

---

# 平台遊戲之程式設計

---

10711042 臺灣藝術大學 旁聽生 范芷瑄

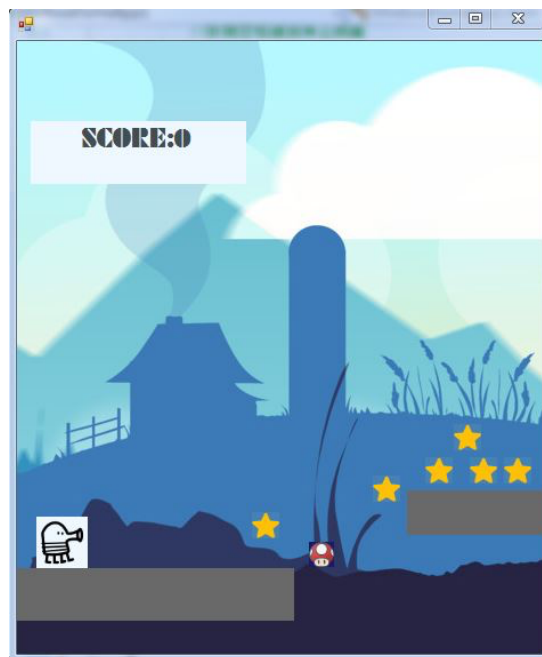
### 一、動機

因為我們這組在討論期末專題時，決定要製作一個類 **doodle jump** 的遊戲，遊戲介面有各種平台浮在空中，玩家需要控制角色的左右方位，因此這次的小作業我便嘗試做一個平台遊戲，之後再逐步發展成最終的期末作業。

### 二、遊戲規則解說

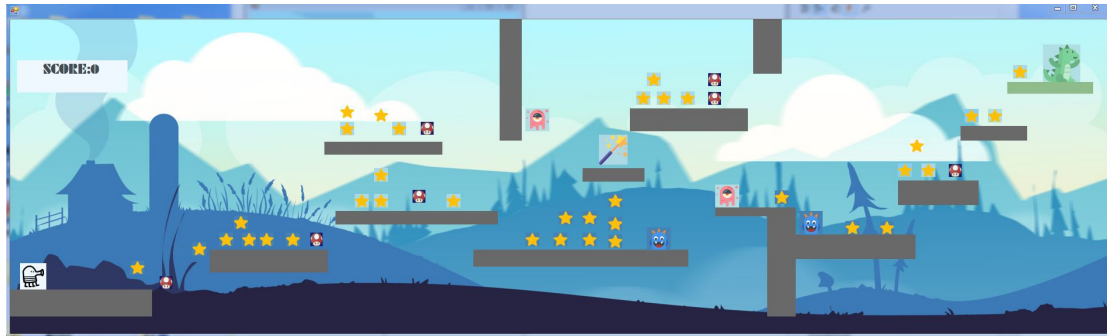
在遊戲中，平台大部分都是浮在空中，玩家需要小心潛伏在四周的怪物，並以左右鍵來操作角色移動，按空白鍵是往上跳躍，來避免角色掉入深淵，使遊戲結束，遊戲畫面會由左至右方移動，玩家需收集魔法師弄丟的魔法棒，且獲得一定數量的星星，才能解救變成恐龍的魔法師。

### 三、遊戲畫面與程式碼



遊戲起始畫面

遊戲中的靜態元素：平台、怪物、星星（分數加一）、蘑菇（分數減一）、分  
遊戲中的動態元素：角色的移動、跳躍與奔跑、背景移動



遊戲全畫面

## 一、角色與背景的移動（使遊戲畫面會由左至右方移動）

（一）設定角色的移動、跳躍與奔跑速度之數值，以及重力數值

```
int scoreNumber = 0;
int force = 8;
int playerJumpSpeed = 10;
int backgroundSpeed = 8;
int playerSpeed = 10;

bool left, right, jump, takemagic;
```

（二）設定角色的移動（避免玩家跑出視窗之外）

```
if (left == true && player.Left > 60)
{
    player.Left -= playerSpeed;
}
if (right == true && player.Left + (player.Width + 60) < this.ClientSize.Width)
{
    player.Left += playerSpeed;
}
```

（三）設定背景位置，使背景和角色跑步速度一樣

```
if (left == true && background.Left < -3)
{
    background.Left += backgroundSpeed;
    GameElements("GOforward");
}
if (right == true && background.Left > -1341)
{
    background.Left -= backgroundSpeed;
    GameElements("Goback");
}
```

#### (四) 使玩家移動時能達到其他物件靜止的效果

```
private void GameElements(string direction)
{
    //*****
    foreach (Control x in this.Controls)
    {
        if (x is PictureBox && (string)x.Tag == "platform" || x is PictureBox && (string)x.Tag == "star" || (string)x.Name == "key" ||
            (string)x.Name == "Dinosaur" || x is PictureBox && (string)x.Tag == "mushroom" || x is PictureBox && (string)x.Tag == "Monster")
        {
            if (direction == "Goback")
            {
                x.Left -= backgroundSpeed;
            }
            if (direction == "Goforward")
            {
                x.Left += backgroundSpeed;
            }
        }
    }
    //設定其餘遊戲元素跟背景同樣速度
    //*****
}
```

## 二、角色跳躍與左右移動

#### (一) 設定一個 force 的值，使角色跳躍時不會跳出視窗之外

```
if (jump == true && force < 0)
{
    jump = false;
}
if (jump == true)
{
    playerJumpSpeed = -12;
    force -= 1;
}
else
{
    playerJumpSpeed = 12;
}
```

