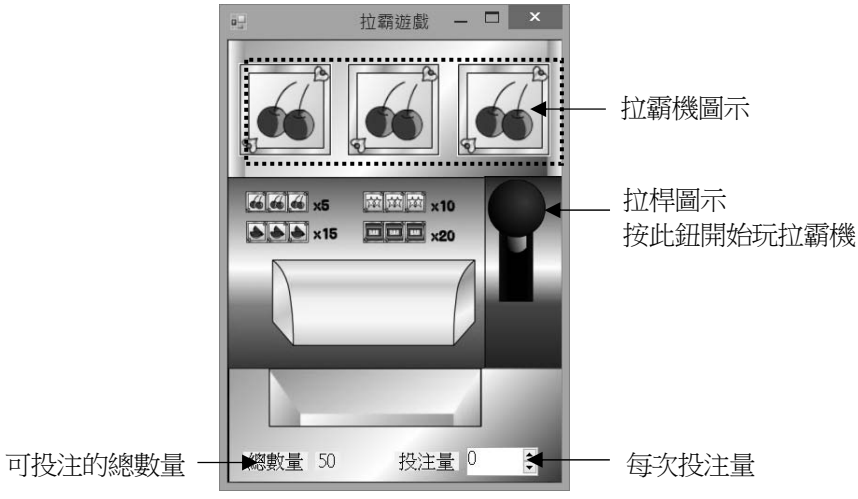
Chapter 15

遊戲與資料庫專題實作

- 拉霸遊戲機實作
- 記憶大考驗實作
- 進銷存管理系統實作
- 專題實作報告格式

15.1 拉霸遊戲機實作

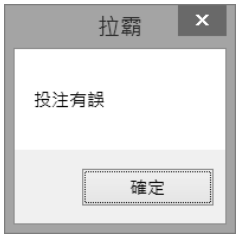
前面章節都只針對每個 Visual C# 或控制項做個別的介紹，本章將整合前面章節所學，應用到日常的生活案例上，介紹如何實際製作一個專題，由系統功能、系統設計、演算法以及程式碼編寫四個部份，來培養初學者具有專題製作能力。本章將介紹遊戲和資料庫專題兩大部分：(若因上課時數不夠或無專題製作要求之課程，本章可略過僅供學生日後需要時參考)拉霸遊戲機是仿間電玩機常見的機台，常見的有九個圖示的拉霸機與三個圖示的拉霸機，九個圖示的拉霸機在得獎的機率與設計的過程比較複雜，本節介紹三個圖示的拉霸機。如下圖：



一、系統功能說明

下列是拉霸遊戲機的遊戲規則說明：

1. 開始時必須先設定本次投注的數量，接著按下 拉桿圖示即開始玩拉霸機。若投注量為 0 或投注量超過你擁有的總數量，即顯示對話方塊，告知您「投注有誤」無法玩拉霸機。投注總量預設 50。



2. 若允許投注時，按 未拉桿圖示即變成 已按拉桿圖示，此時拉霸機開始轉動且表單上的三個圖片方塊控制項中，每個控制項會以亂數的方式由下面 0.jpg~3.jpg 四張圖檔中任選一張來顯示。

