

ГЮИК. 501600.005-01 05 01

ДОДАТОК А

Графічний матеріал атестаційної роботи

ГЮИК. 501600.005

(позначення документу)

ГЮИК. 501600.005-01 05 01

Міністерство освіти і науки України
Харківський Національний Університет Радіоелектроніки

«ЗАТВЕРДЖУЮ»

керівник атестаційної роботи

Петрова Р. В.

Дослідження та оптимізація методів генерації рівнів в ігрових додатках

Графічний матеріал

ЛИСТ ЗАТВЕРДЖЕННЯ

ГЮИК. 501600.005 – ЛУ

УЗГОДЖЕНО:

РОЗРОБИВ:

ст. гр. СПРМ-19-1

Королук В.В.

2020 р

ГЮИК. 501600.005-01 05 01

ЗАТВЕРДЖЕНО

ГЮИК. 501600.005 – ЛУ

Дослідження та оптимізація методів генерації рівнів в ігрових додатках

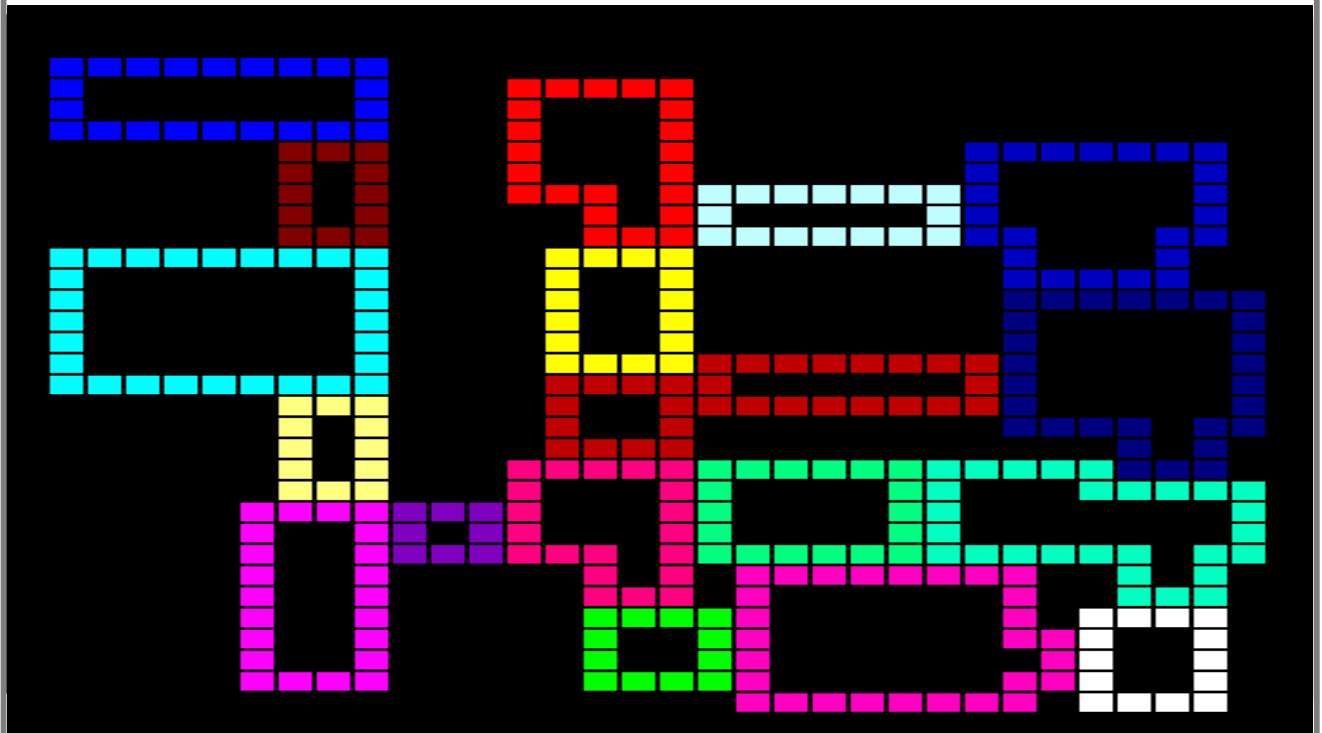
Графічний матеріал

ГЮИК. 501600.005 – ЛУ

АРКУШІВ 6

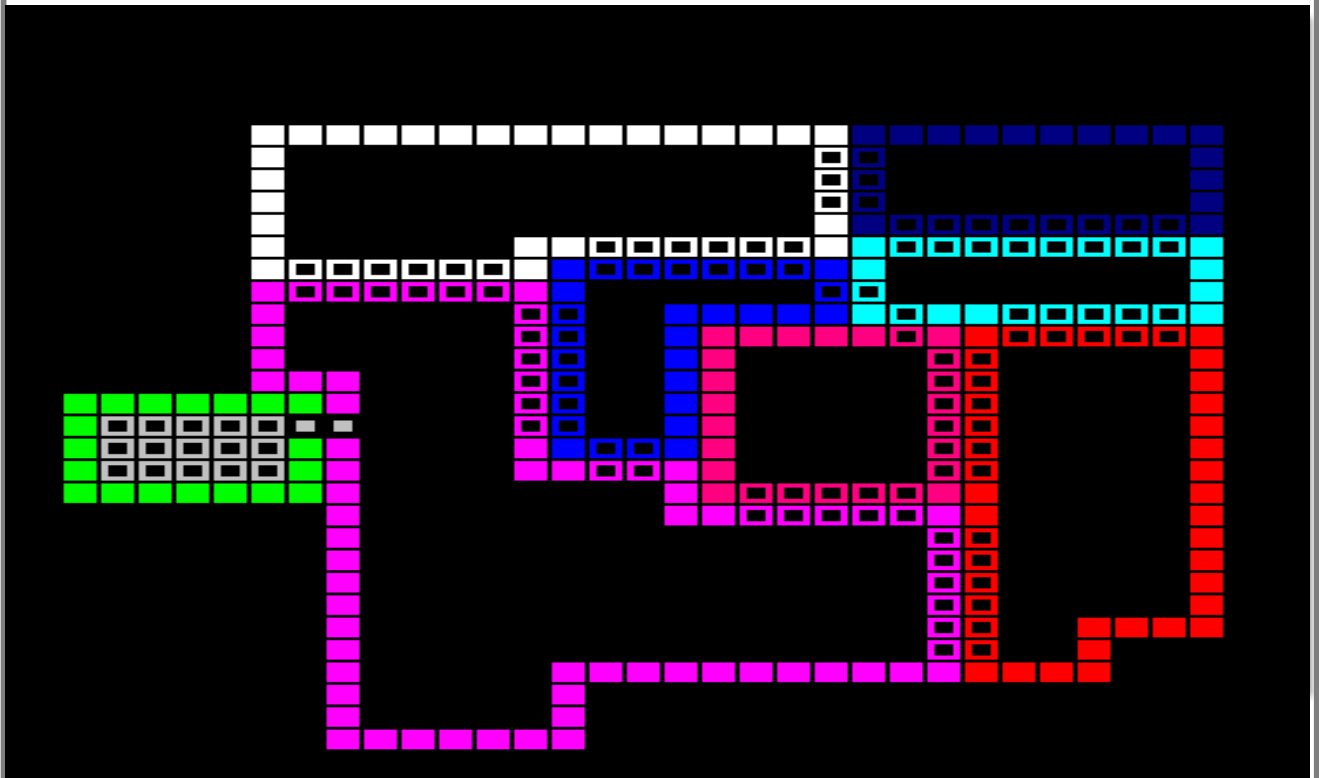
