Національний університет “Острозька академія”

Кафедра економіко-математичного моделювання та інформаційних технологій

**Дизайн документ**

**з дисципліни “Розробка ігрових додатків на Unity”**

Студента 3-го курсу

спеціальності КН-31

Солонінка І. С.

Керівник:

Власюк В.В.

Острог 2020

**Мета**: при виконанні власного проекту, поглиблено розібрати сам рушій Unity, в розробці 2d проектів, та отримати практичні навички для подальшої розробки схожих проектів

**Тема**: Написати 2d проект, в якому присутні анімації, скріпти, своя фізика, і якийсь функціонал для гри.

Гра написана наUnity, грати в неї можуть абсолютно всі.

Сенс в тому щоб складати фігури без пропуску в ряду

Ігровий процес полягає в тому наскільки довго ви зможете складати тетріс, адже чим більше рядів зникло тим збільшуться рівень і фігури падають швидше.

Фігури можна ставити будь як та перевертатии в рамках ішрового поля.Основним функціоналом являється те що кожена наступна фігура може не підходити для завершення ряду.

Спеціальні об’єкти тут це сама платформа та фігури, які падають

Загальний опис:

Проект має назву **Tetris**

****

Рис 1. tetris

Структура проекту

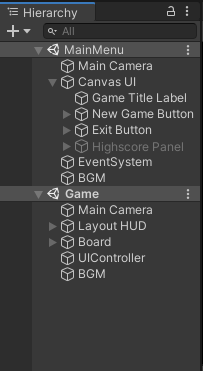


Рис 2. Структура tetris.

Папки з усіма файлами:

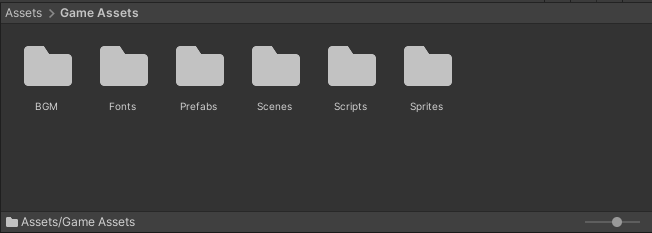


Рис 3. Папки tetris.

1. В папці BGM знаходиться музика , яка граэ на фоні. Це я реалізувавав за допомогоб скрипта(Рис.4)

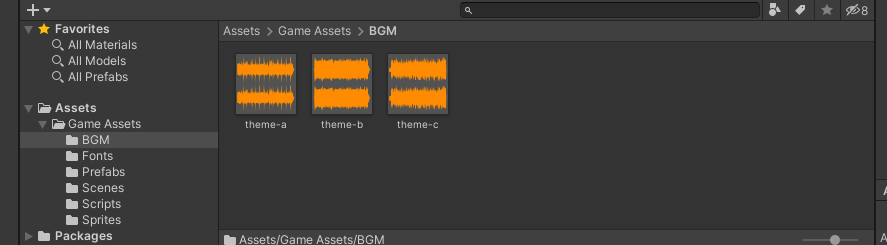


Рис.4 Musik

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class Music : MonoBehaviour {

    public Object[] BGM;

    public AudioSource audioSource;

    public static AudioSource source;

    private bool tryingChange = false;

    public static int nextMusic;

    void Awake() {

        if (source == null) {

            source = audioSource;

            DontDestroyOnLoad(this); // to no restart music on a new game

            randomInitialization();

            source.Play();

        }

    }

    // get the first music by random number

    void randomInitialization() {

        nextMusic = Random.Range(0, BGM.Length);

        source.clip = BGM[nextMusic] as AudioClip;

    }

    // select next music and increment nextMusic by circular reference

    void selectNextMusic(){

        source.clip = BGM[nextMusic] as AudioClip;

        nextCircularPlaylist();

    }

    // Create a circular reference: 0 1 2 0 1 2 0 1 2 ...

    void nextCircularPlaylist() {

        nextMusic = Utils.Mod(nextMusic + 1, BGM.Length);

    }

    // Select the next music and play it

    void playNextMusic() {

        selectNextMusic();

        source.Play();

