

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using UnityEngine.SocialPlatforms.Impl;

public class Manager : MonoBehaviour
{
    public GameObject whiteTilePrefab;
    public GameObject blackTilePrefab;

    private int currentRow = 0;
    private List<BlackTile> blackTiles = new List<BlackTile>();

    private int score = 0;

    public TMPro.TextMeshProUGUI finalScoreText;
    void Start()
    {
        for (int i = 0; i < 4; i++)
        {
            SpawnRow(i);
            currentRow++;
        }
    }

    void Update()
    {
        if (Input.GetKeyDown(KeyCode.A)) CheckColumn(0);
        if (Input.GetKeyDown(KeyCode.S)) CheckColumn(1);
        if (Input.GetKeyDown(KeyCode.D)) CheckColumn(2);
        if (Input.GetKeyDown(KeyCode.F)) CheckColumn(3);
    }

    void CheckColumn(int column)
    {
        float minY = float.MaxValue;
        foreach (var tile in blackTiles)
        {

```

```

        if (tile.transform.position.y < minY)
            minY = tile.transform.position.y;
    }

    BlackTile target = null;
    foreach (var tile in blackTiles)
    {
        if (tile.transform.position.y == minY && tile.columnIndex == column)
        {
            target = tile;
            break;
        }
    }

    if (target != null)
    {
        Camera.main.transform.position += new Vector3(0, 1.5f, 0); // 上移相机
        SpawnNewRow();

        blackTiles.Remove(target);
        Destroy(target.gameObject);

        score++;
        Debug.Log("Score: " + score);

        Debug.Log("Correct key pressed!");
        finalScoreText.text = "";
    }

    else
    {
        // 错误, 变红
        foreach (var tile in blackTiles)
        {
            if (tile.transform.position.y == minY && tile.columnIndex == column)
            {
                tile.GetComponent<Renderer>().material.color = Color.red;
            }
        }
        Debug.Log("Wrong key!");
        Debug.Log("Final Score: " + score);

        if (finalScoreText != null)
            finalScoreText.text = "Final Score: " + score;
    }

```

```

        score = 0;
    }
}

public void SpawnNewRow()
{
    SpawnRow(currentRow);
    currentRow++;
}

void SpawnRow(int rowIndex)
{
    int blackTileIndex = Random.Range(0, 4);
    float prefabWidth = whiteTilePrefab.GetComponent<Renderer>().bounds.size.x;
    float prefabHeight = whiteTilePrefab.GetComponent<Renderer>().bounds.size.y;

    for (int i = 0; i < 4; i++)
    {
        Vector3 pos = new Vector3(i * prefabWidth, rowIndex * prefabHeight, 0);
        GameObject tile = Instantiate(i == blackTileIndex ? blackTilePrefab : whiteTilePrefab,
pos, Quaternion.identity);
        tile.transform.parent = transform;

        if (i == blackTileIndex)
        {
            var blackTile = tile.AddComponent<BlackTile>();
            blackTile.Init(this, rowIndex, i);
            blackTiles.Add(blackTile);
        }
    }
}
}

```