2020 р

Результат роботи алгоритму початкової геометрії BSP

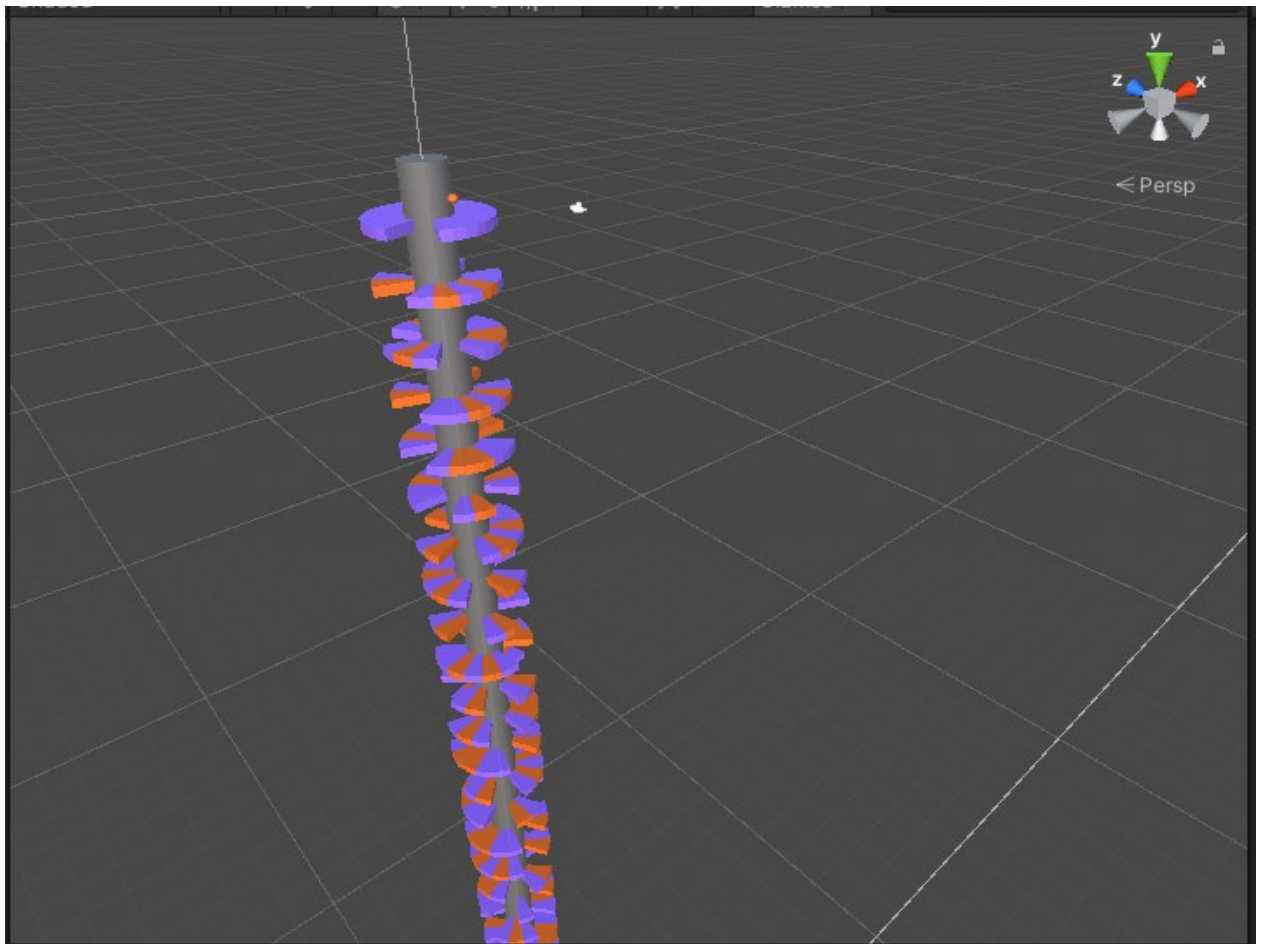


<i>Розроб.</i>	<i>Короліук В.В.</i>			<i>Результат роботи алгоритму початкової геометрії BSP</i>	
<i>Перевір.</i>	<i>Петрова Р. В.</i>				
<i>Н. Контр.</i>	<i>Петрова Р. В.</i>				
				<i>СПРм-19-1</i>	<i>Лист 1</i>
<i>Затверд.</i>	<i>Гребеннік І.В.</i>			<i>СТ</i>	<i>Листів 6</i>

