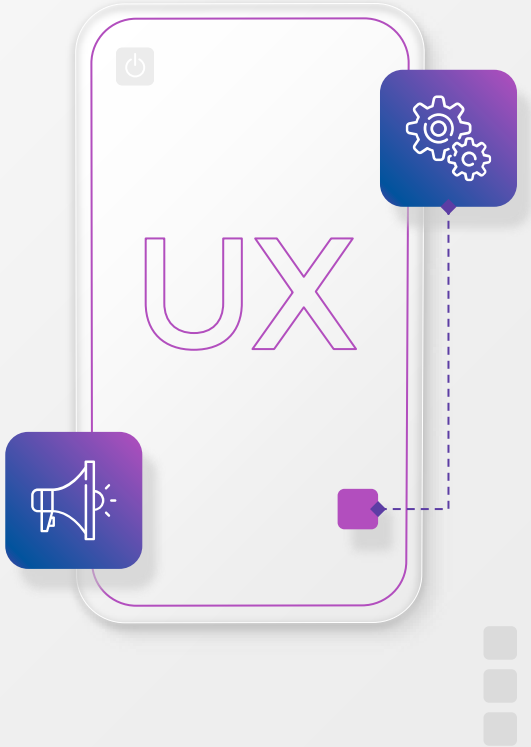
行動裝置小遊戲 — 小朋友下樓梯

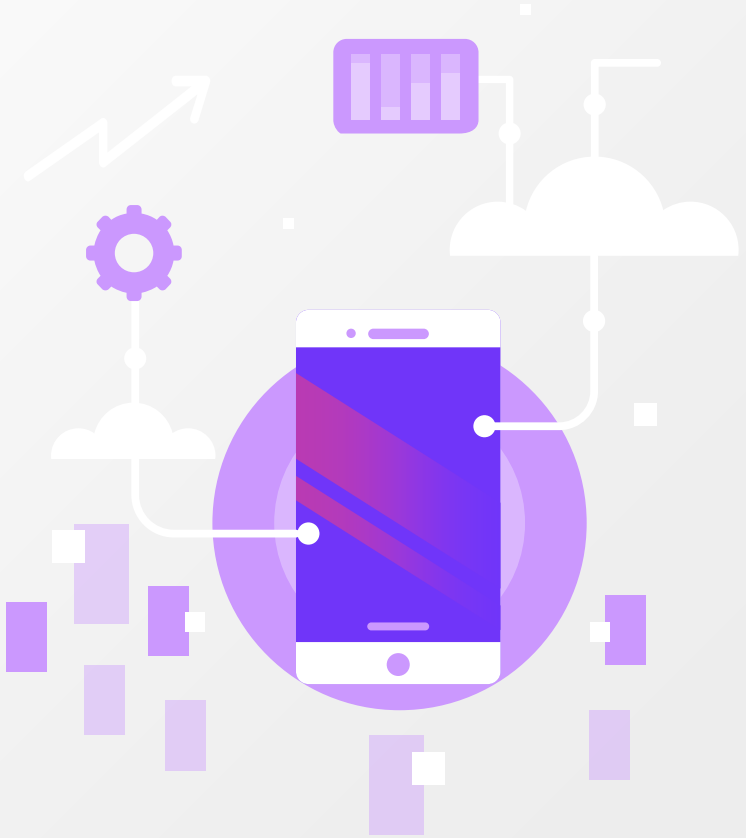
110316108 黃麗文
110316124 曾琨荃
110316129 沈傳諺



