

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

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки (ОТ)

Звіт до лабораторної роботи №2

з дисципліни:

«Розробка ігрових застосувань. Unity рішення»

на тему «Дослідження базового патерну ігрового рушія Unity на прикладі тривимірного ігрового застосування»

Перевірив:
доцент каф. ІСТ
Катін П. Ю.

Виконав:
студент 4 курсу, гр. ІТ-91
Левак О.О. (варіант 1)

2022р.

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 і знання елементів середовища розробки і основи вихідного коду для управління грою

Також лабораторна робота дає основні навички розробки з використанням IDE ігрового рушія.

Вхідні дані ЛР 2.

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

Створений проект IDE (3D) на основі рушія, що містить 2 сцени, ігровий персонаж. Можуть бути включені інші елементи. Розроблений і налагоджений скрипт для управління ігровим персонажем. Додані умови переходу між сценами. В якості ігрового персонажа використовується звичайний примітив. Достатньо продемонструвати рух ліворуч, праворуч, стрибки, коректну фізику, зупинку перед перешкодою, набір балів і перехід з одного рівня на інший. Можливо, за бажанням реалізувати інтерфейс гравця.

Проект розташовано у репозиторій на GitHub, основна мета

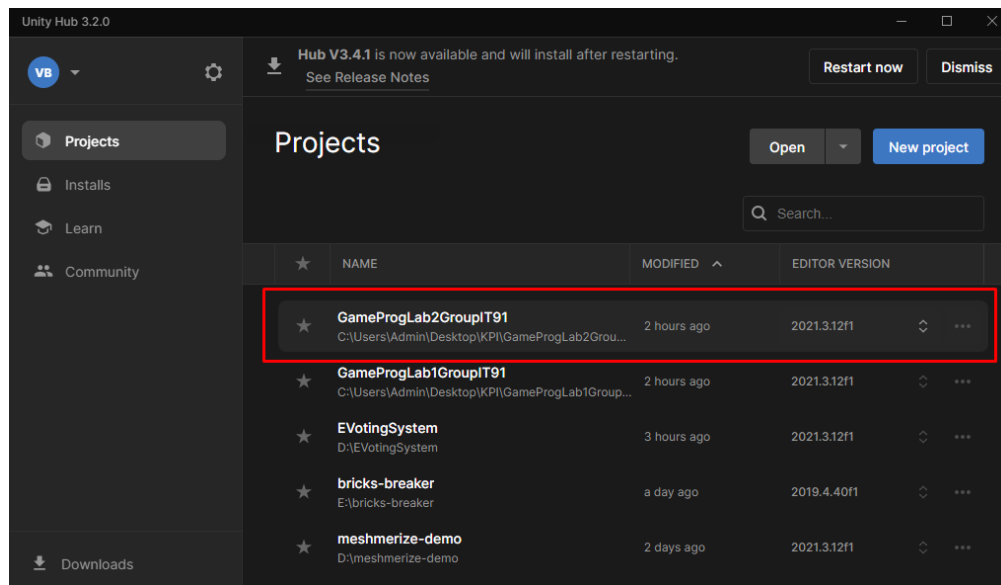
полягає у дослідженні і підтвердженні володіння обраною IDE (3D) і технологією розподіленої системи контролю версій. У разі виконання всіх умов і відмінного захисту надається 12,5 балів. Разом 25 балів.