0.jpg 1.jpg 2.jpg - 3.jpg

大約過了三秒後，拉霸機停止自動轉動，並將拉桿恢復成 圖示，最後再判斷是否有中獎？中獎條件如下：

① 若得到 3 個 圖，則投注量得到 2 倍。

② 若得到 3 個 圖，則投注量得到 10 倍。

③ 若得到 3 個 圖，則投注量得到 15 倍。

④ 若得到 3 個 圖，則投注量得到 20 倍。

二、輸出入介面設計

1. 新增名稱為「拉霸」的 Windows Form 應用程式專案。

2. 建立下面控制項以及設定相關屬性：

① 設定表單的背景圖為書附光碟的「ch15/拉霸素材檔/拉霸背景.jpg」，用來當做拉霸機的底圖。

② 建立名稱為 pic1、pic2、pic3 的圖片方塊控制項，用來當做拉霸機遊戲的三個圖示。

③ 建立名稱為 picBtn 圖片方塊控制項，用來當做拉霸機的拉桿。

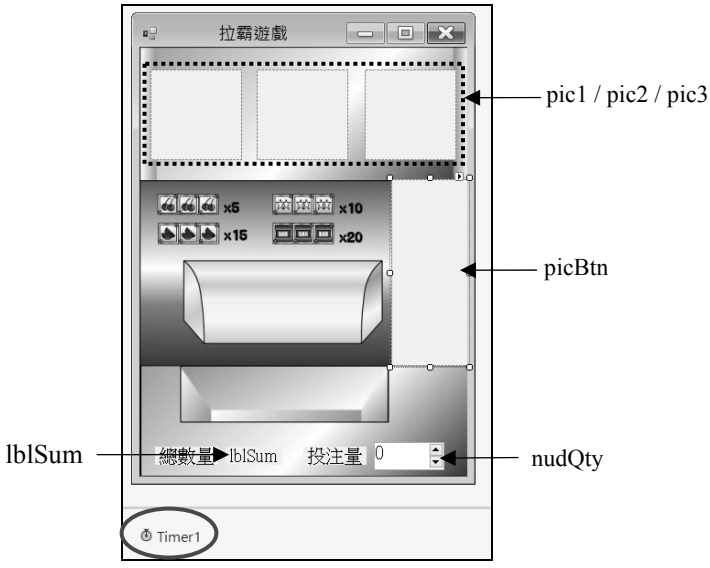
④ 建立名稱為 lblSum 標籤控制項，用來顯示可投注的總數量。

⑤ 建立名稱為 nudQty 數字鈕控制項，用來設定每一次拉霸要投注的數量。

⑥ 建立名稱為 timer1 計時器控制項，使每 0.1 秒讓 pic1、pic2、pic3 以亂數方式顯示 0.jpg~3.jpg。

⑦ 將 0.jpg~3.jpg、up.jpg、down.jpg 圖檔放到目前專案的 bin/Debug 資料夾下。

本例必須在表單建立下列各控制項：



三、系統分析

1. 本遊戲需先設計一張背景圖當做表單的背景，接著在表單上面建立三個圖片方塊，一個未按和已按拉桿圖示、一個標籤控制項用來顯示目前可投注總數量和一個數字按鈕控制項用來設定每次投注量。

2. 拉霸機上面三個圖示如何建立
   為方便使用迴圈來讀取圖片，必須將表單上的三個圖片方塊設成控制項陣列，以及將圖檔變成檔名尾端含有連續數字檔案。本專題建立一個陣列為 PictureBox 圖片方塊控制項陣列，陣列元素為 p[0]~p[3]，其中 p[0]不使用，將 pic1~pic3 三個圖片方塊控制項指定給 p[1]~p[3]，此時即可使用迴圈，透過 p[1]~p[3] 來操作 pic1~pic3 三個圖片方塊控制項的相關屬性。
3. 如何應用計時器由圖片方塊控制項中亂數取圖
   當拉霸機啟動時，timer1 計時器的 Tick 事件即會開始計時，在指定時間內，pic1~pic3 三張圖片方塊控制項分別以亂數方式由 0.jpg~3.jpg 四張圖片中選取一張來顯示。為了讓拉霸機上面的三張水果圖有滾動

的感覺，使其每隔 0.1 秒重新亂數取圖一次，連續 20 次才停止計時，因此必須將 t 整數變數宣告在所有事件處理函式外面以方便讓所有事件一起共用，當計時器執行一次時 t 變數累加 1，當 t 等於 20，計時器即停止且 t 變數馬上還原為 0。

4. 如何判斷是否中獎？

將四張圖分別設定代碼：

① 荔枝代碼為 0　② 星星代碼為 1

③ 西瓜代碼為 2　④ BAR 代碼為 3

num[1]=1

num[2]=2

num[3]=0

譬如：上圖為拉霸機亂數出來的水果圖，將三個水果圖代碼依序存入指定陣列，陣列名稱設為 num。若 num[1]、num[2] 和 num[3] 的代碼都相同表示有中獎，依中獎規則依指定倍數賠，若未中獎必須由目前可投注量扣除本次投注量。

