідних код скрипту наведено нижче:

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

public class CircleMove : MonoBehaviour
{
    private readonly float speed = 10f;
    private readonly float jumpForce = 15f;
    private readonly float lowerYPoint = -6f;
    private Vector3 startPos;
    private bool isGrounded;
    private Rigidbody2D rb;

    // Start is called before the first frame update
    void Start()
    {
        rb = GetComponent<Rigidbody2D>();
        startPos = new Vector3(transform.position.x, transform.position.y, transform.position.z);
    }

    private void OnCollisionEnter2D(Collision2D collision)
    {
        isGrounded = true;
    }

    // Update is called once per frame
    void Update()
    {
```

```

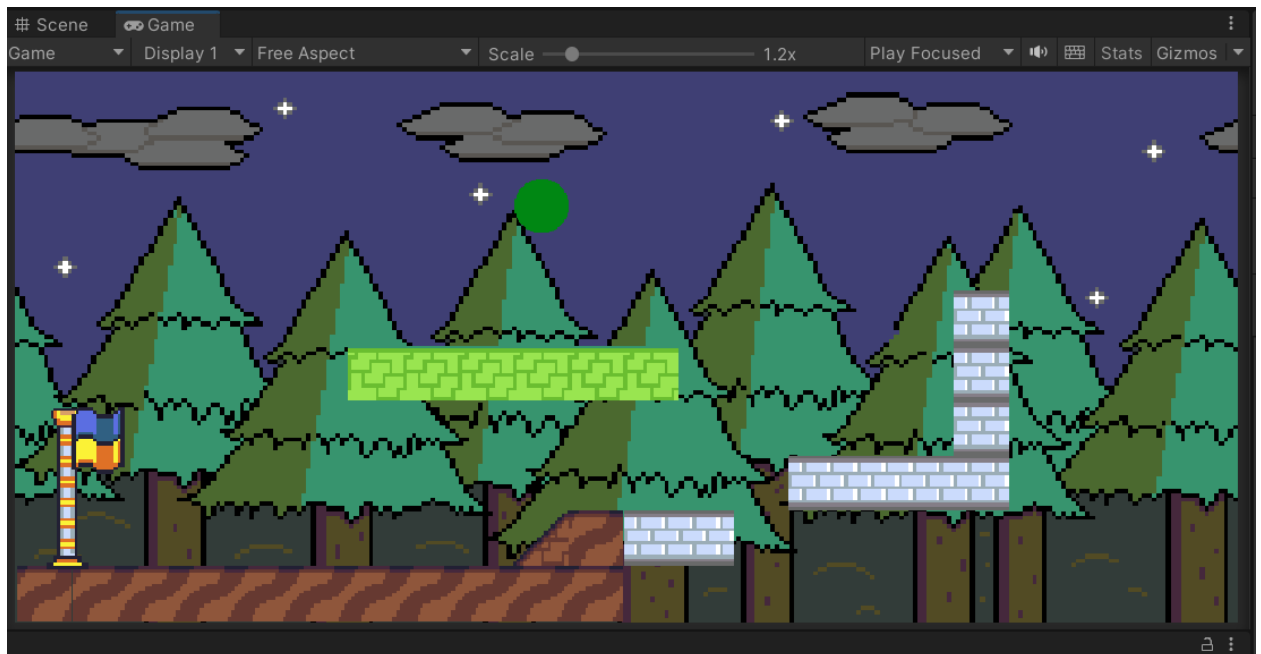
{
    if (Input.GetButtonDown("Jump") && isGrounded) Jump();
    if (Input.GetButton("Horizontal")) Move();
    if (transform.position.y < lowerYPoint) {
        transform.position = startPos;
    }
}

private void Move()
{
    Vector3 dir = transform.right * Input.GetAxis("Horizontal");
    transform.position = Vector3.MoveTowards(transform.position, transform.position + dir, speed *
Time.deltaTime);
}