    }

    // this avoid the behavior of start a new music

    // when the unity stop by desfocusing

    // So we wait for 1 second before change for new music

    // after isPLaying is false

    IEnumerator tryChange() {

        tryingChange = true;

        yield return new WaitForSeconds(1);

        if (!source.isPlaying) {

            playNextMusic();

        }

        tryingChange = false;

    }

    // Update is called once per frame

    void Update () {

        if (!tryingChange && !source.isPlaying) {

            StartCoroutine(tryChange());

        }

    }

}

1. В папціі Fonts лише один фон, за допомогою якого використовую надписи в початковому вікні програми.Всі мої префаби в папці Prefabs.(Рис.5)

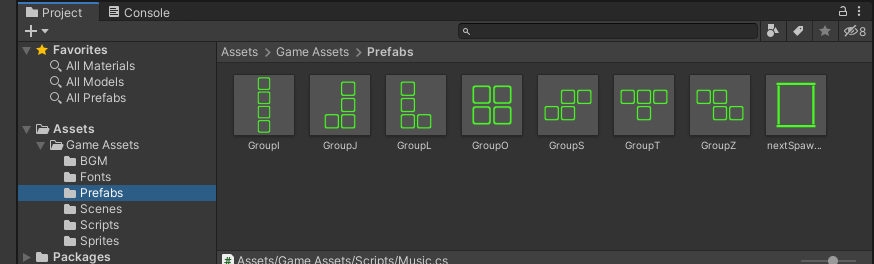
****

Рис.5 Prefabs

1. Всі скріпти в папці Scripts

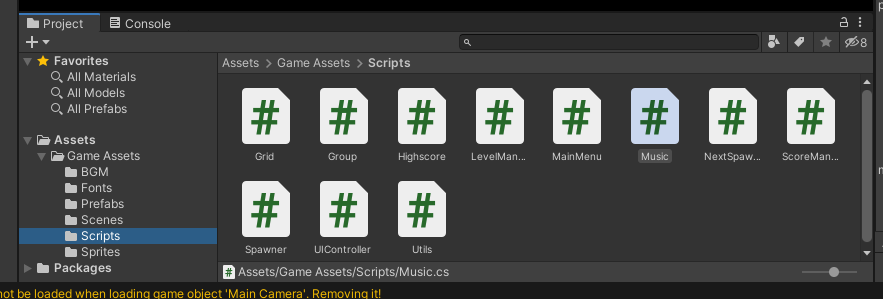


Рис 6. Scriptes.

Розберемо кілька головних скриптів(Описи їх роботит в коментарях):

* 1. Grid –сітка:

1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4. // Цей клас обробляє всю логіку щодо видалення рядків, коли користувач
5. // отримано повний рядок з блоками. А також властивість Grid.grid
6. // зберігає на матриці стан гри, якщо в заданому положенні (x, y)
7. // є блок чи ні.
8. public class Grid : MonoBehaviour {
9. // Сама сітка
10. public static int w = 10;
11. public static int h = 20;
12. // сітка, що зберігає елемент Transform
13. public static Transform[,] grid = new Transform[w, h];
14. // перетворюємо дійсний вектор у дискретні координати за допомогою Mathf.Round
15. public static Vector2 roundVector2(Vector2 v) {
16. return new Vector2 (Mathf.Round (v.x), Mathf.Round (v.y));
17. }
18. // перевіряємо, чи є якийсь вектор у межах гри (межі ліворуч, праворуч та вниз)
19. public static bool insideBorder(Vector2 pos) {
20. return ((int)pos.x >= 0 &&
21. (int)pos.x < w &&
22. (int)pos.y >= 0);
23. }
24. // знищуємо рядок у рядку y
25. public static void deleteRow(int y) {
26. for (int x = 0; x < w; x++) {
27. Destroy(grid[x, y].gameObject);
28. grid[x, y] = null;
29. }
30. }
31. // Щоразу, коли рядок видалявся, вищезазначені рядки повинні падати внизу на одну одиницю.
32. // Наступна функція подбає про це:
33. public static void decreaseRow(int y) {
34. for (int x = 0; x < w; x++) {
35. if (grid[x, y] != null) {
36. // рухаємось одним донизу
37. grid[x, y - 1] = grid[x, y];
38. grid[x,y] = null;
39. // Оновлення позиції блоку
40. grid[x, y-1].position += new Vector3(0, -1, 0);
41. }
42. }
43. }
44. // щоразу, коли рядок видаляється, усі вищезазначені рядки слід зменшувати на 1
45. public static void decreaseRowAbove(int y) {
46. for (int i = y; i < h; i++) {
47. decreaseRow(i);
48. }
49. }
50. // перевіряємо, чи рядок заповнений, а потім може бути видалений (оцінка +1)
51. public static bool isRowFull(int y){
52. for (int x = 0; x < w; x++) {
53. if (grid[x, y] == null) {
54. return false;
55. }
56. }
57. return true;
58. }
59. public static void deleteFullRows() {
60. for (int y = 0; y < h; y++) {
61. if (isRowFull(y)) {
62. deleteRow(y);
63. decreaseRowAbove(y + 1);
64. // додаємо нові бали, щоб набрати, коли рядок буде видалено
65. ScoreManager.score += (h - y) \* 10;
66. --y;
68. }
69. }
70. }
72. }