Результат роботи алгоритму планування приміщень

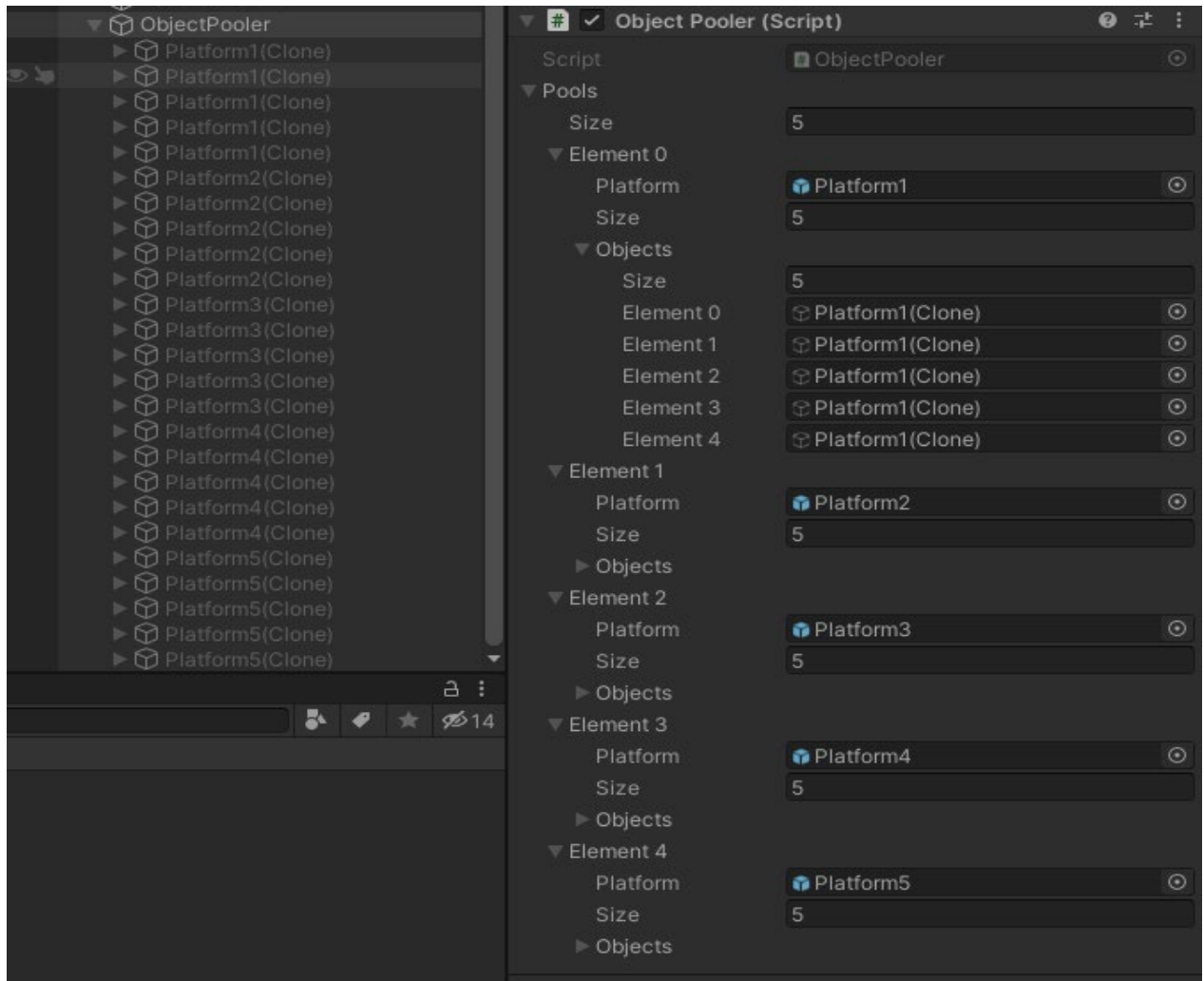


<i>Розроб.</i>	<i>Короліук В.В.</i>			<i>Результат роботи алгоритму планування приміщень</i>	
<i>Перевір.</i>	<i>Петрова Р. В.</i>				
<i>Н. Контр.</i>	<i>Петрова Р. В.</i>				
				<i>СПРм-19-1</i>	<i>Лист 2</i>
<i>Затверд.</i>	<i>Гребеннік І.В.</i>			<i>СТ</i>	<i>Листів 6</i>