#### (二) Key Down 事件

```
private void Key_Down(object sender, KeyEventArgs e)
{
    //*****

    if (e.KeyCode == Keys.Right)
    {
        right = true;
    }
    if (e.KeyCode == Keys.Left)
    {
        left = true;
    }
    //當Player往上跳時，再次點擊空格鍵也不影響Player

    if (e.KeyCode == Keys.Space && jump == false)
    {
        jump = true;
    }

    //*****
}
```

### (三) Key Up 事件

```
private void Key_Up(object sender, KeyEventArgs e)
{
    //*****
    if (e.KeyCode == Keys.Right)
    {
        right = false;
    }
    if (e.KeyCode == Keys.Left)
    {
        left = false;
    }
    //放開空格鍵就停止跳躍
    if (jump == true)
    {
        jump = false;
    }
    //*****
}
```

### 三、角色碰到各項物件邊緣

```
foreach (Control x in this.Controls)
{
    if (x is PictureBox && (string)x.Tag == "platform")
    {
        if (player.Bounds.Intersects(x.Bounds) && jump == false)
        {
            {
                force = 8;
                player.Top = x.Top - player.Height;
                playerJumpSpeed = 0;
            }
        }
    }
    if (x is PictureBox && (string)x.Tag == "mushroom")
    {
        if (player.Bounds.Intersects(x.Bounds) && x.Visible == true)
        {
            x.Visible = false;
            scoreNumber -= 1;
        }

        if (player.Bounds.Intersects(key.Bounds))
        {
            key.Visible = false;
            takemagic = true;
        }
    }
    if (x is PictureBox && (string)x.Tag == "star")
    {
        if (player.Bounds.Intersects(x.Bounds) && x.Visible == true)
        {
            x.Visible = false;
            scoreNumber += 1;
        }
    }
}
```

#### 四、當遊戲結束或是遊戲獲勝時，會跳出視窗（以碰到怪物為例）

按下是：再重新開啟新視窗

按下否：視窗關閉

```
if (x is PictureBox && (string)x.Tag == "Monster")
    if (player.Bounds.Intersects(x.Bounds) && x.Visible == true)
    {
        score.Text = "SCORE:" + scoreNumber + Environment.NewLine + "GAME OVER !";
        Timer.Stop();
        DialogResult Result = MessageBox.Show("碰到怪獸! 遊戲結束" + Environment.NewLine + "是否要重新開始遊戲?", "GAME OVER", MessageBoxButtons.YesNo, MessageBoxIcon.Warning);
        if (Result == DialogResult.Yes)
        {
            RestartGame();
            return;
        }
        else if (Result == DialogResult.No)
        {
            this.Close();
        }
    }
}
```

#### 五、重新開一個新的遊戲視窗

```
private void RestartGame()
{
    form Window = new form(); //創建子視窗
    Window.Show(); //顯示子視窗
    this.Hide(); //隱藏父視窗
}
```

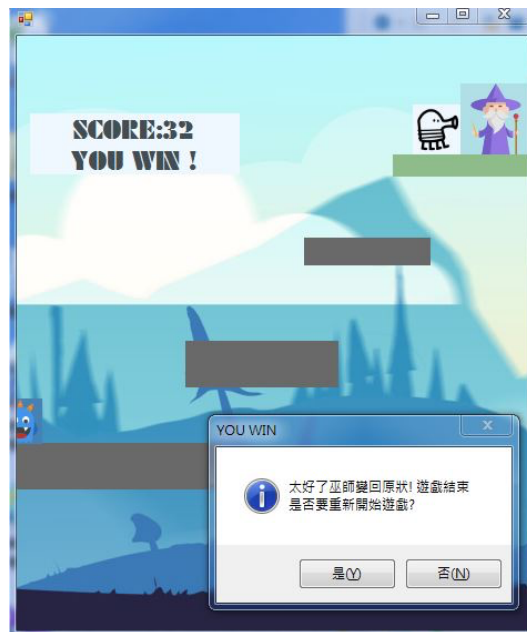
#### 四、遊戲執行畫面



碰到怪獸的遊戲畫面



角色掉落的遊戲畫面



玩家獲勝的遊戲畫面

#### 四、遇到的困難與反思

問題 1:目前玩家可以往後走幾步，但無法返回原地。(若往左走的話，遊戲中其他物件會跟著移動)

問題 2:玩家在移動時會有角色抖動、延遲的問題。

之後發展方向：

1. 怪獸和平台能夠上下左右移動。
2. 讓玩家可以發射子彈，攻擊怪獸。
3. 讓玩家隨著背景一起往前，做出類似 doodle jump，一直往上或往右移動的效果。

## 五、參考文獻

[WinForm] 多重視窗的操作

<https://yuchungchuang.wordpress.com/2018/06/09/winform-%E5%A4%9A%E9%87%8D%E8%A6%96%E7%AA%97%E7%9A%84%E6%93%8D%E4%BD%9C-multiple-forms/> 本報告完成時最後瀏覽日期:2021/6/9

鄭士康(2013)《以 C#學程式設計基本:使用 Visual Studio 2012 Express for Windows Desktop》。台北，未出版。

素材來源: <https://www.flaticon.com/>

【程式設計-C#】馬力歐跳跳跳

<https://fgchen.com/wp/%E3%80%90%E7%A8%8B%E5%BC%8F%E8%A8%AD%E8%A8%88%E3%80%91%E3%80%90c%E3%80%91%E3%80%90%E9%81%8A%E6%88%B2%E8%A8%AD%E8%A8%88%E3%80%91%E8%A7%92%E8%89%B2jump/> 本報告完成時最後瀏覽日期:2021/6/9