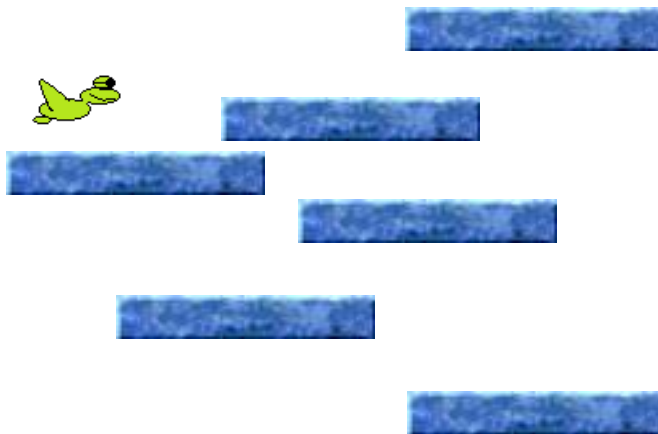


# 介紹-閃躲遊戲

我們的主題是跳躍遊戲和閃躲遊戲，製作的動機是疫情宅在家沒事做想找點遊戲玩，而且現在越來越多 3A 大作，各種華麗的特效、精緻的場景和角色、特殊的操作方式，讓人不禁懷念以前小遊戲樸素的場景，簡單明瞭的控制，加上覺得老師上課範例的遊戲很有趣，我們剛好藉由這次的機會來製作一個遊戲。我們是使用 unity 來製作，因為 unity 裡面有多種專門為了製作遊戲而有的功能。遊戲的玩法很簡單，使用空白件來跳躍來閃躲障礙物，若觸碰到障礙物則結束遊戲。



# 遊戲內容物

我們使用 unity 來加入腳色的動畫、音效、背景音樂、腳色碰撞判斷、重力效果。

## 問題

隨機生成的障礙物會有重疊的問題

遊戲太過單調容易玩膩

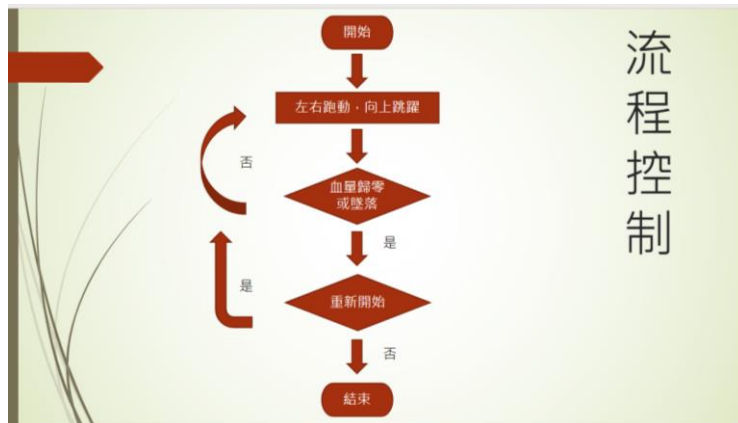
# 跳躍小遊戲

## 1. Introduction

時下疫情嚴重，開始出現在家工作或停工、各所學校開始遠距教學，在家沒事做時想玩個遊戲，但現今遊戲產業發展繁盛，多樣化的遊戲，投入的成本和時間越來越多，像是 GTA5 仿真的場景和角色、多功能的操作、3D 的視角移動，APEX 3D 化的射擊視角，讓人不禁懷念以前小遊戲樸素的場景，簡單明瞭的控制，所以我們打算開發 2D 小遊戲。

## 2. 系統架構

流程圖：



開始畫面：



```

3         public void StartGame()
4         {
5             Time.timeScale = 1f;
6             startButton.SetActive(false);
7         }
8     }
9

```

(程式碼)