1.2 Хід виконання

Варіант 2: примітив – сфера, асет -

<https://assetstore.unity.com/packages/3d/props/exterior/pbr-fences-181524>

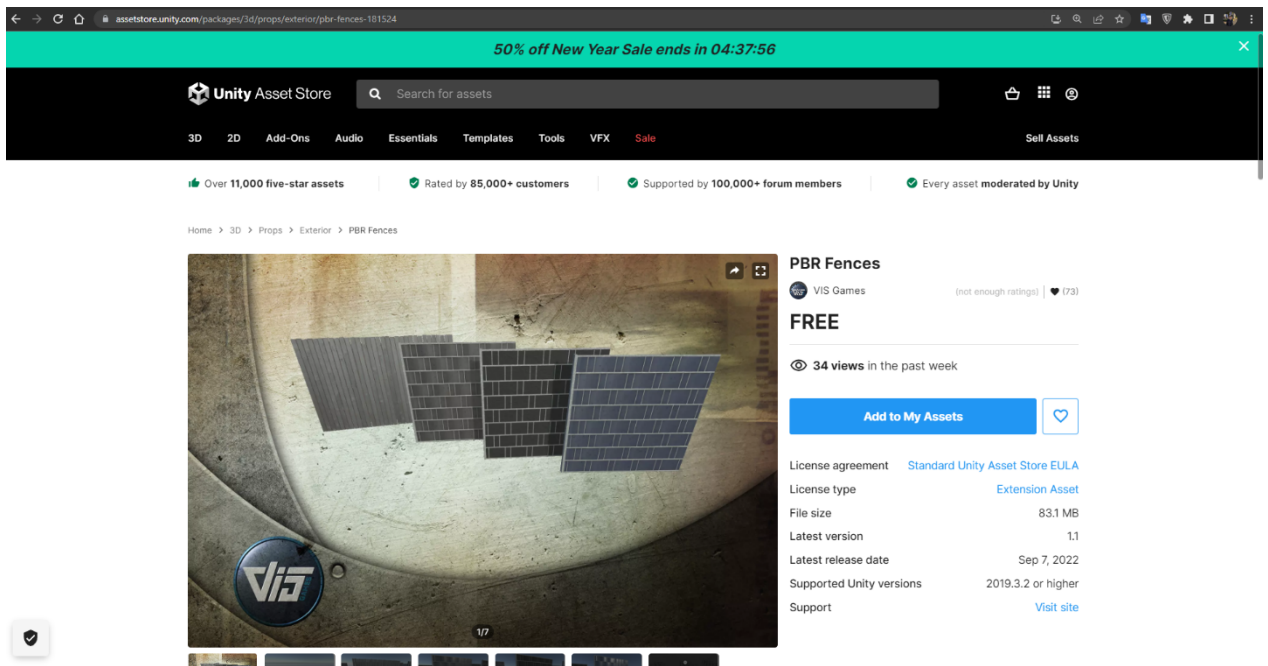
- Було створено 3D проект за допомогою UnityHub з назвою GameProgLab2GroupIT91, котрий поміщено у репозиторій GameProgLab2Group**IT91**:



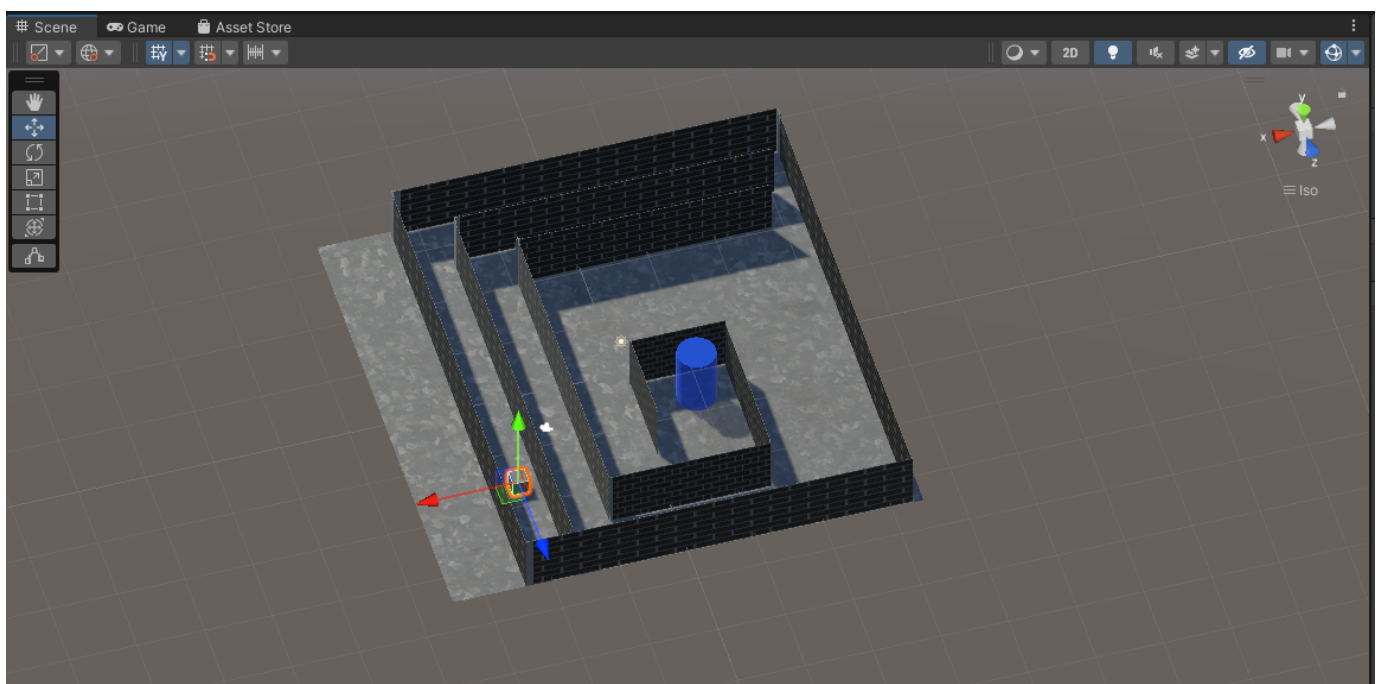
- Наступним кроком було створено примітив відповідно варіанту:



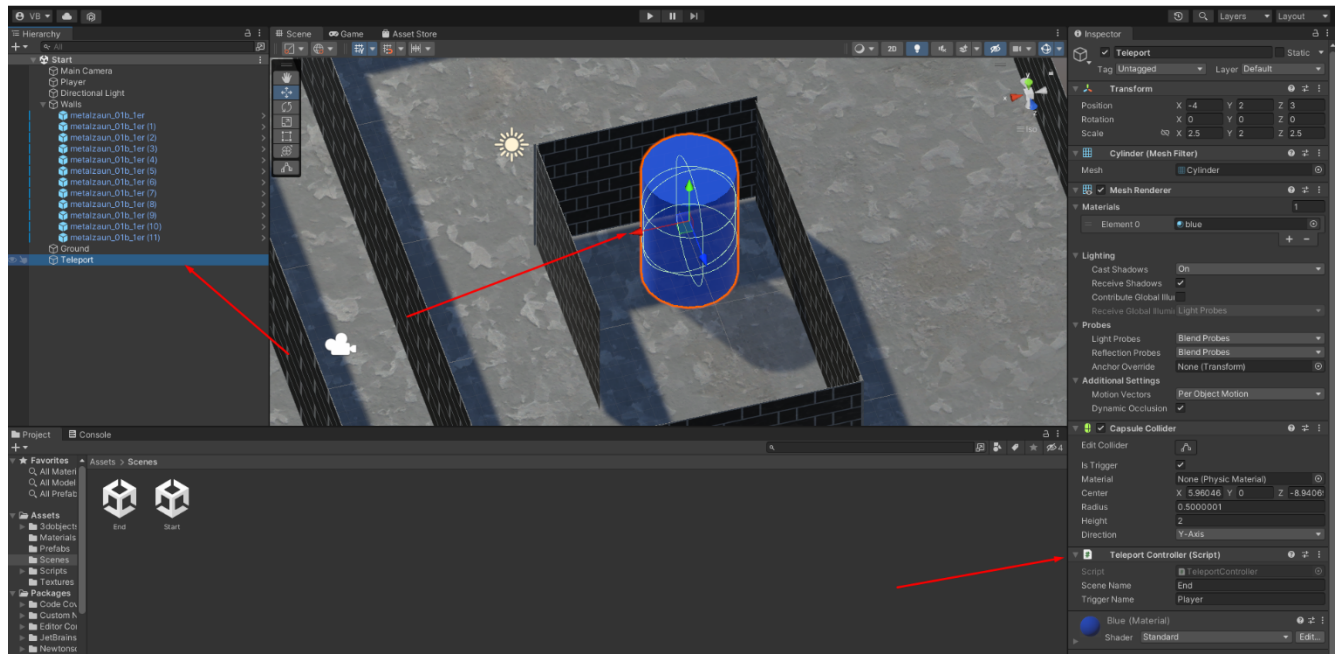
- Далі було обрано та завантажено відповідний до варіанту набір асетів



- Після здійснених вищенаведених підготовчих дій, було імпортовано набір асетів, та обрано з них префаби для конструювання мапи гри, також змінено матеріал ігрового об'єкту.