3.2 Group.cs:

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class Group : MonoBehaviour {

   // час останнього випадання елемента, що використовується для автоматичного падіння після

     // час, параметризований за `рівнем`

    private float lastFall;

  // час останнього натискання клавіші для обробки поведінки тривалого натискання

    private float lastKeyDown;

    private float timeKeyPressed;

    public void AlignCenter() {

        transform.position += transform.position - Utils.Center(gameObject);

    }

    bool isValidGridPos() {

        foreach (Transform child in transform) {

            Vector2 v = Grid.roundVector2(child.position);

            // не всередині кордону?

            if(!Grid.insideBorder(v)) {

                return false;

            }

           // Блок в комірці сітки ?

            if (Grid.grid[(int)(v.x), (int)(v.y)] != null &&

                Grid.grid[(int)(v.x), (int)(v.y)].parent != transform) {

                return false;

            }

        }

        return true;

    }

 // оновлення сітки

    void updateGrid() {

       // Видалити старі обєкти із сітки

        for (int y = 0; y < Grid.h; ++y) {

            for (int x = 0; x < Grid.w; ++x) {

                if (Grid.grid[x,y] != null &&

                    Grid.grid[x,y].parent == transform) {

                    Grid.grid[x,y] = null;

                }

            }

        }

        insertOnGrid();

    }

    void insertOnGrid() {

       // додаємо нових обєктів до сітки

        foreach (Transform child in transform) {

            Vector2 v = Grid.roundVector2(child.position);

            Grid.grid[(int)v.x,(int)v.y] = child;

        }

    }

    void gameOver() {

        Debug.Log("GAME OVER!");

        while (!isValidGridPos()) {

            transform.position  += new Vector3(0, 1, 0);

        }

        updateGrid(); // щоб не перекривати недійсні групи

        enabled = false; // вимкнути скрипт при смерті

        UIController.gameOver(); // активна панель Game Over

        Highscore.Set(ScoreManager.score); // встановити рекорд

    }

   // Використовуй це для ініціалізації

    void Start () {

        lastFall = Time.time;

        lastKeyDown = Time.time;

        timeKeyPressed = Time.time;

        if (isValidGridPos()) {

            insertOnGrid();

        } else {

            Debug.Log("KILLED ON START");

            gameOver();

        }

    }

    void tryChangePos(Vector3 v) {

        transform.position += v;

        // See if valid

        if (isValidGridPos()) {

            updateGrid();

        } else {

            transform.position -= v;

        }

    }

    void fallGroup() {

       // модифікувати

        transform.position += new Vector3(0, -1, 0);

        if (isValidGridPos()){

           // Дійсний. Оновити сітку ... ще раз

            updateGrid();

        } else {

           // це недійсне. повернути

            transform.position += new Vector3(0, 1, 0);