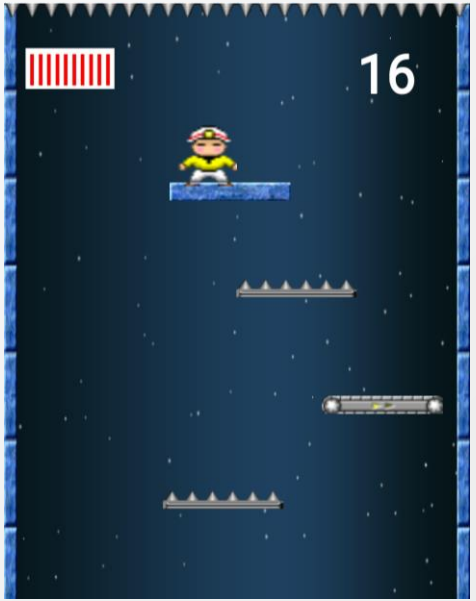
目錄

01

遊戲說明



遊戲說明



- 小朋友下樓梯是一款平台遊戲。
- 平台會向上移動，在屏幕的頂部有一堆尖刺。
- 遊戲目標：

1. 控制小朋友從一個平台移動到另一個平台。
2. 控制速度讓小朋友不會碰到尖刺或掉落到底部。
3. 碰到尖刺會扣生命值。耗盡生命值，遊戲即結束。
4. 存活越久，分數越高。

遊戲說明

藍色平台
(普通平台 +1分)

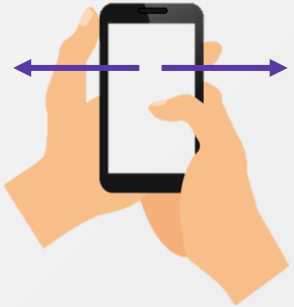
傳送帶
(人物站在上面會向左右移動 +2分)

尖刺
(碰到天花板 -5滴血
碰到平台上 -3滴血
平安經過 +3分)

彈簧板
(人物在上面會彈跳 +2分)

玩法說明

控制小朋友移動到安全的平台，盡量存活下去。



1. 按住左右螢幕控制人物往兩側移動。
2. 碰到尖刺會扣生命值。
3. 耗盡生命值或掉落到底部，遊戲結束。
4. 依據相對應的平台得分，存活越久，分數越高。

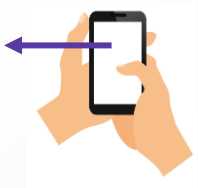
遊戲影片


小朋友下樓梯
開始遊戲
遊戲說明
控制人物移動到安全平台並存活下去。
1. 按住左右螢幕控制人物往兩側移動。
2. 碰到尖刺會扣生命值。
(天花板-5 平台-3)
3. 耗盡生命值或掉到底部，遊戲結束。
4. 存活越久，分數越高。
藍色普通平台 +1
傳送帶/彈簧板 +2
(經過)尖刺 +3