角色控制:方向鍵上為跳躍，左右鍵為向左移動或向右移動。

```

void Update()
{
    if(Input.GetKey(KeyCode.RightArrow))
    {
        transform.Translate(moveSpeed * Time.deltaTime, 0, 0);
        render.flipX = false;
        anim.SetBool("run", true);
    }
    else if(Input.GetKey(KeyCode.LeftArrow))
    {
        transform.Translate(-moveSpeed * Time.deltaTime, 0, 0);
        render.flipX = true;
        anim.SetBool("run", true);
    }
    else
    {
        anim.SetBool("run", false);
    }

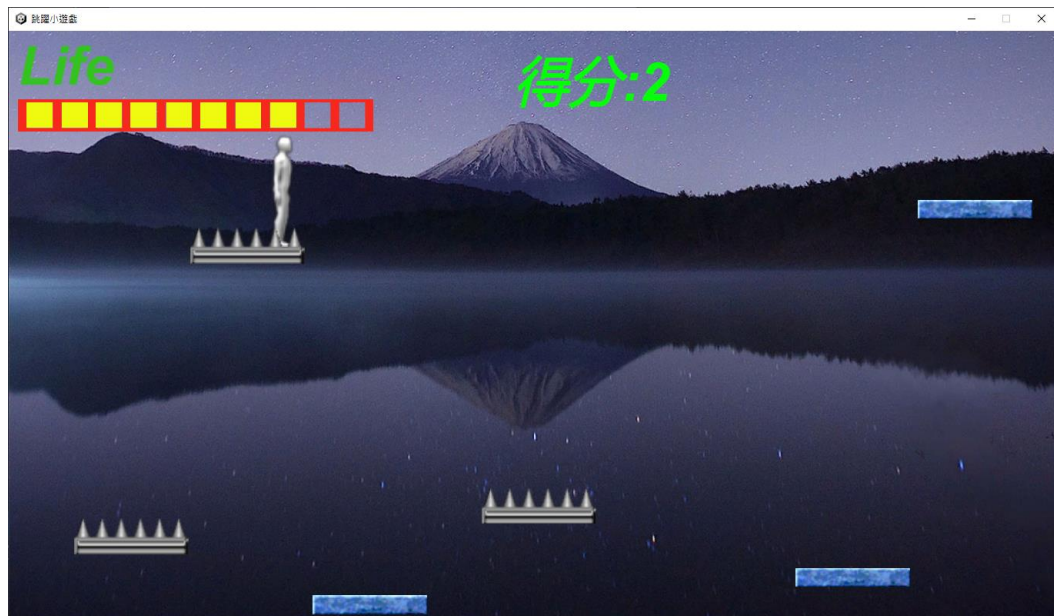
    if(Input.GetKey(KeyCode.UpArrow) && anim.GetBool("onGround") == true)
    {
        GetComponent<Rigidbody2D>().AddForce (Vector2.up * 500);
        anim.SetBool("jumping", true);
        anim.SetBool("onGround", false);
    }
    UpdateScore();
}

```

(程式碼)

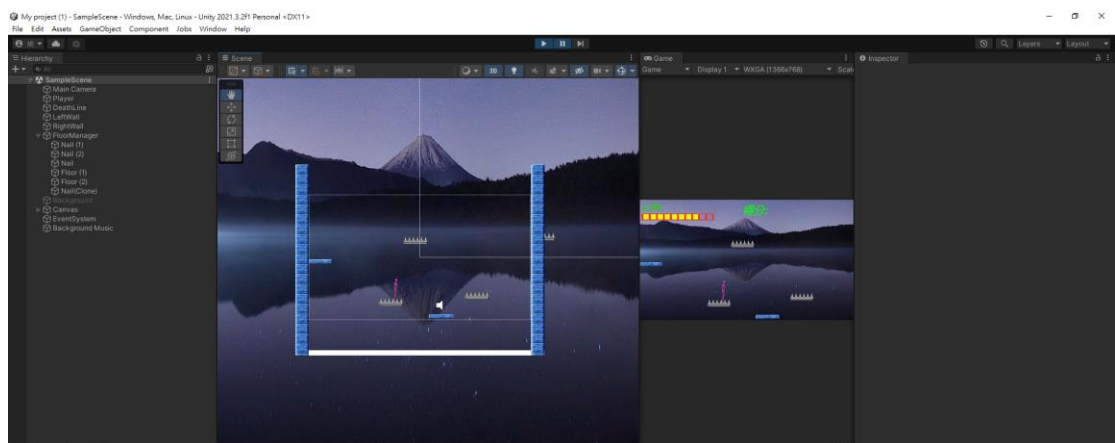
跳躍動作只能在地面上才能觸發，所以加了布林值 onGround 做判斷。

地板物件:



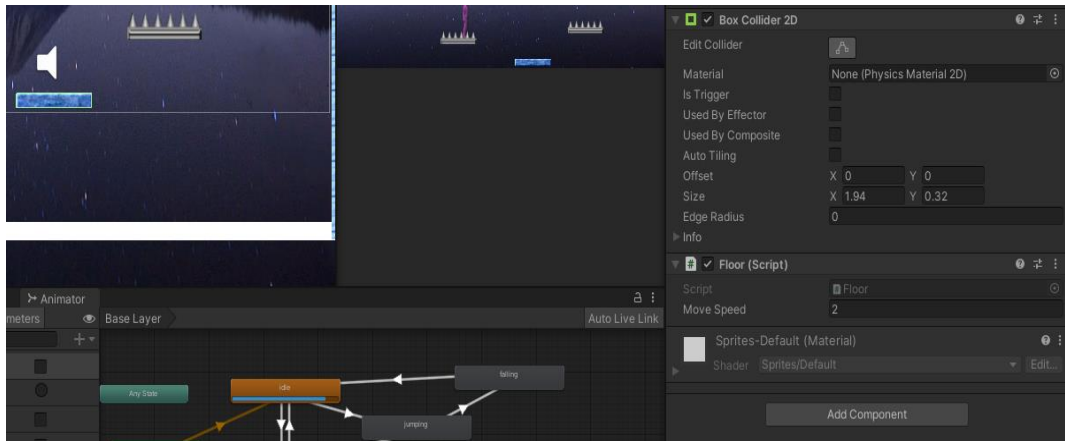
兩邊隔牆:

讓玩家在畫面中移動，避免掉出界外。

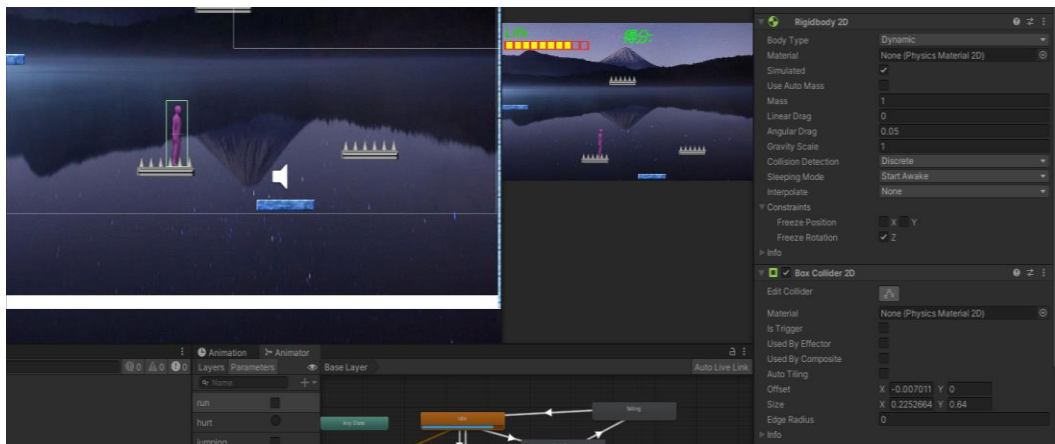


碰撞判定:

地板四周以 Box Collider 2D 進行判定。



角色一樣以 Box Collider 2D 進行判定，Rigidbody 2D 則是控制角色受到重力影響。



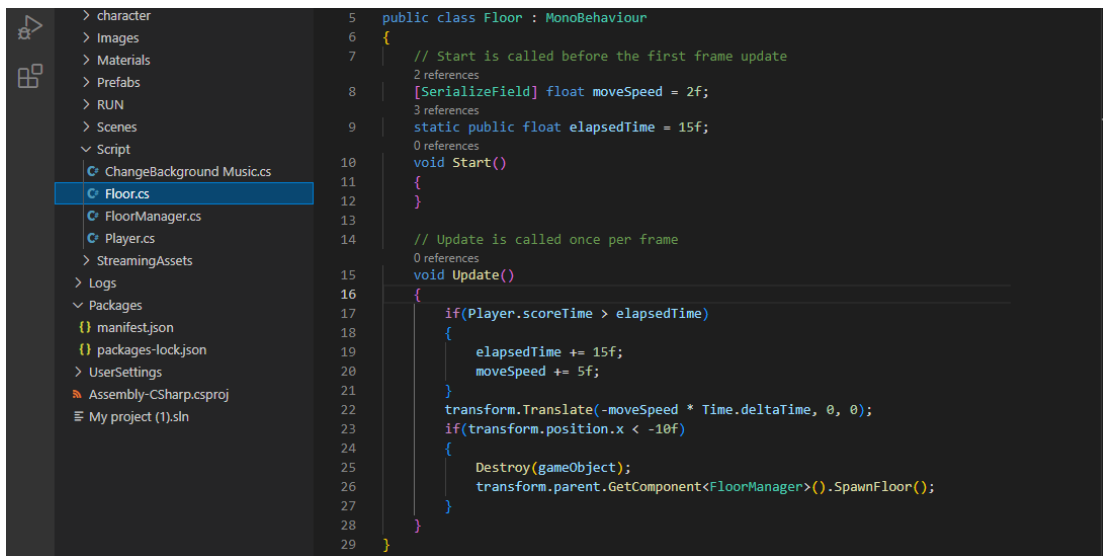
```

0 references
void OnCollisionEnter2D(Collision2D other) |
{
    if(other.gameObject.tag == "Floor")
    {
        if(other.contacts[0].normal == new Vector2(0f, 1f))
        {
            ModifyHp(1);
            anim.SetBool("onGround", true);
            anim.SetBool("jumping", false);
        }
    }
    else if(other.gameObject.tag == "Nail")
    {
        if(other.contacts[0].normal == new Vector2(0f, 1f))
        {
            ModifyHp(-2);
            anim.SetTrigger("hurt");
            anim.SetBool("onGround", true);
            anim.SetBool("jumping", false);
        }
    }
}

