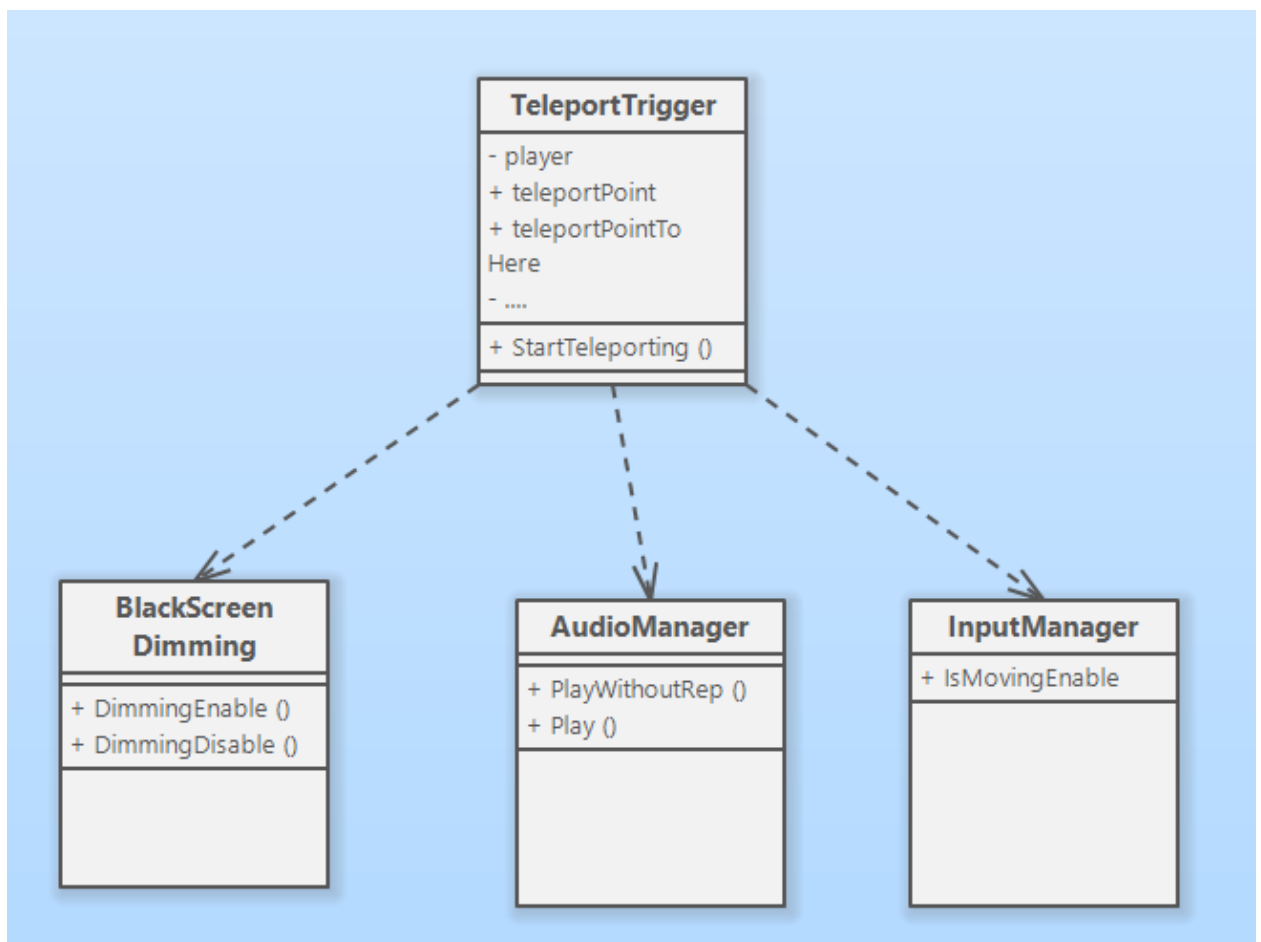


Фасад — це структурний патерн проектування, який надає простий інтерфейс до складної системи класів, бібліотеки або фреймворку.

Прикладів цього патерну в моєму проєкті багато, але зазвичай в них цей патерн реалізован не в чистому вигляді, а в дуже модифікованому. Тому я наведу один приклад найбільш схожий на той, що дається в книжках.

Мені потрібна була проста операція, де наш гравець потрапляє з точки А в точку В (телепортується). Операція на словах звісно проста, але щоб реалізувати її потрібно задіяти багато класів (відносно багато) та багато їх функцій. А в мене гра, де телепортація відіграє велику роль, тому я використав цей патерн заради економії часу та оптимізації коду.

В мене є клас `TeleportTrigger`, де всі функції для телепортації присутні, а весь інший непотрібний код прихований.



Тобто в мене є основний метод класу `TeleportTrigger` — **`StartTeleporting()`**, цей метод використовує всі інші методи та поля, інших класів для того, щоб перемістити гравця.

DimmingEnable та DimmingDisable відповідають за затемнення (та навпаки) екрану при телепортації.

PlayWithoutRep – це відтворення звуку телепортації без повторень.

Play – це відтворення звуку телепортації з повтореннями.

IsMovingEnable відповідає за те, чи може наш гравець рухатися під час телепортації.

Всі ці функції та поля клас TeleportTrigger розміщує в одну просто функцію за для простої задачі у вигляді переміщення з точка А до точки В.

Код виглядає так:

В цьому методі ми знаходимо всі потрібні об'єкти.

```
Unity Message | 0 references
private void Start()
{
    if (!dimming)
        dimming = GameObject.Find(CommonConstants.BLACK_SCREEN_DIMMING);
    if (dimming)
        dimming.GetComponent<BlackScreenDimming>().fadeSpeed = 0.5f;
    if (!audioManager)
        audioManager = GameObject.Find(CommonConstants.MAIN_AUDIOMANAGER);
    if (!player)
        player = GameObject.Find(CommonConstants.PLAYER);
}
```

Цей метод публічний він запускає телепортацію та ефекти. Параметр isActivated відповідає на те, чи потрібно телепортувати гравця. Якщо цей параметр активен, ми запускаємо інший приватний метод (Update), який як раз і переміщує нашого гравця у просторі.

```
2 references
public void StartTeleporting()
{
    isActivated = true;
    dimming?.GetComponent<BlackScreenDimming>().DimmingEnable();
    audioManager?.GetComponent<AudioManager>().PlayWithoutRep(MainAudioManagerConstants.TRANSITION);
}
```

```

private void Update()
{
    if (isActivated)
    {
        timeElapsed += Time.deltaTime;
        if (timeElapsed >= spawnAfter)
        {
            StartCoroutine(Teleport());
            dimming?.GetComponent<BlackScreenDimming>().DimmingDisable();
            timeElapsed = 0;
            isActivated = false;
        }
    }
}

```

Метод який відключає можливість рухатися та переміщує гравця, а потім через деякий час включає можливість назад.

```

IEnumerator Teleport()
{
    if (player.TryGetComponent(out InputManager inputManager))
        inputManager.IsMovingEnable = false;
    yield return new WaitForSeconds(0.05f);
    player.transform.position = teleportPoint.position;
    player.transform.localRotation = teleportPoint.rotation;
    yield return new WaitForSeconds(0.05f);
    if (inputManager != null)
        inputManager.IsMovingEnable = true;
}

```

Класи можете знайти за шляхом:

Assets\Scripts\Map\TeleportTrigger.cs