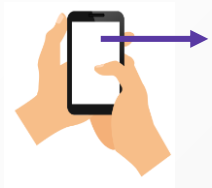
02

角色設計

角色設計



按住螢幕左滑 — 小朋友向左移動



按住螢幕右滑 — 小朋友向右移動

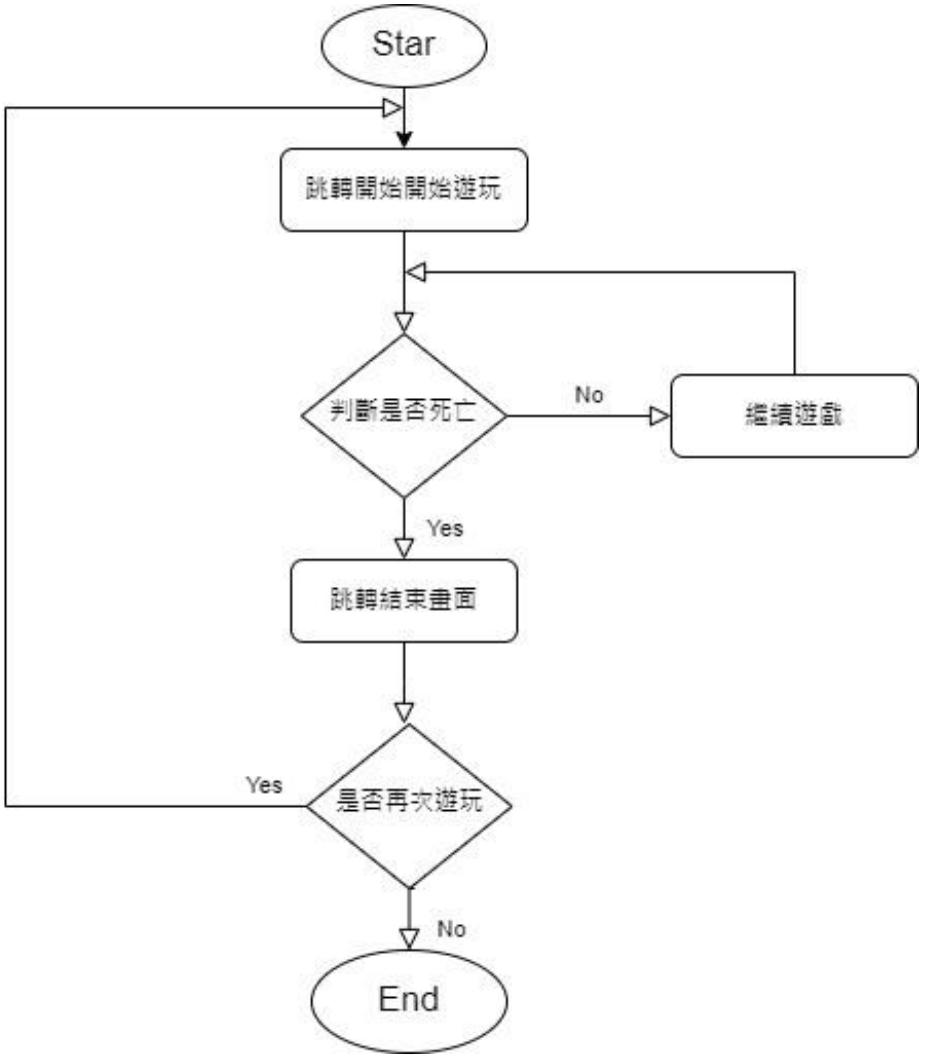
碰到彈簧板 — 小朋友向上跳躍

03

GameLoop

流程圖

GameLoop 流程圖



04

遊戲邏輯

遊戲邏輯

player.cs

控制人物移動偵測

```
if (mx < this.x - 30)  //滑鼠點在人物左邊向左移
{
    dx = -10;
    frame = 0 + g;
}
else if (mx > this.x + 30)  //滑鼠點在人物右邊,向右移
{
    dx = 10;
    frame = 9 + g;
}
else  //滑鼠在人物附近就停止
{
    dx = 0;
    frame = 8;
}
```

遊戲邏輯

player.cs

碰到傳送帶
改變移動速度

```
if (conveyor_type == 1)
{
    if (mx < this.x - 20)  //滑鼠點在人物左邊向左移
    {
        dx = -15;
        frame = 8;
    }
    else if (mx > this.x + 20)  //滑鼠點在人物右邊,向右移
    {
        dx = 5;
        frame = 8;
        if (sss == true)
        {
            dx = -10;
            frame = 8;
        }
    }
    else  //滑鼠在人物附近就停止
    {
        dx = -10;
        frame = 8;
        sss = true;
    }
}
```

```
else if (conveyor_type == 0)
{
    if (mx < this.x - 20)  //滑鼠點在人物左邊向左移
    {
        dx = -5;
        frame = 8;
    }
    else if (mx > this.x + 20)  //滑鼠點在人物右邊,向右移
    {
        dx = 15;
        frame = 8;
        if (sss == true)
        {
            dx = 10;
            frame = 8;
        }
    }
    else  //滑鼠在人物附近就停止
    {
        dx = 10;
        frame = 8;
        sss = true;
    }
}
```