```

(程式碼)

地板移動速度隨著遊戲時間增加而加快。



The screenshot shows the Unity IDE with the 'Floor.cs' script selected in the Hierarchy panel. The script is a C# class that inherits from 'MonoBehaviour'. It contains two methods: 'Start()' and 'Update()'. The 'Start()' method is empty. The 'Update()' method is called once per frame and contains logic to increase the 'moveSpeed' variable over time and to destroy the floor object when it reaches a certain position.

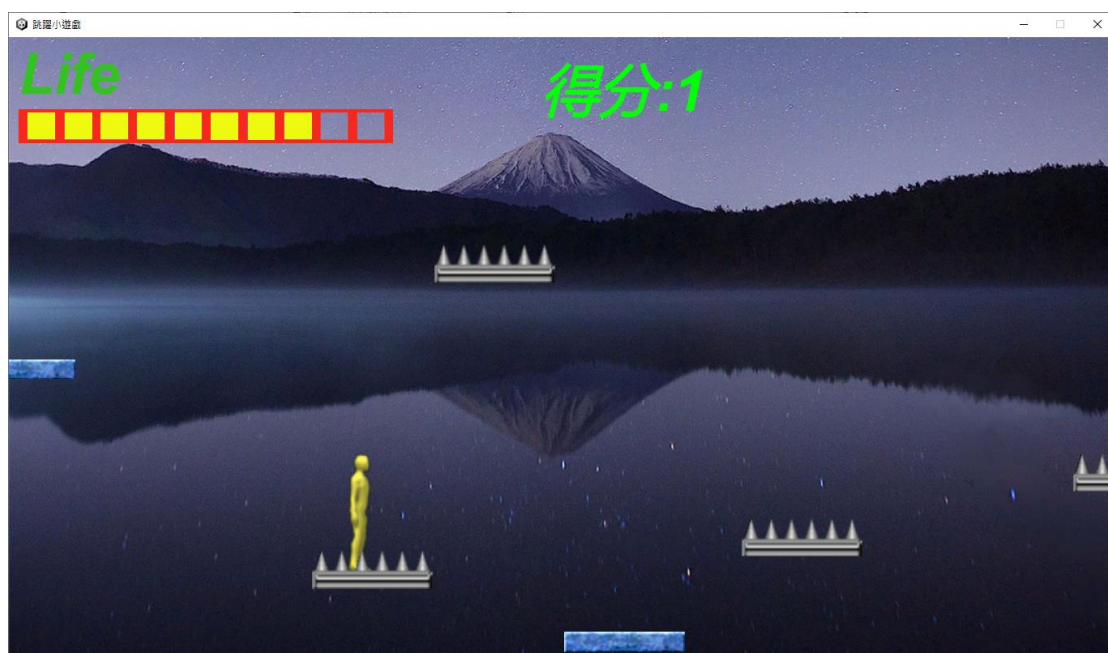
```
5 public class Floor : MonoBehaviour
6 {
7     // Start is called before the first frame update
8     [SerializeField] float moveSpeed = 2f;
9     static public float elapsedTime = 15f;
10    void Start()
11    {
12    }
13
14    // Update is called once per frame
15    void Update()
16    {
17        if(Player.scoreTime > elapsedTime)
18        {
19            elapsedTime += 15f;
20            moveSpeed += 5f;
21        }
22        transform.Translate(-moveSpeed * Time.deltaTime, 0, 0);
23        if(transform.position.x < -10f)
24        {
25            Destroy(gameObject);
26            transform.parent.GetComponent<FloorManager>().SpawnFloor();
27        }
28    }
29 }
```

(程式碼)

