

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Лабораторна робота №1

з дисципліни

"Розробка ігрових застосувань. Unity рішення"

на тему "Дослідження базового патерну ігрового рушія Unity на прикладі
двовимірної технології"

Виконав:
студент групи ІП-93
Завальнюк М.
Викладач:
Катін П.Ю.

Київ – 2022

Зміст

| | |
|--|---|
| 1. Мета, завдання та вхідні дані | 3 |
| Мета | 3 |
| Завдання..... | 3 |
| Вхідні дані..... | 3 |
| 2. Встановлення інструментів | 4 |
| 3. Створення проекту та програмування об'єктів | 5 |
| 5. Додатки | 7 |

1. Мета, завдання та вхідні дані

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

Завдання:

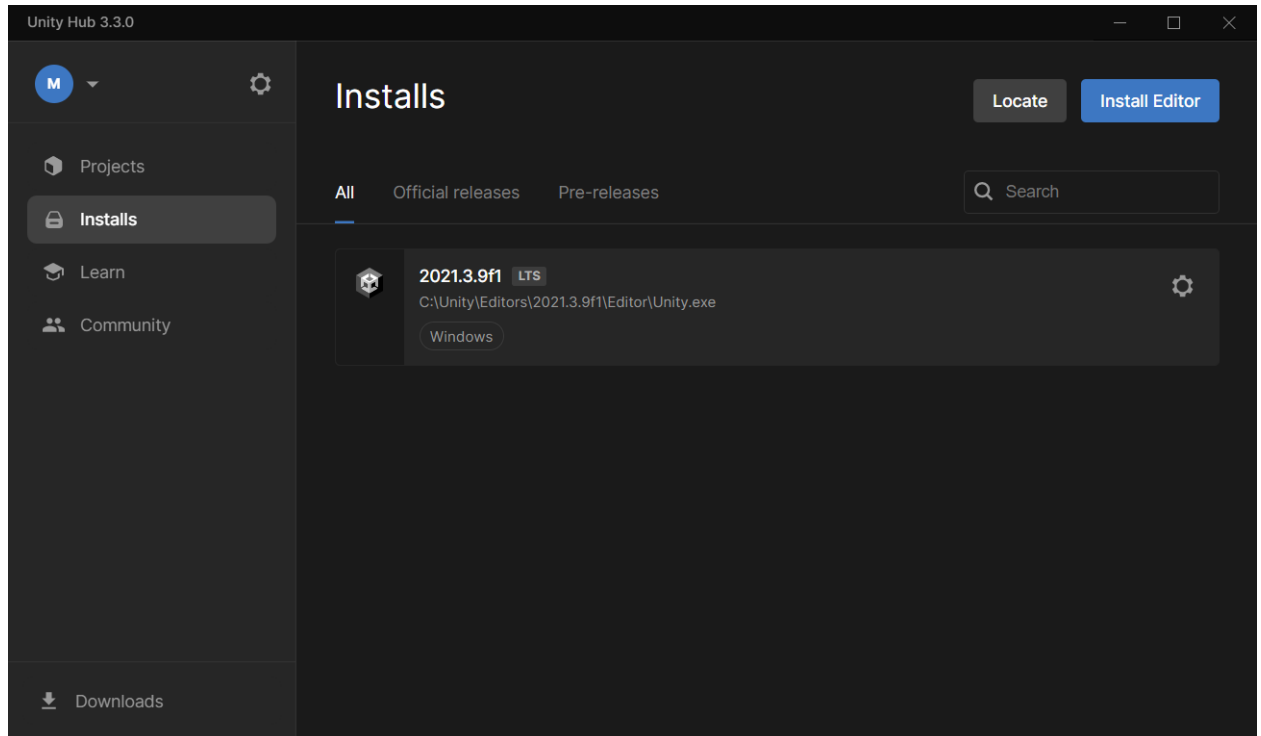
1. Обрати варіант.
2. Встановити ігровий двигун.
3. Встановити IDE.
4. Створити репозиторій.
5. Створити 2D проект із об'єктами, запрограмувати «гравця» рухатись.

Вхідні дані:

- Прізвище – Завальнюк;
- Ім'я – Максим;
- Шифр групи – ІІІ-93;
- Назва факультету – ФІОТ;
- Варіант – 7.

2. Встановлення інструментів

Спочатку, я завантажив [Unity Hub](#). За допомогою цієї програми легко керувати встановленими ігровими рушіями, проектами та ін. Потім завантажив безпосередньо Unity версії 2021.3.9f1:



Під час встановлення Unity можна було завантажити Visual Studio, але я цього не робив, оскільки буду використовувати [Visual Studio Code](#). І ось декілька причин:

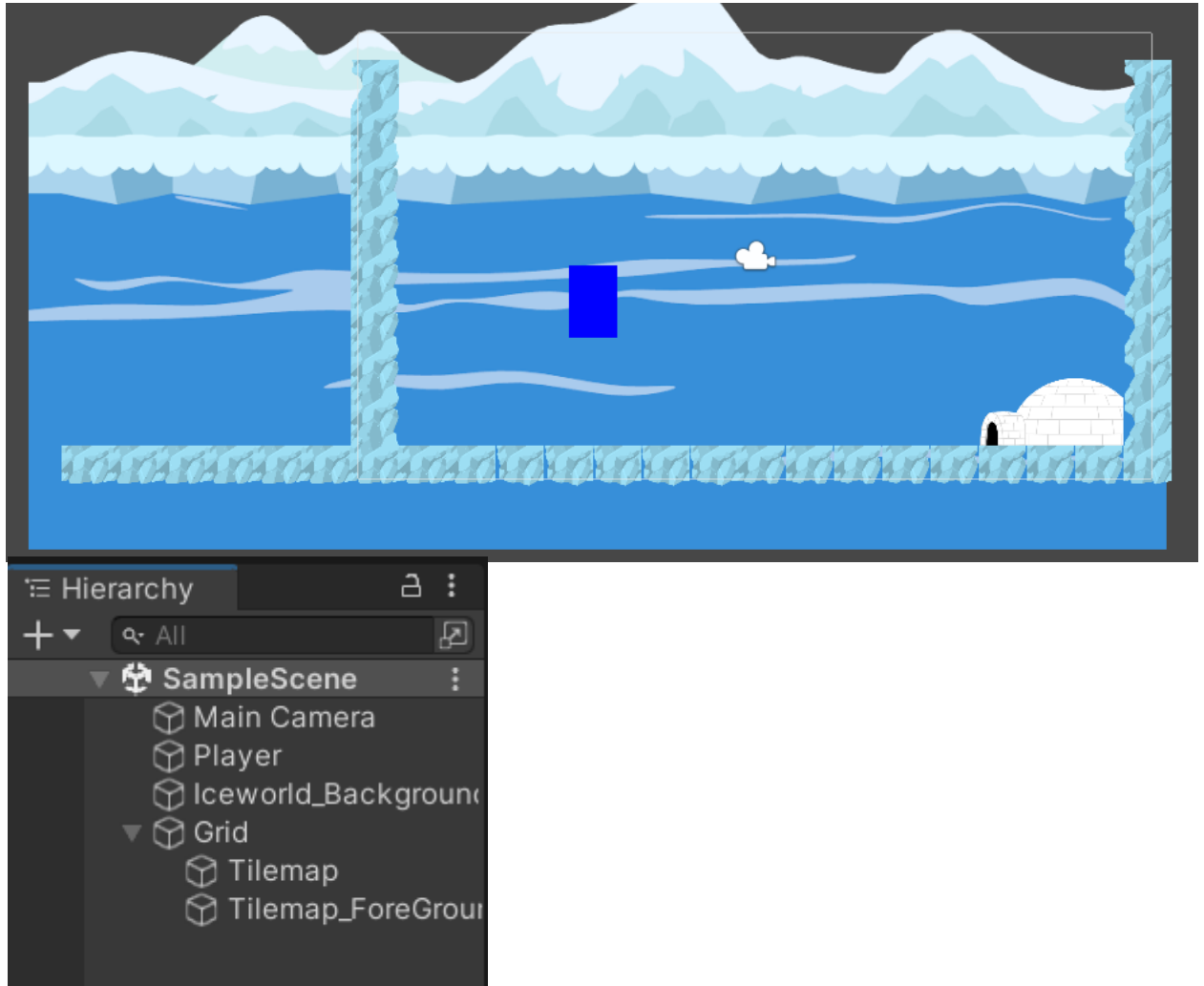
- Займає менше місця фізичної пам'яті та споживає менше ресурсів комп'ютера;
- Має усі інструменти для розробки на технології .NET за допомогою розширення C#;
- Має підтримку зі сторони Unity;
- Вже встановлений на моєму комп'ютері.

Далі за допомогою системи контролю версій git я створив репозиторій та завантажив його на GitHub:

| | | | |
|---------------------|------------------|--------------------|------|
| .git | 17.09.2022 11:48 | Папка с файлами | |
| GameProgLab1Group93 | 17.09.2022 11:13 | Папка с файлами | |
| Test project | 12.09.2022 15:48 | Папка с файлами | |
| .gitignore | 15.09.2022 16:01 | Текстовый докум... | 9 KB |

3. Створення проекту та програмування об'єктів

Запустивши Unity, я створив 2D проект. Потім завантажив ассет відповідно до свого варіанту. Додавши об'єкти до сцени, отримав наступне:



Для програмування руху «гравця» створив папку Scripts, у ній C#-скрипт та підв'язав його через компонент до об'єкту гравця. Код можна переглянути у додатках.

4. Висновок

Під час виконання даної лабораторної роботи, я ознайомився з базовим патерном 2D проєктів у Unity, навчився імпортувати ассети та користуватись тайлмепом. Ознайомився із методами Unity та синтаксисом C#.

5. Додатки

Посилання на проекти – [GitHub](#).

Код **PlayerController.cs**:

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

public class PlayerController : MonoBehaviour
{
    private Rigidbody2D rb2D;

    private float moveSpeed;
    private float jumpForce;
    private bool isJumping;
    private float moveHorizontal;
    private float moveVertical;

    // Start is called before the first frame update
    void Start()
    {
        rb2D = gameObject.GetComponent<Rigidbody2D>();

        moveSpeed = 3f;
        jumpForce = 60f;
        isJumping = false;
    }

    // Update is called once per frame
    void Update()
    {
        moveHorizontal = Input.GetAxisRaw("Horizontal");
        moveVertical = Input.GetAxisRaw("Vertical");
    }

    void FixedUpdate()
    {
        if (moveHorizontal > 0.1f || moveHorizontal < -0.1f)
        {
            rb2D.AddForce(new Vector2(moveHorizontal * moveSpeed, 0f),
ForceMode2D.Impulse);
        }

        if (!isJumping && moveVertical > 0.1f)
        {
            rb2D.AddForce(new Vector2(0f, moveVertical * jumpForce),
ForceMode2D.Impulse);
        }
    }

    void OnTriggerEnter2D(Collider2D collision)
    {
        if (collision.gameObject.tag == "Platform")
        {
            isJumping = false;
        }
    }

    void OnTriggerExit2D(Collider2D collision)
```

```
    {
        if (collision.gameObject.tag == "Platform")
        {
            isJumping = true;
        }
    }
}
```