遊戲邏輯

GamePage.xaml.cs

隨機產生不同平台

```
int generate = 0; //出現磚塊時間
int Attack_twice_in_a_row = 0; //不要一直都出現刺
int add_tpye = 0; //新增的磚塊類型
void add_brick()
{
    if (generate <= 0) //隨機時間產生新的平台
    {
        float xx, yy;
        xx = r.Next(770); //亂數產生座標及位移向量
        yy = 0;
        int temp = s.Count();
        int nauls_count = nails.Count();
        int trampoline_count = trampoline.Count();
        int conveyor_right_count = conveyor_right.Count();
        int conveyor_left_count = conveyor_left.Count();
```

```
if (add_tpye == 0) //產生藍色普通平台
{
    if (temp != 0) //將新產生的平台加入串列
    {
        if (Math.Abs(s[temp - 1].x - xx) < 140)
            xx = r.Next(770);
    }
    Stone S = new Stone(xx, yy, -4, this);
    s.Add(S);
    add_tpye = r.Next(0, 5);
    if (int.TryParse(textScore, out int result))
    {
        //轉換成功，result 中為轉換後的整數值
        result = result + 1;
        textScore = result.ToString();
    }
}
```

遊戲邏輯

GamePage.xaml.cs

隨機產生不同平台

```
else if (add_tpye == 1) //產生彈簧板
{
    if (trampoline_count != 0)
    {
        if (Math.Abs(trampoline[trampoline_count - 1].x - xx) < 140)
            xx = r.Next(770);
    }
    Trampoline t = new Trampoline(xx, yy, -4, this);
    trampoline.Add(t);
    add_tpye = r.Next(0, 5);
    if (int.TryParse(textScore, out int result))
    {
        //轉換成功，result 中為轉換後的整數值
        result = result + 2;
        textScore = result.ToString();
    }
}
```

```
else if (add_tpye == 2) //產生尖刺平台
{
    if (nauls_count != 0)
    {
        if (Math.Abs(nails[nauls_count - 1].x - xx) < 140)
            xx = r.Next(770);
    }
    Nails n = new Nails(xx, yy, -4, this);
    nails.Add(n);
    add_tpye = r.Next(0, 5);
    if (int.TryParse(textScore, out int result))
    {
        //轉換成功，result 中為轉換後的整數值
        result = result + 3;
        textScore = result.ToString();
    }
}
```

遊戲邏輯

GamePage.xaml.cs

隨機產生不同平台

```
else if (add_tpye == 3) //產生往右傳送帶
{
    if (conveyor_right_count != 0)
    {
        if (Math.Abs(conveyor_right[conveyor_right_count - 1].x - xx) < 140)
            xx = r.Next(770);
    }
    conveyor_right n = new conveyor_right(xx, yy, -4, this);
    conveyor_right.Add(n);
    add_tpye = r.Next(0, 5);
    if (int.TryParse(textScore, out int result))
    {
        //轉換成功，result 中為轉換後的整數值
        result = result + 2;
        textScore = result.ToString();
    }
}
```

```
        else if (add_tpye == 4) //產生往左傳送帶
        {
            if (conveyor_left_count != 0)
            {
                if (Math.Abs(conveyor_left[conveyor_left_count - 1].x - xx) < 140)
                    xx = r.Next(770);
            }
            conveyor_left n = new conveyor_left(xx, yy, -4, this);
            conveyor_left.Add(n);
            add_tpye = r.Next(0, 5);
            if (int.TryParse(textScore, out int result))
            {
                //轉換成功，result 中為轉換後的整數值
                result = result + 2;
                textScore = result.ToString();
            }
        }
        generate = r.Next(50, 70);
    }
    else
    {
        generate--;
    }
}
```

遊戲邏輯

GamePage.xaml.cs

碰撞偵測

```
foreach (Stone b in s) //判斷接觸藍色平台
{
    if (p.Coll(b))
    {
        p.dy = -4;
        touch_tpye = 0;
        st = 1;
        break;
    }
}
```

```
foreach (Trampoline t in trampoline) //判斷接觸彈簧板
{
    if (p.Coll_Trampoline(t))
    {
        p.dy = -4;
        touch_tpye = 0;
        st = 3;
        break;
    }
}
```

```
foreach (Nails n in nails)
{
    if (p.Coll_nails(n)) //判斷接觸尖刺平台停下
    {
        p.dy = -4;
        touch_tpye = 1;
        st = 2;
        break;
    }
}
```