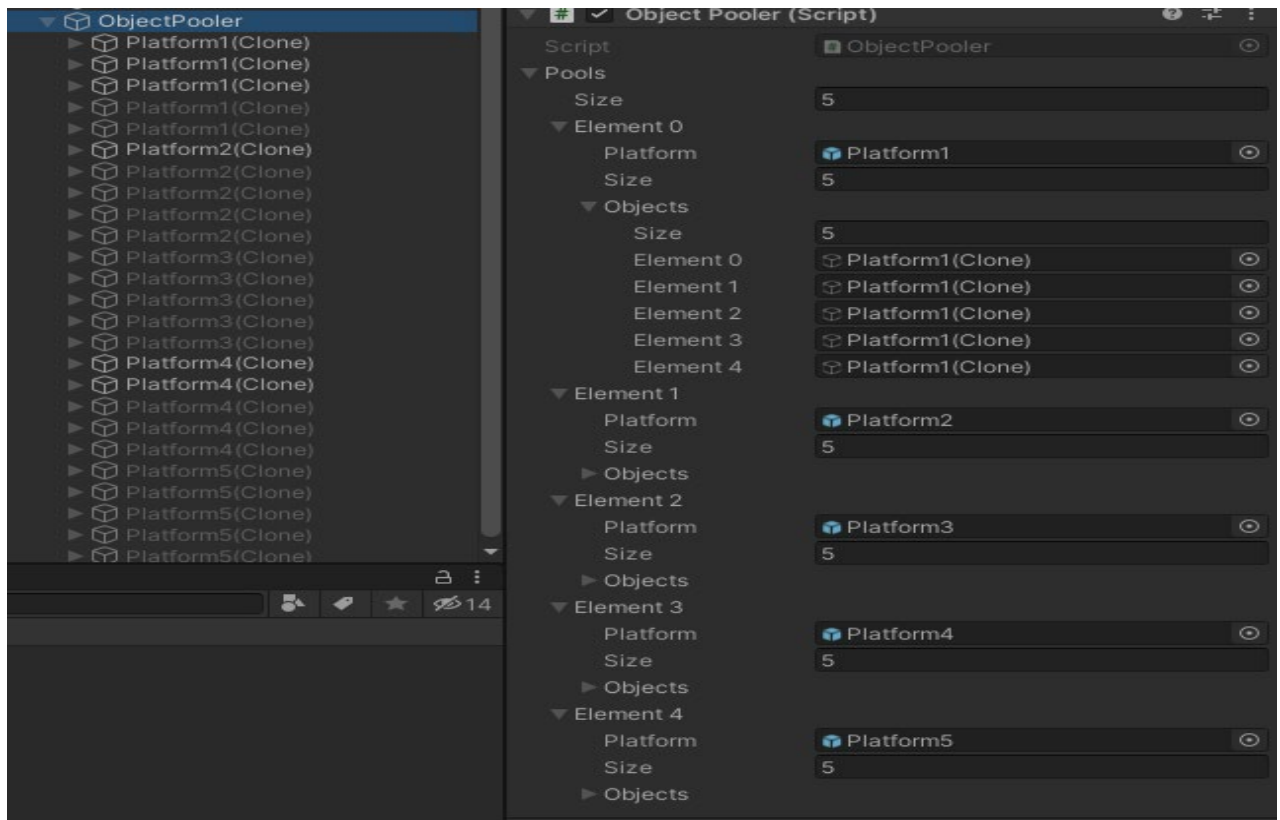
<i>Розроб.</i>	<i>Короліук В.В.</i>			<i>Результат генерації рівня за допомогою заготовлених частин</i>	
<i>Перевір.</i>	<i>Петрова Р. В.</i>				
<i>Н. Контр.</i>	<i>Петрова Р. В..</i>				
				<i>СПРМ-19-1</i>	<i>Лист 3</i>
<i>Затверд.</i>	<i>Гребеннік І.В.</i>			<i>СТ</i>	<i>Листів 6</i>