            // Очистити заповнені горизонтальні лінії

            Grid.deleteFullRows();

            // Вимкнути сценарій

            FindObjectOfType<Spawner>().spawnNext();

            enabled = false;

        }

        lastFall = Time.time;

    }

   // getKey, якщо натиснути зараз на довше натиснути на 0,5 секунди | якщо це істина, застосовуйте клавішу кожного 0,05f під час натискання

    bool getKey(KeyCode key) {

        bool keyDown = Input.GetKeyDown(key);

        bool pressed = Input.GetKey(key) && Time.time - lastKeyDown > 0.5f && Time.time - timeKeyPressed > 0.05f;

        if (keyDown) {

            lastKeyDown = Time.time;

        }

        if (pressed) {

            timeKeyPressed = Time.time;

        }

        return keyDown || pressed;

    }

   // Оновлення викликається один раз на кадр

    void Update () {

        if (UIController.isPaused) {

            return; // нічого не робити

        }

        if (getKey(KeyCode.LeftArrow)) {

            tryChangePos(new Vector3(-1, 0, 0));

        } else if (getKey(KeyCode.RightArrow)) {  // Рухатися вправо

            tryChangePos(new Vector3(1, 0, 0));

        } else if (getKey(KeyCode.UpArrow) && gameObject.tag != "Cube") { // Обертати

            transform.Rotate(0, 0, -90);

         // перевірити, чи дійсно

            if (isValidGridPos()) {

             // він дійсний. Оновити сітку

                updateGrid();

            } else {

               // це недійсне. повернути

                transform.Rotate(0, 0, 90);

            }

        } else if (getKey(KeyCode.DownArrow) || (Time.time - lastFall) >= (float)1 / Mathf.Sqrt(LevelManager.level)) {

            fallGroup();

        } else if (Input.GetKeyDown(KeyCode.Space)) {

            while (enabled) {  // падати до дна

                fallGroup();

            }

        }

    }

}

3.3 MainMenu: using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.SceneManagement;

using UnityEngine.UI;

public class MainMenu : MonoBehaviour {

   // це панелі кнопок

    public GameObject[] buttons;

    public GameObject highScorePanel;

    public Text highscoreText;

    private int buttonSelected;

    private int numberOfButtons;

    void Awake() {

        numberOfButtons = buttons.Length;

        buttonSelected = 0;

        SelectNewGame();

        if (Highscore.highscore > 0) {

            highscoreText.text = Highscore.Get();

            highScorePanel.SetActive(true);

        }

    }

    public void NewGame() {

        SceneManager.LoadScene(1);

    }

    public void Exit() {

        Application.Quit ();

    }

    void openSelected() {

        if (buttonSelected == 0) {

            NewGame();

        } else if (buttonSelected == 1) {

            Exit();

        }

    }

    public void SelectNewGame() {

        buttons[0].SetActive(true);

        buttons[1].SetActive(false);

        buttonSelected = 0;

    }

    public void SelectExitGame() {

        buttons[1].SetActive(true);

        buttons[0].SetActive(false);

        buttonSelected = 1;

    }

    void changePanel(int direction) {

        buttons[buttonSelected].SetActive(false);

        buttonSelected = Utils.Mod(buttonSelected + direction,  numberOfButtons);

        buttons[buttonSelected].SetActive(true);

    }

    public void Update() {

        // меню перегляду стрілки

        if (Input.GetKeyDown(KeyCode.UpArrow)) {

            changePanel(-1); // вгору

        } else if (Input.GetKeyDown(KeyCode.DownArrow)) {

            changePanel(1); // вниз

        } else if (Input.GetKeyDown(KeyCode.Return)) {

            openSelected(); // відкрити вибрану кнопку

        } else if (Input.GetKeyDown(KeyCode.Escape)) {

            Application.Quit(); // покинути

        }

    }

}

Та інші скрипти

Під час виконання проекту зіткнувся з великою кількість іноді простих, а іноді не дужу простих проблем, які потребували загалом кілька днів для їх усунення та розробки додатка таким, яким я його хотів бачити.