

```

using UnityEngine;
using System.Collections.Generic;
using UnityEngine.SceneManagement;

//На просторах інтернету був знайдений цей код який генерує платформи
згідно заданих префабів і відображає при заході на поинт і перестає при
виході з іншого
public class GenerationGround : MonoBehaviour {
    private GameObject playerTarget;
    public Transform[] koordinatyTransform;//Вказати 3 точки ліва, центр,
права
    private GameObject prefabStart;//префаб на якому буде відбуватись
старт гри
    private GameObject prefabFinish;//префаб на якому буде відбуватись
кінець гри
    private Transform[] section;//масив координат всіх префабів які
необхідні для побудови карти
    private List<Transform> sectionDisabled;//масив рандомізації префабів
карти
    private GameObject[] generationPrefabs;//Масив всіх префабів з папки
Resources
    private float posEnd, posStart;//позиції початку і кінця відображень
префабів по X
    private int index;
    private GameObject link;
    private GameObject[] Enemys;//для економії пам'яті видалення всіх
генерованих п еерешкод після проходження лівого поинта
    private GameObject Enemy;
    void Awake()
    {
        playerTarget = GameObject.Find("PlayerCar");
    }
    //private void FixedUpdate()
    //{
    //    Enemys = GameObject.FindGameObjectsWithTag("Clone");
    //    Debug.Log($"Enemy clone {Enemys.Length}");
    //}
    void Start()
    {
        posEnd = koordinatyTransform[0].position.x;
        posStart = Mathf.Abs(posEnd) * 3f;
        StartGame();
    }
    Transform RandomSection();//функція вибору рандомного префаба з тих
що містяться в section а та в свою чергу бере з папки GenerationPrefabs
{
    sectionDisabled = new List<Transform>();//створення списку
Transform(
    foreach (Transform tr in section)
    {
        if(!tr.gameObject.activeSelf)//Якщо
clone.SetActive(false)
{
            sectionDisabled.Add(tr);//додаємо до масиву
"Вимкнених"
        }
    }
    int rnd = Random.Range(0, sectionDisabled.Count);//Рандом від
0 до кількості вимкнених
    return sectionDisabled[rnd];//Повернути 1 з масиву вимкнутих
}

```

```

void AddSection()// Ввімкнення відображення префабів
{
    Transform bock = RandomSection();
    if(index == koordinyatyTransform.Length) index = 0;
    bock.parent = koordinyatyTransform[index];
    bock.localPosition = Vector3.zero;
    bock.gameObject.SetActive(true);//робити видимим якщо пападає
між лівий правий і центральний
    index++;
}

void StartGame()
{
    generationPrefabs =
Resources.LoadAll<GameObject>("GenerationPrefabs");//назва папки, в
середині папки Resources в якій лежать префаби карт
    section = new Transform[generationPrefabs.Length];//Масив
координат для дублікатів розмірністю масиву всіх префабів
    for(int i = 0; i < generationPrefabs.Length; i++)
    {
        GameObject clone = Instantiate(generationPrefabs[i]) as
GameObject;//створення дублікатів префабів
        clone.SetActive(false);//дублікати робимо не активними
        section[i] = clone.transform;//записуємо їх координати в
масив section
    }
    link = Resources.Load<GameObject>("Start");//імя
префабу(початкової сцени) з якого буде стартувати гра
    prefabStart = Instantiate(link) as GameObject;
    prefabStart.SetActive(true);
    prefabStart.transform.parent =
koordinyatyTransform[1];//координати центру
    prefabStart.transform.localPosition = Vector3.zero;

    Transform bock = RandomSection();
    bock.parent = koordinyatyTransform[0];//рандомізований префаб
за межами лівої межі
    bock.localPosition = Vector3.zero;
    bock.gameObject.SetActive(false);//стає не активним за межами
лівого пойнта

    bock = RandomSection();
    bock.parent = koordinyatyTransform[2];//рандомізований префаб в
межах правої межі
    bock.localPosition = Vector3.zero;
    bock.gameObject.SetActive(true);//Робити активним в межах
правого пойнта

    bock.parent = koordinyatyTransform[2];//рандомізований префаб в
межах правої межі
    bock.localPosition = Vector3.zero;
    bock.gameObject.SetActive(true);//Робити активним в межах правого
пойнта
}

void Update()
{
    Enemies = GameObject.FindGameObjectsWithTag("Clone");
    foreach (GameObject en in Enemies)
    {
        Enemy = en;
    }
}

```

```

        foreach (Transform tr in koordinyatyTransform)
        {
            //tr.position -= new Vector3(MoveCar._Speed * Time.deltaTime,
0, 0);
            tr.position -= new
Vector3(playerTarget.transform.position.x*Time.deltaTime, 0, 0);

            //Enemy.transform.position = new
Vector3(transform.position.x, tr.position.y, tr.position.z);

            if (tr.position.x < posEnd)
            {
                tr.position += new Vector3(posStart, 0, 0);
                tr.GetChild(0).gameObject.SetActive(false); //вимкнути
префаб який вийшов за межі
                tr.DetachChildren();
                AddSection();
            }
        }
        foreach (GameObject en in Enemys) //розгрузка пам'яті
        {
            if (en.gameObject.transform.position.x < posEnd ||
en.gameObject.transform.position.y < 0)
                if (gameObject.tag == "Clone")
                    Destroy(gameObject);
        }
    }
}

```