Ініціалізація Пулу об'єктів

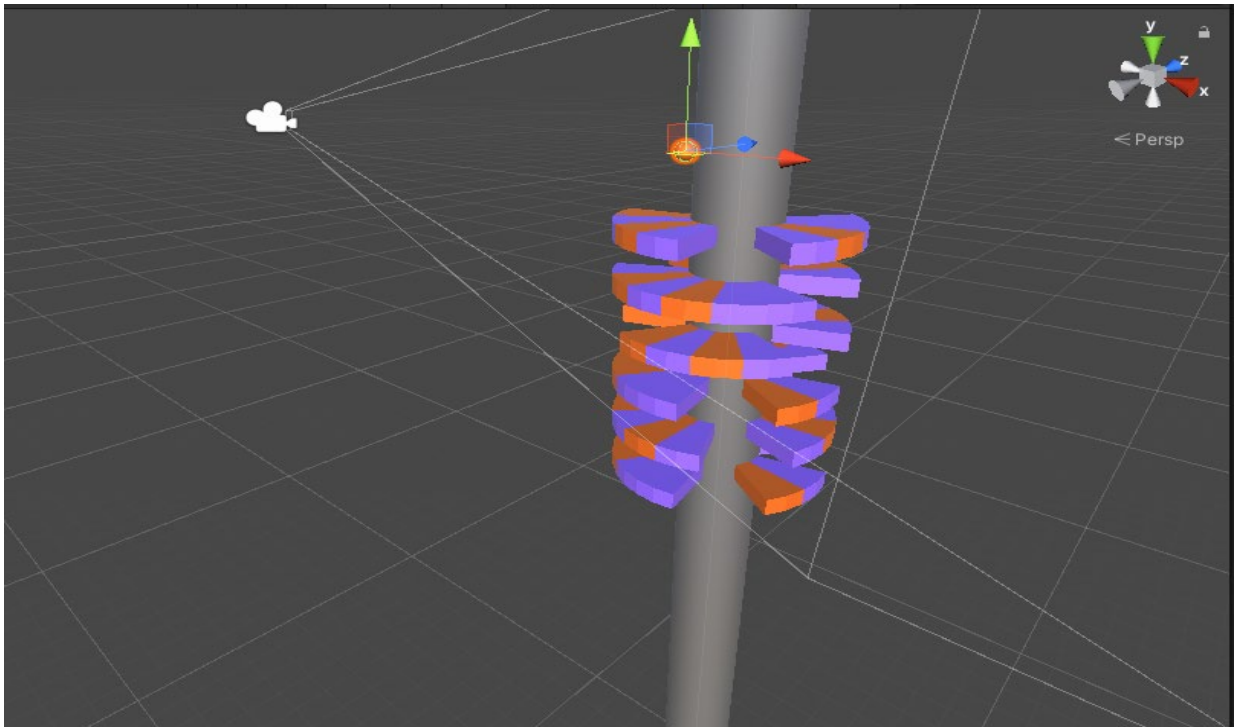


Розроб.	Короліук В.В.			<i>Ініціалізація Пулу об'єктів</i>	
Перевір.	Петрова Р. В.				
Н. Контр.	Петрова Р. В.				
				<i>СПРМ-19-1</i>	<i>Лист 4</i>
Затверд.	Гребеннік І.В.			<i>СТ</i>	<i>Листів 6</i>

Робота Пулу об'єктів під час динамічної генерації рівня



Розроб.	Короліук В.В.			Робота Пулу об'єктів під час динамічної генерації рівня	
Перевір.	Петрова Р. В.				
Н. Контр.	Петрова Р. В.				
				СПРМ-19-1	Лист 5
Затверд.	Гребеннік І.В.			СТ	Листів 6



