

Міністерство освіти і науки України  
Прикарпатський національний університет  
імені Василя Стефаника

*Факультет математики та інформатики  
Кафедра інформаційних технологій*

Лабораторна робота № 1  
з дисципліни  
"Програмування ігрових застосувань"

Тема: «Лабораторна робота №1. Вступ до Unity 3D. Основні ігрові 2D-об'єкти».

Виконав: Вінтоняк В.Д.  
Група ІПЗ-31  
Дата: 15 березня 2024р.  
Викладач: Горелов В.О.

## Завдання

Гру придумав у 1878 р. Ной Чепмен. Посилання на Вікіпедію.

Реалізуйте сцену у вигляді більшого квадрата, що містить 15 менших квадратиків-пластинок (пластинки займають комірки між лініями уявної сітки).

Кожен із менших квадратів пронумерований цифрами від 1 до 15. Таким чином, залишається одна вільна комірка.

Положення пластинок вибирається випадковим чином.

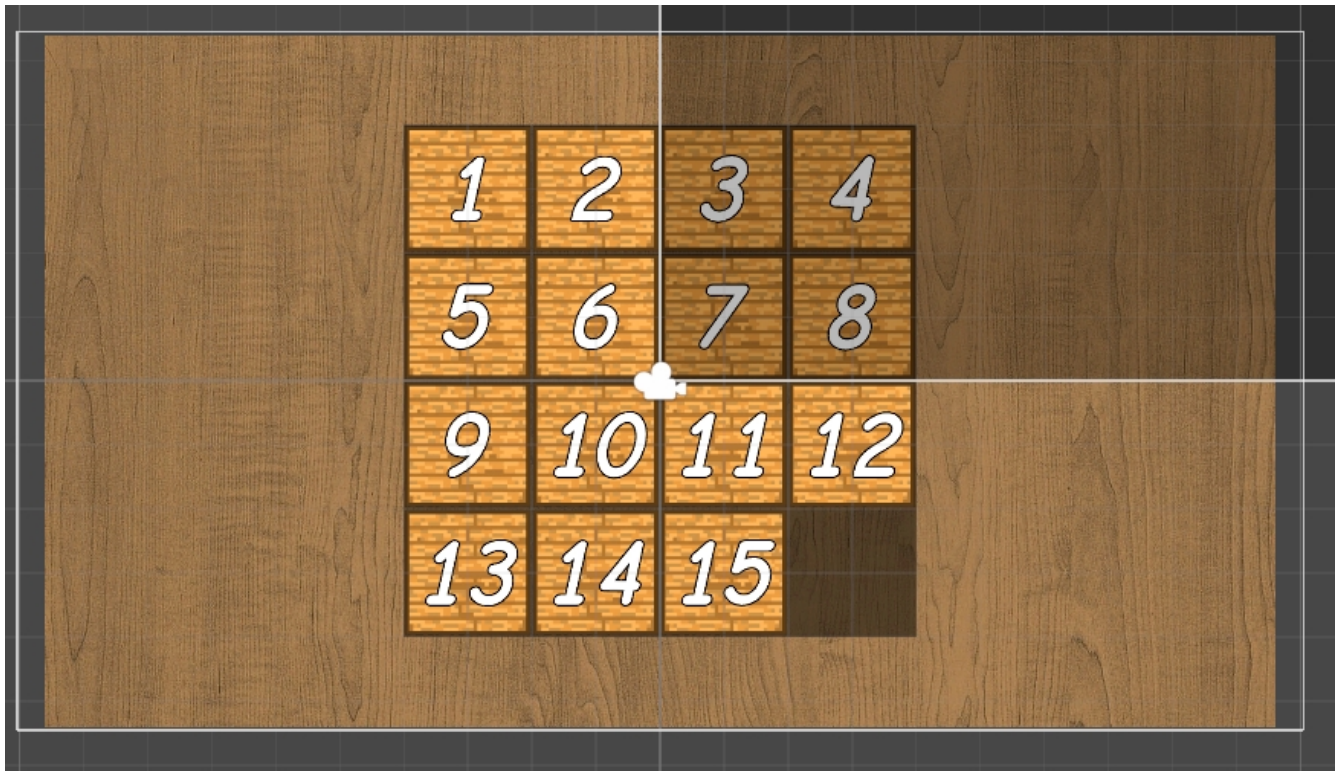
Завдання гравця: використовувати вільну позицію і переміщувати пластинки таким чином, щоб побудувати послідовність від 1 до 15, починаючи з лівої верхньої комірки. Після завершення ряду справа нумерація продовжується з лівої комірки наступного (нижнього) ряду.

Найпростіший спосіб взаємодії: клік по потрібній пластинці викликає її переміщення на місце порожньої клітинки, якщо такий рух можливий.

Дозволено переміщення тільки сусідніх до порожньої клітинки пластинок. Також можна реалізувати Drag (якщо відпустити пластинку раніше, ніж її бік пройде центр вільної комірки – пластинка повертається назад, відпустити після проходження середини вільної комірки (по вертикалі чи горизонталі, залежно від напрямку переміщення) – пластинка займає вільну комірку.