▶ 解題技巧

1. 使用 lblSum 標籤控制項來顯示可投注的總數量、使用 rndQty 數字鈕控制項來設定每次投注量、使用 picBtn 圖片方塊當拉桿圖示鈕。
2. 設定 p[1]~p[3]為圖片方塊控制項陣列、num[1]~num[3] 整數陣列存放對應圖片代碼，因 t 為整數變數要多次使用，必須保留其值。所以上述變數必須宣告在所有事件處理函式之外。
3. 在表單設計階段，設定表單的背景圖為書附光碟「ch15/拉霸素材檔/拉霸背景.jpg」，該圖檔用來當做拉霸機的底圖。
4. 當表單載入時，必須在 Form1_Load 事件處理函式做下列事情：
   ① 透過表單的 AutoSizeMode 屬性使表單設為無法被調整大小，如此才不會破壞拉霸遊戲的背景圖。
   ② 使 picBtn 圖片方塊控制項顯示 拉桿圖示(up.jpg)。

③ 將 pic1 指定給 p[1]、pic2 指定給 p[2]、pic3 指定給 p[3]，此時指定 p[1] 的屬性就是指定 pic1 一樣，此舉的好處是方便使用迴圈來設定 p[1]~p[3] 圖片方塊控制項陣列的相關屬性。

④ 使用迴圈使 p[1]~p[3] 顯示 0.jpg 荔枝圖，即是使 pic1~pic3 顯示 0.jpg 荔枝圖。

⑤ 使 timer1 計時器每 0.1 秒執行一次。

⑥ 可投注的總數量 lblSum 為 50。

5. 當按下 picBtn 拉桿圖示時會執行 picBtn_Click 事件處理函式，請在此事件做下列事情。

① 判斷投注量是否大於 0 或投注量是否小於總數量？若成立繼續下一步驟，否則跳至步驟③。

② 先啟動 timer1 計時器，減掉本次投注量，使 rndQty 數字鈕控制項無法使用、btnPic 圖示鈕失效、btnPic 以 圖示(down.jpg)顯示。