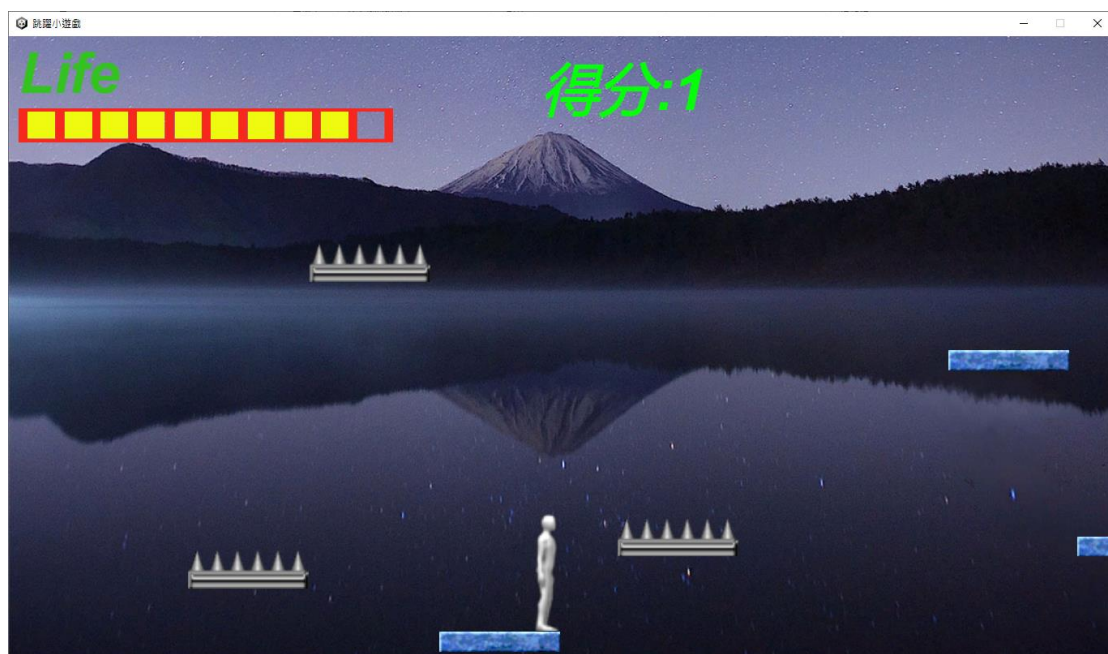
血量條:



碰到尖刺地板時會扣 2 格血量，一般地板則會回一格血量。



(扣 2 格血量)



(回了一格血量)

```

2 references
void ModifyHp(int num)
{
    Hp += num;
    if(Hp > 10) Hp = 10;
    else if(Hp <= 0)
    {
        Hp = 0;
        Die();
    }
    ModifyHpBar();
}

1 reference
void ModifyHpBar()
{
    for(int i = 0; i < HpBar.transform.childCount; i++)
    {
        if(Hp > i)
        {
            HpBar.transform.GetChild(i).gameObject.SetActive(true);
        }
        else
        {
            HpBar.transform.GetChild(i).gameObject.SetActive(false);
        }
    }
}

```

(修改血量條)

```

0 references
void OnCollisionEnter2D(Collision2D other)
{
    if(other.gameObject.tag == "Floor")
    {
        if(other.contacts[0].normal == new Vector2(0f, 1f))
        {
            ModifyHp(1);
            anim.SetBool("onGround", true);
            anim.SetBool("jumping", false);
        }
    }
    else if(other.gameObject.tag == "Nail")
    {
        if(other.contacts[0].normal == new Vector2(0f, 1f))
        {
            ModifyHp(-2);
            anim.SetTrigger("hurt");
            anim.SetBool("onGround", true);
            anim.SetBool("jumping", false);
        }
    }
}

```

(扣血量)

得分:

每過一段時間可以得到分數。



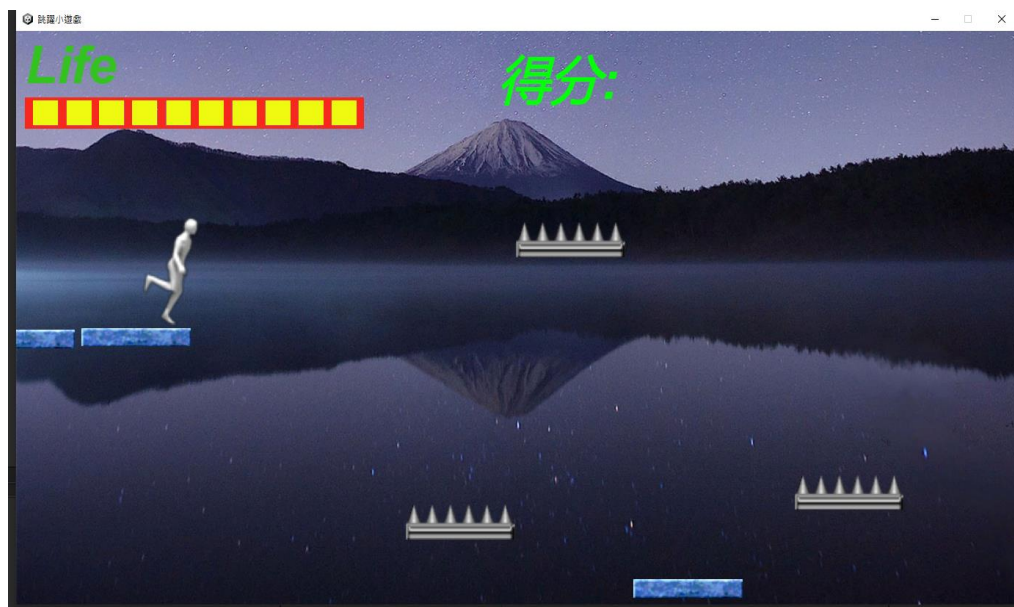
```

void UpdateScore()
{
    scoreTime += Time.deltaTime;
    if(scoreTime > Floor.elapsedTime)
    {
        scoreChange += 1;
    }
    if(scoreTime > 2f)
    {
        score += scoreChange;
        scoreTime = 0f;
        scoreText.text = "得分:" + score.ToString();
    }
}

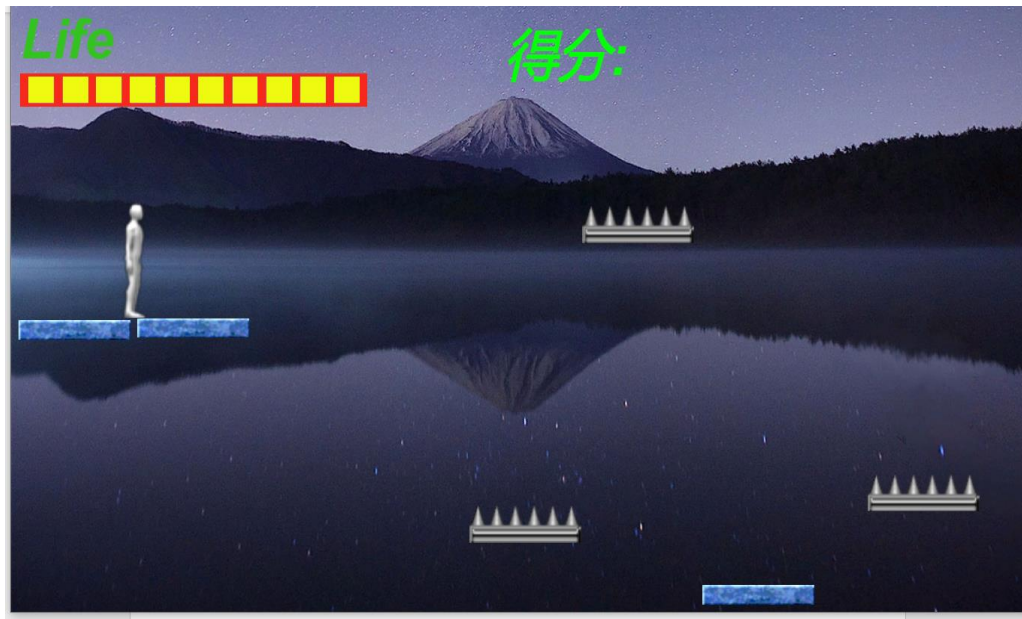
```

(程式碼)

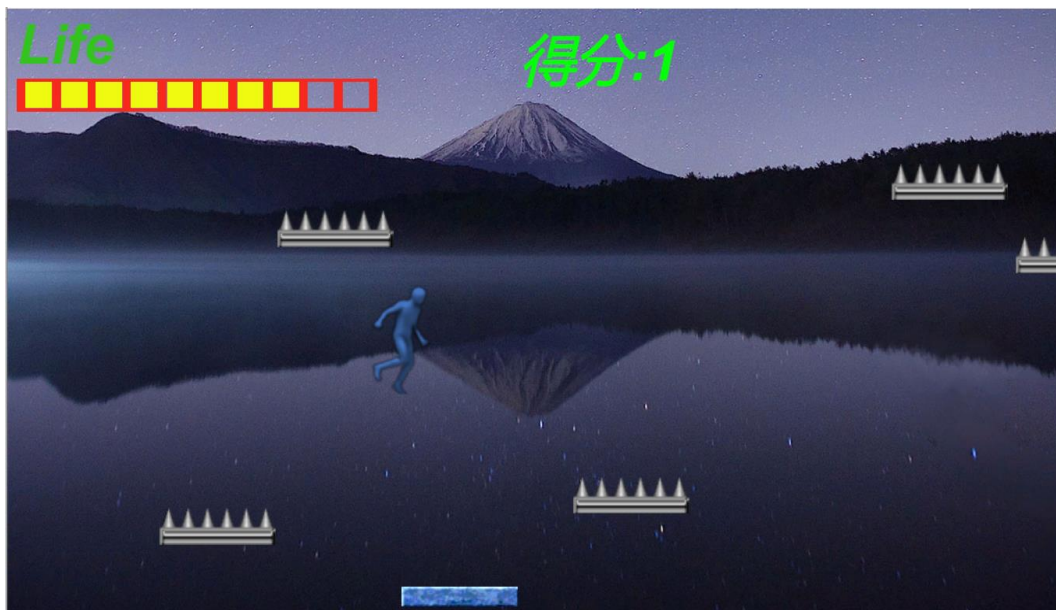
動畫:



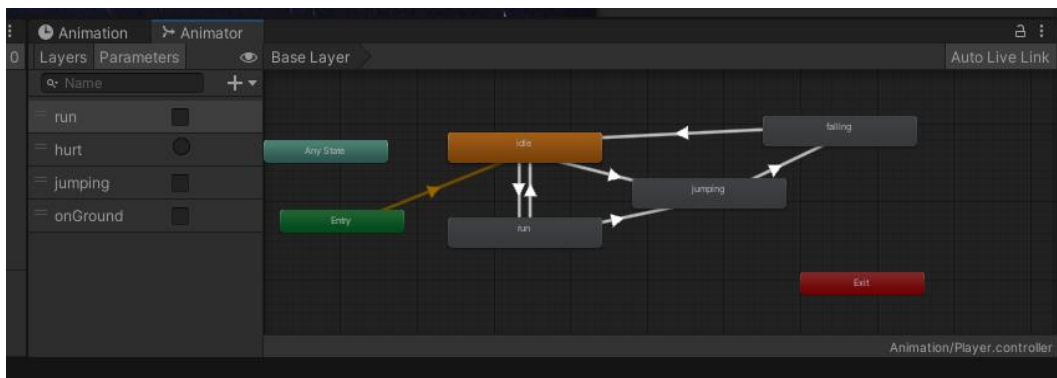
(跑動時的動畫)



(静止時)



(跳躍和扣血時)



(動畫控制器和參數)

死亡條件:

沒踩到地板墜落或血量歸零。





(墜落)

```
0 references
void OnTriggerEnter2D(Collider2D other)
{
    if(other.gameObject.tag == "DeathLine") Die();
}
```

(程式碼)



(血量歸零)

```
2 references
void Die()
{
    Time.timeScale = 0f;
    replayButton.SetActive(true);
}
```

(程式碼)

點擊重新開始即可重玩。

```
0 references
public void Replay()
{
    Time.timeScale = 1f;
    SceneManager.LoadScene("SampleScene");
}
```

(程式碼)

背景音樂:

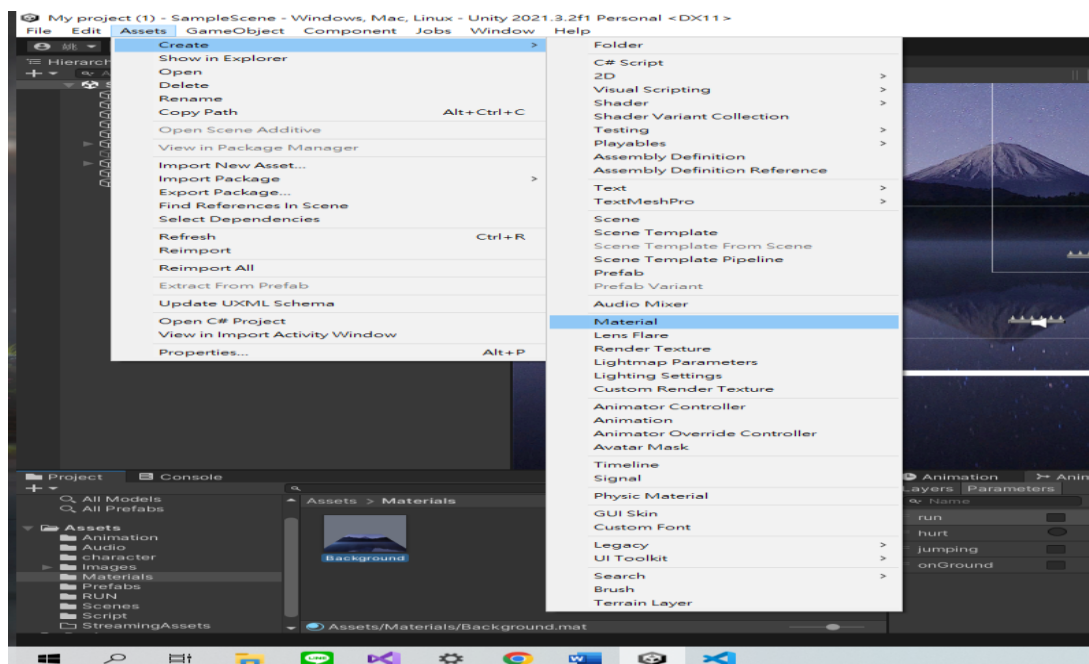
進入遊戲即可聽到循環音樂。

```
0 references
void Start()
{
    AudioSource source = GameObject.FindGameObjectWithTag("Background Music");
    source.Stop();
    source.pitch = pitch;
    source.Play();
}
```