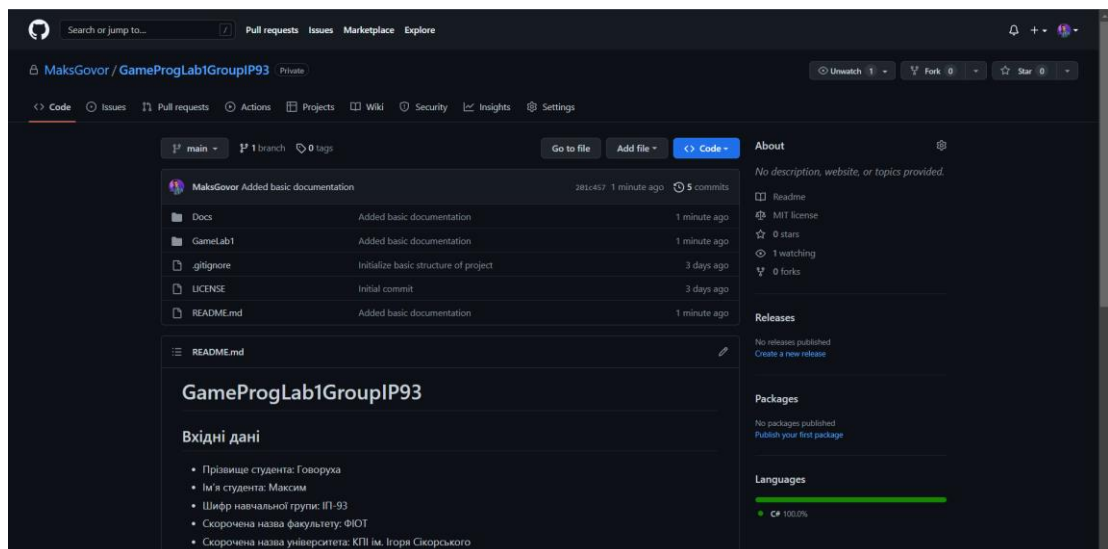
private void Jump()
{
    isGrounded = false;
    rb.AddForce(transform.up * jumpForce, ForceMode2D.Impulse);
}
}

```

- Тепер все готово для запуску, результат запуску наведено нижче:



- Під проєкт відповідно створено віддалений репозиторій GitHub і відповідно до вимог оформлено основну інформацію в README.md:



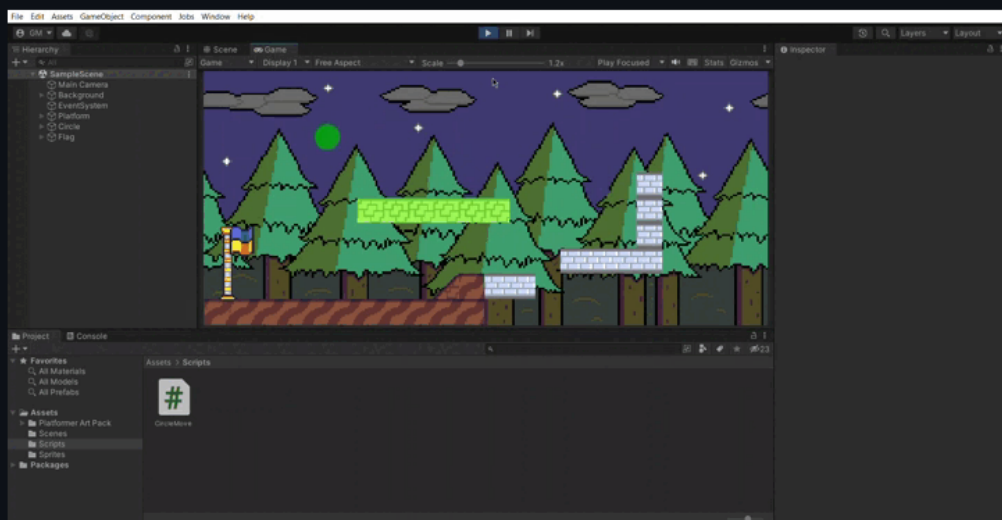
GameProgLab1GroupIP93

Вхідні дані

- Прізвище студента: Говоруха
- Ім'я студента: Максим
- Шифр навчальної групи: ІП-93
- Скорочена назва факультету: ФІОТ
- Скорочена назва університету: КПІ ім. Ігоря Сікорського

Документація

- [Звіт до лабораторної роботи](#)
- Результат виконання програми:



1.3 Висновки

За період виконання даної роботи я набув базових навичок розробки з використанням IDE ігрового рушія Unity, шляхом створення проекту (2D) на основі рушія, що містить 1 сцену та ігрового персонажа. Ігровий персонаж згідно завдання було створено як примітив, а інші об'єкти потрібно було взяти з assetstore, тому для цього було попередньо створено обліковий запис юніті за допомогою якого здійснено завантаження обраного асету. Також в ході виконання завдання я познайомивсь з такими базовими поняттями як спрайти та як їх додавати до сцени, класами BoxCollider2d, CircleCollider2d, Rigidbody2d та їх основними призначеннями і налаштуваннями. Окремим кроком було розроблено скрипт, що відповідає за основні рухи, стрибки та додано до ігрового об'єкта. Під час розробки використовувалась система контролю версій git та і основні зміни фіксувались відповідними комітами, також проект було опубліковано в віддалений репозиторій GitHub.