③ 出現對話方塊並顯示「投注有誤」的訊息。

6. 當 timer1 計時器啟動時會每 0.1 秒執行一次 timer1_Tick 事件處理函式，請在該事件處理函式做下列事情：

① 使用 Random 類別建立 r 亂數物件：

```
Random r = new Random();          // 建立亂數物件 r
```

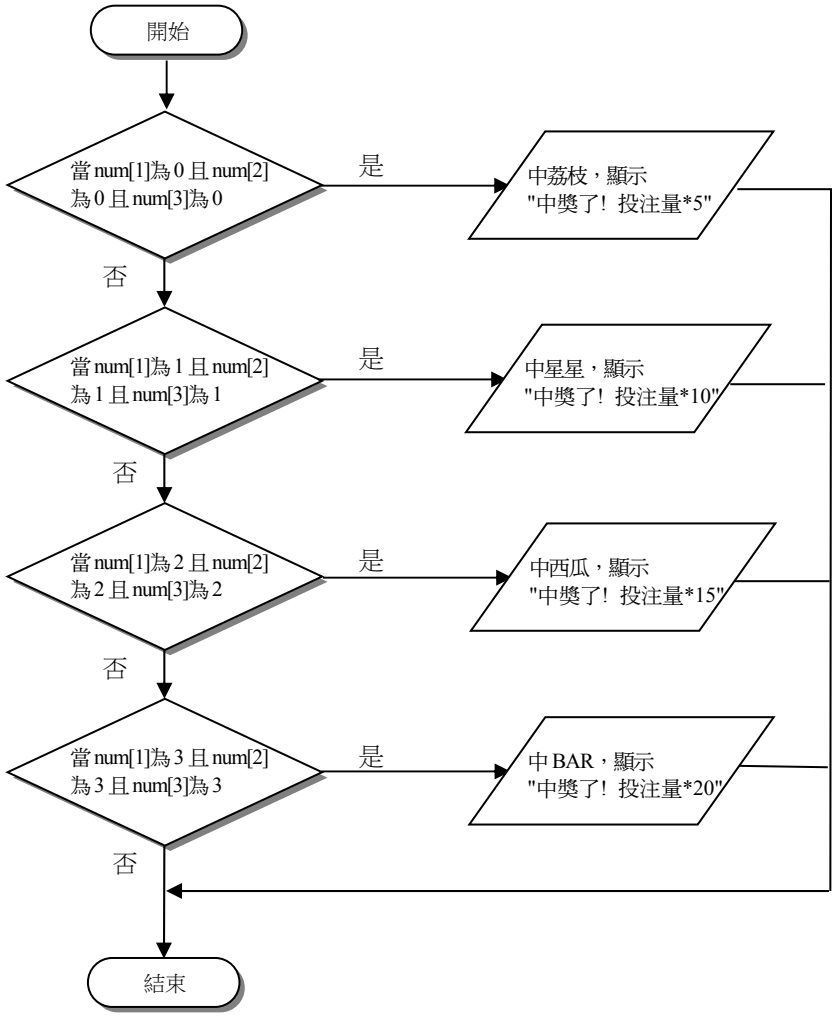
② 使用迴圈並配合 r.Next(0,4) 方法使 pic1~pic3 皆以亂數顯示 0.jpg~3.jpg。寫法：

```
for (int i = 1; i <= p.GetUpperBound(0); i++)
{
        num[i] = r.Next(0, 4);    // 產生 0~3 的亂數並指定給
num[1]~num[3]  // 使 pic1~pic3 以亂數的方式顯示 0.jpg~3.jpg
        p[i].Image = new Bitmap(Convert.ToString(num[i])
+ ".jpg"); }
```

③ 使 t 加 1，用來表示 timer1 計時器控制項的 timer1_Tick 事件處理函式執行的次數。

④ 當 t 等於 20，即 timer1 計時器控制項的 timer1_Tick 事件處理函式執行了 20 次，此時即做下列事情：

- timer1 計時器停止、picBtn 圖片方塊啟用、rndQty 數字鈕控制項啟用可進行投注。
- btnPic 以 圖示顯示，另 t 等於 0。
- 判斷是否中獎，流程圖如下：