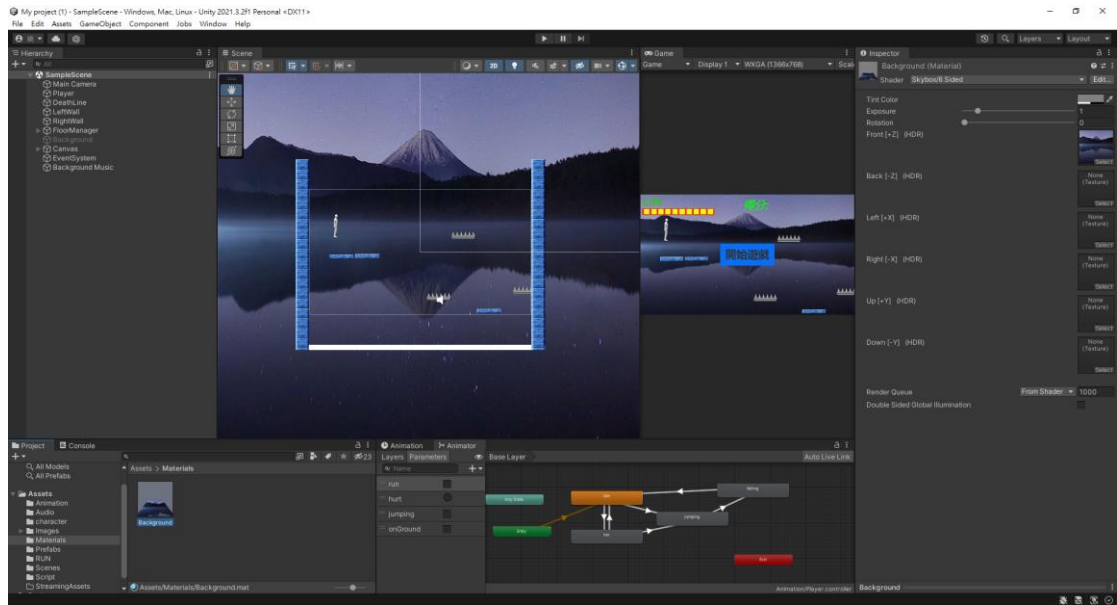
(程式碼)

背景圖片:

使用 Material 內的 Skybox 當作背景物件，在遊戲期間都可以看見此背景。







## 4.未來期望

- 增加不同種類的障礙物。
- 新增不同的音樂或讓使用者自行加入。
- 能在得到一定分數後轉換關卡、場景。
- 開發多人遊玩功能。
- 加入一些道具，像是可以復活或無敵一段時間。
- 跳躍時能更順暢、較符合物理規則。
- 能讓使用者更換背景圖片。

## 5.總結

啟動遊戲後進入開始畫面，遊戲會撥放背景音樂，按下開始按鈕即開始遊戲，期間可以左右移動和往上跳躍，隨著時間增加得分也會累積，進行得越久難度愈高得分也就越多，但地板會隨著時間移動，並且有碰撞判定，如果沒踩到地板則角色會受到重力影響向下墜落，超出界外時及判定死亡，地板還有尖刺種類，踩到會扣 2 滴血量，一般地板則會回復一滴，血量歸零時一樣判定死亡，死亡後會跳出重新按鈕，點擊即可重新開始。

## 參考文獻:

- [1] [https://www.youtube.com/watch?v=nPW6tKeapsM&list=WL&index=5&ab\\_channel=GrandmaCan-%E6%88%91%E9%98%BF%E5%AC%A4%E9%83%BD%E6%9C%83](https://www.youtube.com/watch?v=nPW6tKeapsM&list=WL&index=5&ab_channel=GrandmaCan-%E6%88%91%E9%98%BF%E5%AC%A4%E9%83%BD%E6%9C%83)。
- [2] <https://forum.gamer.com.tw/Co.php?bsn=60602&sn=4579>。
- [3] <https://forum.gamer.com.tw/Co.php?bsn=60602&sn=2484>。
- [4] <https://ithelp.ithome.com.tw/articles/10273183>。
- [5] <https://ithelp.ithome.com.tw/articles/10187347>。
- [6] <https://zhuanlan.zhihu.com/p/371100199>。