遊戲邏輯

```
foreach (conveyor_right c_r in conveyor_right) //判斷接觸往右傳送帶
{
    if (p.Coll_conveyor_right(c_r))
    {
        p.dy = -4;
        touch_tpye = 0;
        st = 4;
        break;
    }
}
```

GamePage.xaml.cs

碰撞偵測

```
foreach (conveyor_left c_l in conveyor_left) //判斷接觸往左傳送帶
{
    if (p.Coll_conveyor_left(c_l))
    {
        p.dy = -4;
        touch_tpye = 0;
        st = 5;
        break;
    }
}
```

遊戲邏輯

GamePage.xaml.cs

碰撞偵測

```
if (st == 3) //碰到彈跳
{
    if (jump_voice == 0)
    {
        tmusic.playSound();
        jump_voice = 1;
    }
    p.dy = -20;
    if (stt == 0)
    {
        jump = 10;
        stt = 1;
    }
}
if (jump > 0)
{
    jump_voice = 0;
    jump--;
    if (jump == 0)
    {
        p.dy = 18;
        stt = 0;
    }
}
```

遊戲邏輯

GamePage.xaml.cs

碰撞偵測

```
if (decide == 2) //掉落到底部
{
        string filePath = Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.Personal), filename);
        // 使用 StreamWriter 將字串寫入檔案
        using (StreamWriter writer = new StreamWriter(filePath))
        {
            writer.Write(textScore);
        }
        pageIsActive = false;
    media.Stop();
    await Navigation.PushAsync(new EndPage());
    drop = 0;
}
```

遊戲邏輯

GamePage.xaml.cs

碰撞偵測

```
int blood_count = blood.Count();
blood_count = blood_count - 5; //扣5滴
if (blood_count <= 0 && drop != 0) //碰到天花板
{
    string filePath = Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.Personal), filename);
    // 使用 StreamWriter 將字串寫入檔案
    using (StreamWriter writer = new StreamWriter(filePath))
    {
        writer.Write(textScore);
    }
    pageIsActive = false;
    media.Stop();
    await Navigation.PushAsync(new EndPage());
}
else
{
    string ex = "";
    for (int i = 0; i < blood_count; i++)
    {
        ex += "|";
    }
    blood = ex;
}
```

遊戲邏輯

GamePage.xaml.cs

碰撞偵測

```
int blood_count = blood.Count();
blood_count = blood_count - 3;//扣三滴
if (blood_count <= 0)
{
    string filePath = Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.Personal), filename);
    // 使用 StreamWriter 將字串寫入檔案
    using (StreamWriter writer = new StreamWriter(filePath))
    {
        writer.Write(textScore);
    }
    pageIsActive = false;
    media.Stop();
    await Navigation.PushAsync(new EndPage());

}
else
{
    string ex = "";
    for (int i = 0; i < blood_count; i++)
    {
        ex += "|";
    }
    blood = ex;
}
```

05

感想心得

感想心得

- **110316108 黃麗文**：我負責背景畫面和音樂、遊戲說明及PPT製作。過程中因為還不太熟悉哪些程式碼應該怎麼應用，所以研究了很久，可能一兩個小時才弄好一小部分，但是做出來就很有成就感，也謝謝組員大力的幫忙。
- **110316124 曾琨荃**：這次的遊戲製作我負責的部分是基本的畫面呈現與基本的遊戲機制，做這個遊戲充滿挑戰同時也很有趣，與其他團隊成員合作，保持溝通是至關重要的。
- **110316129 沈傳諺**：我負責功能方塊、畫面切換以及血量、分數紀錄的設計，讓我體會到設計一個遊戲的不容易。與其他團隊成員的緊密合作使我學到了溝通和協調的重要性，加強了我的團隊協作和解決問題的能力。

Thanks