<i>Розроб.</i>	<i>Короліук В.В.</i>			<i>Результат роботи модифікованого методу генерації</i>	
<i>Перевір.</i>	<i>Петрова Р.В.</i>				
<i>Н. Контр.</i>	<i>Петрова Р.В.</i>				
				<i>СПРМ-19-1</i>	<i>Лист 6</i>
<i>Затверд.</i>	<i>Гребеннік І.В.</i>			<i>СТ</i>	<i>Листів 6</i>

ГЮИК. 501600.005-01 05 01

ДОДАТОК Б

Текст програми

ГЮИК. 501600.005 – 01 05 01

(позначення документу)

ГЮИК. 501600.005-01 05 01

Міністерство освіти і науки України
Харківський Національний Університет Радіоелектроніки

«ЗАТВЕРДЖУЮ»
керівник атестаційної роботи
Петрова Р. В.

Дослідження та оптимізація методів генерації рівнів в ігрових додатках

Текст програми

ЛИСТ ЗАТВЕРДЖЕННЯ

ГЮИК. 501600.005 – 01 05 01 – ЛУ

УЗГОДЖЕНО:

РОЗРОБИВ:
ст. гр. СПРм-19-1
Королюк В.В.

2020 р.

ГЮИК. 501600.005-01 05 01

ЗАТВЕРДЖЕНО

ГЮИК. 501600.005 – 01 05 01 – ЛУ

Дослідження та оптимізація методів генерації рівнів в ігрових додатках

Текст програми

ГЮИК. 501600.005 – 01 05 01 – ЛУ

АРКУШІВ 5

2020 р

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

public class ObjectPooler : MonoBehaviour
{
    [SerializeField]
    private List<Pool> pools;

    private void Start()
    {
        Initialize();
    }

    private void Initialize()
    {
        for (int i = 0; i < pools.Count; i++)
        {
            for (int j = 0; j < pools[i].size; j++)
            {
                var tmp = Instantiate(pools[i].platform);
                tmp.transform.parent = transform;
                tmp.transform.localScale = transform.localScale;
                tmp.transform.rotation = transform.rotation;
                tmp.gameObject.SetActive(false);
                pools[i].objects.Add(tmp);
            }
        }
    }

    public GameObject GetObjectFromPool(int poolNumber)
    {
        var freeObject = pools[poolNumber].objects.Where(item => item.activeSelf ==
false).FirstOrDefault();
        if (freeObject == null)
        {
            var tmp = Instantiate(pools[poolNumber].platform);
            tmp.transform.parent = transform;
            tmp.transform.localScale = transform.localScale;
            tmp.transform.rotation = transform.rotation;
            tmp.gameObject.SetActive(false);
            pools[poolNumber].objects.Add(tmp);
            return tmp;
        }
        return freeObject;
    }
}
```

```
[System.Serializable]
public class Pool
{
    public GameObject platform;
    public int size;
    public List<GameObject> objects;
}

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.EventSystems;

public class PlatformManager : MonoBehaviour, IDragHandler
{
    [SerializeField]
    private GameObject obstackles;
    [SerializeField]
    private GenerateLevel generator;
    [SerializeField]
    private PlayerController player;

    private static List<Transform> platforms = new List<Transform>();

    private void Start()
    {
        player.onPlayerBreakPlatform += RefreshPlatformsList;
        RefreshPlatformsList();
    }

    private void RefreshPlatformsList()
    {
        platforms = generator.GetActivaPlatforms();
    }