四、完整程式碼

FileName: 拉霸.sln

```
01 namespace 拉霸
02 {
03     public partial class Form1 : Form
04     {
05         public Form1()
06         {
07             InitializeComponent();
08         }
09
10         // 建立p[0]~p[3]的PictureBox陣列元素，可用來存放0.jpg~3.jpg
11         // 其中p[0]不使用，p[1]~p[3]代表三個拉霸圖示
12         PictureBox[] p = new PictureBox[4];
13         // 建立num[0]~num[3]的整數陣列，用來存放0~3的數值
14         // 其中num[0]不使用
15         int[] num = new int[4];
16         int t; // 宣告t用來計算timer1_Tick事件處理函式共執行幾次
17
18         //表單載入時執行Form1_Load事件處理函式
19         private void Form1_Load(object sender, EventArgs e)
20         {
21             // 使表單無法被調整大小
22             this.AutoSizeMode = AutoSizeMode.GrowAndShrink;
23             picBtn.Image = new Bitmap("up.jpg"); //使picBtn顯示up.jpg
24             // 使載入的圖片隨picBtn大小伸縮
25             picBtn.SizeMode = PictureBoxSizeMode.StretchImage;
26             p[1] = pic1;     // 將pic1指定給p[1]
27             p[2] = pic2;     // 將pic2指定給p[2]
28             p[3] = pic3;     // 將pic3指定給p[3]
29             // 使用迴圈使pic1~pic3顯示0.jpg圖
30             for (int i = 1; i <= p.GetUpperBound(0); i++)
31             {
32                 p[i].Image = new Bitmap("0.jpg");
33                 p[i].SizeMode = PictureBoxSizeMode.Zoom;
34             }
```

```
35          timer1.Interval = 100;   // 使timer1計時器的Tick事件每0.1秒執行一次
36          lblSum.Text = "50";     // 可投注的總數量lblSum為50
37      }
38
39      // 按下picBtn鈕時會執行picBtn_Click事件處理函式
40      private void picBtn_Click(object sender, EventArgs e)
41      {
42          // 判斷 投注量nudQty.Value是否大於 0
43          // 且 投注量nudQty.Value是否大於可投注總數量lblSum.Text
44          if (nudQty.Value > 0 &&
                nudQty.Value <= Convert.ToInt32(lblSum.Text))
45          {
46              timer1.Enabled = true;       //計時器timer1啟動
47              //可投注量減掉本次的投注量
48              lblSum.Text = Convert.ToString
                    ((Convert.ToInt32(lblSum.Text) - nudQty.Value));
49              nudQty.Enabled = false;   // 無法投注
50              picBtn.Image = new Bitmap("down.jpg"); //使picBtn顯示down.jpg
51              picBtn.Enabled = false;   // picBtn圖片按鈕失效
52          }
53          else
54          {   // 若投注量小於0且投注量大於可投注的總數量
55              MessageBox.Show("投注有誤");
56          }
57      }
58
59      // 當timer1啟動時，每0.1秒皆會執行timer1_Tick事件處理函式一次
60      private void timer1_Tick(object sender, EventArgs e)
61      {
62          Random r = new Random();   //建立亂數物件r
63
64          //使用迴圈讓pic1~pic3每次執行皆亂數的方式顯示0.jpg~3.jpg
65          for (int i = 1; i <= p.GetUpperBound(0); i++)
66          {
67              num[i] = r.Next(0, 4); //產生 0~3 的亂數並指定給 num[1]~num[3]
68              // 使pic1~pic3以亂數的方式顯示 0.jpg~3.jpg
```

```
69                  p[i].Image = new Bitmap(Convert.ToString(num[i]) + ".jpg");
70              }
71              t += 1;
72              //  當計時器執行20次時，即馬上判斷是否中獎
73              if (t == 20)
74              {
75                  timer1.Enabled = false;     //計時器timer1停止
76                  nudQty.Enabled = true;      //可以開始投注
77                  picBtn.Enabled = true;       // picBtn圖形按鈕可啟用
78              //當num[1]為0且num[2]為0且num[3]為0表示pic1~pic3三個圖示皆是荔枝
79                  if (num[1] == 0 && num[2] == 0 && num[3] == 0)
80                  {
81                      lblSum.Text = (Convert.ToInt32
                            (lblSum.Text) + (nudQty.Value * 5)).ToString();
82                      MessageBox.Show("中獎了!  投注量*5");
83                  }
84              //當num[1]為1且num[2]為1且num[3]為1表示pic1~pic3三個圖示皆是星星
85                  else if (num[1] == 1 && num[2] == 1 & num[3] == 1)
86                  {
87                      lblSum.Text = (Convert.ToInt32(lblSum.Text) +
                            (nudQty.Value * 10)).ToString();
88                      MessageBox.Show("中獎了!  投注量*10");
89                  }
90              //當num[1]為2且num[2]為2且num[3]為2表示pic1~pic3三個圖示皆是西瓜
91                  else if (num[1] == 2 && num[2] == 2 && num[3] == 2)
92                  {
93                      lblSum.Text = (Convert.ToInt32(lblSum.Text) +
                            (nudQty.Value * 15)).ToString();
94                      MessageBox.Show("中獎了!  投注量*15");
95                  }
96              //當num[1]為3且num[2]為3且num[3]為3表示pic1~pic3三個圖示皆是BAR
97                  else if (num[1] == 3 && num[2] == 3 && num[3] == 3)
98                  {
99                      lblSum.Text = (Convert.ToInt32(lblSum.Text) +
                              (nudQty.Value * 20)).ToString();
100                     MessageBox.Show("中獎了!  投注量*20");
101                 }
```