## Виконання роботи

Game scene



## Materials, prefabs, scripts



### GameScript:

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

public class GameScript : MonoBehaviour
{
    [SerializeField] private Transform emptySpace = null;
    private Camera _camera;
    [SerializeField] private TileScript[] tiles;
    private int emptySpaceIndex = 15;
    private bool _isFinished;
    [SerializeField] private GameObject endPanel;
    [SerializeField] private Text endPanelTimerText;

    void Start()
    {
        endPanel.SetActive(false);
        _camera = Camera.main;
        Shuffle();
    }

    void Update()
    {

```

```

if (Input.GetMouseButtonDown(0))
{
    Ray ray = _camera.ScreenPointToRay(Input.mousePosition);
    RaycastHit2D hit = Physics2D.Raycast(ray.origin, ray.direction);
    if (hit)
    {
        if (Vector2.Distance(emptySpace.position, hit.transform.position) < 2.5)
        {
            Vector2 lastEmptySpacePosition = emptySpace.position;
            TileScript thisTile = hit.transform.GetComponent<TileScript>();
            emptySpace.position = thisTile.targetPostition;
            thisTile.targetPostition = lastEmptySpacePosition;

            int tileIndex = findIndex(thisTile);
            tiles[emptySpaceIndex] = tiles[tileIndex];
            tiles[tileIndex] = null;
            emptySpaceIndex = tileIndex;
        }
    }
}

if (!_isFinished)
{
    int correctTiles = 0;

    foreach (var a in tiles)
    {
        if (a != null)
        {
            if (a.inRightPlace)
                correctTiles++;
            endPanel.SetActive(false);
        }
    }

    if (correctTiles == tiles.Length - 1)
    {
        _isFinished = true;
        endPanel.SetActive(true);

        var a = GetComponent<TimerScript>();
        a.StopTimer();
        endPanelTimerText.text = (a.minutes < 10 ? "0" : "") + a.minutes + ":" +
(a.seconds < 10 ? "0" : "") + a.seconds;
    }
}

public void PlayAgain()
{
    SceneManager.LoadScene(SceneManager.GetActiveScene().name);
}

public void Shuffle()
{
    if (emptySpaceIndex != 15)
    {
        var tile15LastPos = tiles[15].targetPostition;
        tiles[15].targetPostition = emptySpace.position;
        emptySpace.position = tile15LastPos;
    }
}

```

```

        tiles[emptySpaceIndex] = tiles[15];
        tiles[15] = null;
        emptySpaceIndex = 15;
    }

    int inversion;

    do
    {
        for (int i = 0; i <= 14; i++)
        {
            if (tiles[0] != null)
            {
                var lastpos = tiles[i].targetPostition;
                int randomPosition = Random.Range(0, 14);

                tiles[i].targetPostition = tiles[randomPosition].targetPostition;
                tiles[randomPosition].targetPostition = lastpos;

                var tile = tiles[i];
                tiles[i] = tiles[randomPosition];
                tiles[randomPosition] = tile;
            }
        }
        inversion = GetInversions();
    }while(inversion%2 != 0);
}

public int findIndex(TileScript ts)
{
    for (int i = 0; i < tiles.Length; i++)
    {
        if (tiles[i] != null)
        {
            if (tiles[i] == ts)
            {
                return i;
            }
        }
    }

    return -1;
}

int GetInversions()
{
    int inversionsSum = 0;
    for (int i = 0; i < tiles.Length; i++)
    {
        int thisTileInversion = 0;
        for (int j = i; j < tiles.Length; j++)
        {
            if (tiles[j] != null)
            {
                if (tiles[i].number > tiles[j].number)
                {
                    thisTileInversion++;
                }
            }
        }
        inversionsSum += thisTileInversion;
    }
}

```

```

    }
    return inversionsSum;
}
}

```

## TileScript:

```

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

public class TileScript : MonoBehaviour
{
    public Vector3 targetPostition;
    private Vector3 correctPosition;
    public int number;
    public bool inRightPlace;
    void Awake()
    {
        targetPostition = transform.position;
        correctPosition = transform.position;
    }

    void Update()
    {
        transform.position = Vector3.Lerp(transform.position, targetPostition, 0.05f);

        if(targetPostition == correctPosition)
        {
            inRightPlace = true;
        }
        else
        {
            inRightPlace = false;
        }
    }
}

```

## TimerScript:

```

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

public class TimerScript : MonoBehaviour
{

```

```

[SerializeField] private Text timeText;
public int seconds, minutes;
void Start()
{
    AddToSecond();
}

private void AddToSecond()
{
    seconds++;
    if (seconds > 59)
    {
        minutes++;
        seconds = 0;
    }

    timeText.text = (minutes < 10 ? "0" : "") + minutes + ":" + (seconds < 10 ? "0" :
"" ) + seconds;
    //minutes + ":" + seconds

    Invoke(nameof(AddToSecond), 1);
}

public void StopTimer()
{
    CancelInvoke(nameof(AddToSecond));
    timeText.gameObject.SetActive(false);
}
}

```