- Було створено нову сцену, і додано скрипт, що має метод переходу на нову сцену:



Скрипти проекту:

- PlayerController.cs:

```
using UnityEngine;

namespace Player
{
    public class PlayerController : MonoBehaviour
    {
        [SerializeField] private Rigidbody _rb;
        [SerializeField] private Transform _camera;

        [SerializeField] private float distance = 10;
        [SerializeField] private float sensitivity = 3;

        private Vector3 _cameraOffset;

        private void Start()
        {
            _cameraOffset = (Vector3.up + Vector3.back).normalized * distance;
        }

        private void FixedUpdate()
        {
            float horizontalInput = Input.GetAxis("Horizontal");
            float verticalInput = Input.GetAxis("Vertical");

            Vector3 verticalMovement = new Vector3(_camera.transform.forward.x, 0,
            _camera.transform.forward.z);
            Vector3 horizontalMovement = new Vector3(_camera.transform.right.x, 0,
            _camera.transform.right.z);
            Vector3 movement = horizontalInput * horizontalMovement.normalized +
            verticalInput * verticalMovement.normalized;

            _rb.AddForce(movement.normalized, ForceMode.VelocityChange);
        }

        private void Update()
        {
            Vector2 cameraRotation = new Vector2(-Input.GetAxis("Mouse Y") *
            sensitivity, Input.GetAxis("Mouse X") * sensitivity);
            _cameraOffset = Quaternion.Euler(cameraRotation) * _cameraOffset;
            _camera.position = transform.position + _cameraOffset;
            _camera.forward = -_cameraOffset.normalized;
        }
    }
}
```

```
}  
}
```

- TeleportController.cs

```
using UnityEngine;  
using UnityEngine.SceneManagement;  
  
namespace Teleport  
{  
    public class TeleportController : MonoBehaviour  
    {  
        [SerializeField] private string _sceneName;  
        [SerializeField] private string _triggerName;  
  
        private void OnTriggerEnter(Collider collider)  
        {  
            if (collider.name == _triggerName)  
            {  
                SceneManager.LoadScene(_sceneName);  
            }  
        }  
    }  
}
```

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

The screenshot shows the GitHub interface for a repository named 'GameProgLab2GroupIT91' by user 's-any-ok'. The repository is public and has 1 branch (master) and 0 tags. The file list shows the following files and their commit status:

File	Commit	Status
idea/idea.GameProgLab2GroupIT91...	first commit	now
Assets	first commit	now
Packages	first commit	now
ProjectSettings	first commit	now
.gitignore	first commit	now
README.md	first commit	now

The README.md file is expanded, showing the title 'GameProgLab2GroupIT91' and a section 'Вхідні дані' (Input data) with the following list:

- Прізвище студента: Левак
- Ім'я студента: Олександр
- Шифр навчальної групи: IT-91
- Скорочена назва факультету: ФІОТ
- Скорочена назва університету: КПІ ім. Ігоря Сікорського

1.3 Висновки

За період виконання даної роботи я набув базових навичок розробки з використанням IDE ігрового рушія Unity, шляхом створення проекту (3D) на основі рушія, що містить 2 сцени 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 додавання скрипта до ігрового об'єкту, управління камерою в 3D середовищі, класами BoxCollider, MeshCollider, Rigidbody та їх основними призначеннями і налаштуваннями.

Окремим кроком було розроблено скрипти, що відповідають за основні рухи, стрибки ігрового об'єкта, переміщення камери, набір балів, переходу на нову сцену. Під час розробки використовувалась система контролю версій git та і основні зміни фіксувались відповідними комітами, також проект було опубліковано в віддалений репозиторій GitHub.