    public void OnDrag(PointerEventData eventData)
    {
        foreach (Transform g in platforms)
        {
            g.Rotate(0, -eventData.delta.x, 0);
        }
    }
}
```

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

public class PlayerController : MonoBehaviour
{
    public delegate void PlayerThroughPlatformHandler();
    public event PlayerThroughPlatformHandler onPlayerBreakPlatform;

    [SerializeField]
    private float jumpVelocity = 10f;

    private Rigidbody rigidBody;
    private int comboCounter;

    private void Awake()
    {
        comboCounter = 0;
        rigidBody = gameObject.GetComponent<Rigidbody>();
    }

    private void OnCollisionEnter(Collision collision)
    {
        if (collision.collider.tag == "Finish")
        {
            SceneManager.LoadScene("Level");
            return;
        }
        if (comboCounter > 2)
        {
            Destroy(collision.transform.parent.gameObject);
            onPlayerBreakPlatform?.Invoke();
            rigidBody.velocity = Vector3.up * jumpVelocity;
            comboCounter = 0;
            return;
        }
        if (collision.collider.tag == "Obstacle")
        {
            rigidBody.velocity = Vector3.up * jumpVelocity;
        }
        if (collision.collider.tag == "DangerZone")
        {
            SceneManager.LoadScene("Level");
        }
        comboCounter = 0;
    }
}
```

```
}

private void OnTriggerEnter(Collider other)
{
    onPlayerBreakPlatform?.Invoke();
    comboCounter++;
    Destroy(other.gameObject);
}
}

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

public class GenerateLevel : MonoBehaviour
{
    [SerializeField]
    private List<GameObject> prefabs;
    [SerializeField]
    private PlayerController player;
    [SerializeField]
    private ObjectPooler pooler;
    [SerializeField]
    private int startPosY, distanceY;

    private int endposY;

    [SerializeField]
    private Transform parent;

    private List<Transform> activePlatforms = new List<Transform>();

    private void Awake()
    {
        Init();
        player.onPlayerBreakPlatform += ReturnPlatformToPool;
    }

    private void Init()
    {
        endposY = startPosY - distanceY;
        for (int i = 0; i < 6; i++)
        {
            SetNextPlatform();
        }
    }
}
```

```
}  
  
private void ReturnPlatformToPool()  
{  
    var tmp = activePlatforms[0];  
    activePlatforms.RemoveAt(0);  
    tmp.gameObject.SetActive(false);  
    SetNextPlatform();  
}  
  
private void SetNextPlatform()  
{  
    var platform = pooler.GetObjectFromPool(Random.Range(0, 4));  
    platform.transform.position = new Vector3(0, endposY, 0);  
    platform.SetActive(true);  
    platform.transform.Rotate(new Vector3(0, Random.Range(-180f, 180f), 0));  
    activePlatforms.Add(platform.transform);  
    endposY -= distanceY;  
}  
  
public List<Transform> GetActivaPlatforms()  
{  
    return activePlatforms;  
}
```

ДОДАТОК В

«Специфікація»

ГЮИК. 501600.005

ДОДАТОК Г

«Відомість атестаційної роботи»

ГЮИК. 501600.005 ДЗ

(позначення документу)

№	Позначення				Найменування	Додаткові відомості		
					Текстові документи			
1.	ГЮИК. 501600.005 ПЗ				Пояснювальна записка	68 стор.		
2.	ГЮИК. 501600.005				Специфікація	1 стор		
3.	ГЮИК. 501600.005 - 01 05 01				Текст програми	5 стор.		
					Графічні документи			
4.	ГЮИК. 501600.005 С10				Результат роботи алгоритму початкової геометрії BSP	1 аркуш		
5.					Результат роботи алгоритму планування приміщень	1 аркуш		
6.					Результат генерації рівня за допомогою заготовлених частин	1 аркуш		
7.					Ініціалізація Пулу об'єктів	1 аркуш		
8.					Робота Пулу об'єктів під час динамічної генерації рівня	1 аркуш		
9.					Результат роботи модифікованого методу генерації	1 аркуш		
10.								
11.								
12.								
16.								
17.								
	Арк..							
Змін.	<i>Корольок В.В.</i>	№ докум	Підп.		ГЮИК. 501600.005 ДЗ			
Розр об.	<i>Петрова Р.В.</i>			Дата				
Перевір.	<i>Петрова Р.В.</i>				<i>Дослідження та оптимізація методів генерації рівнів в ігрових додатках</i>		Аркуш	Аркушів
Н. Контр.	<i>Гребеннік І.В.</i>						1	1
							<i>ХНУРЕ Кафедра СТ</i>	