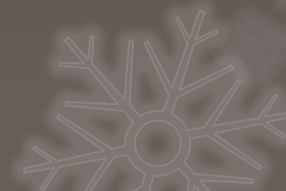


軟體開發

撲克牌專案

學生：03356013 黃啟倫
指導教授：曾修宜 教授



主要流程

洗牌

→ 隨機產生52張牌








→ 從中隨機挑出5張牌

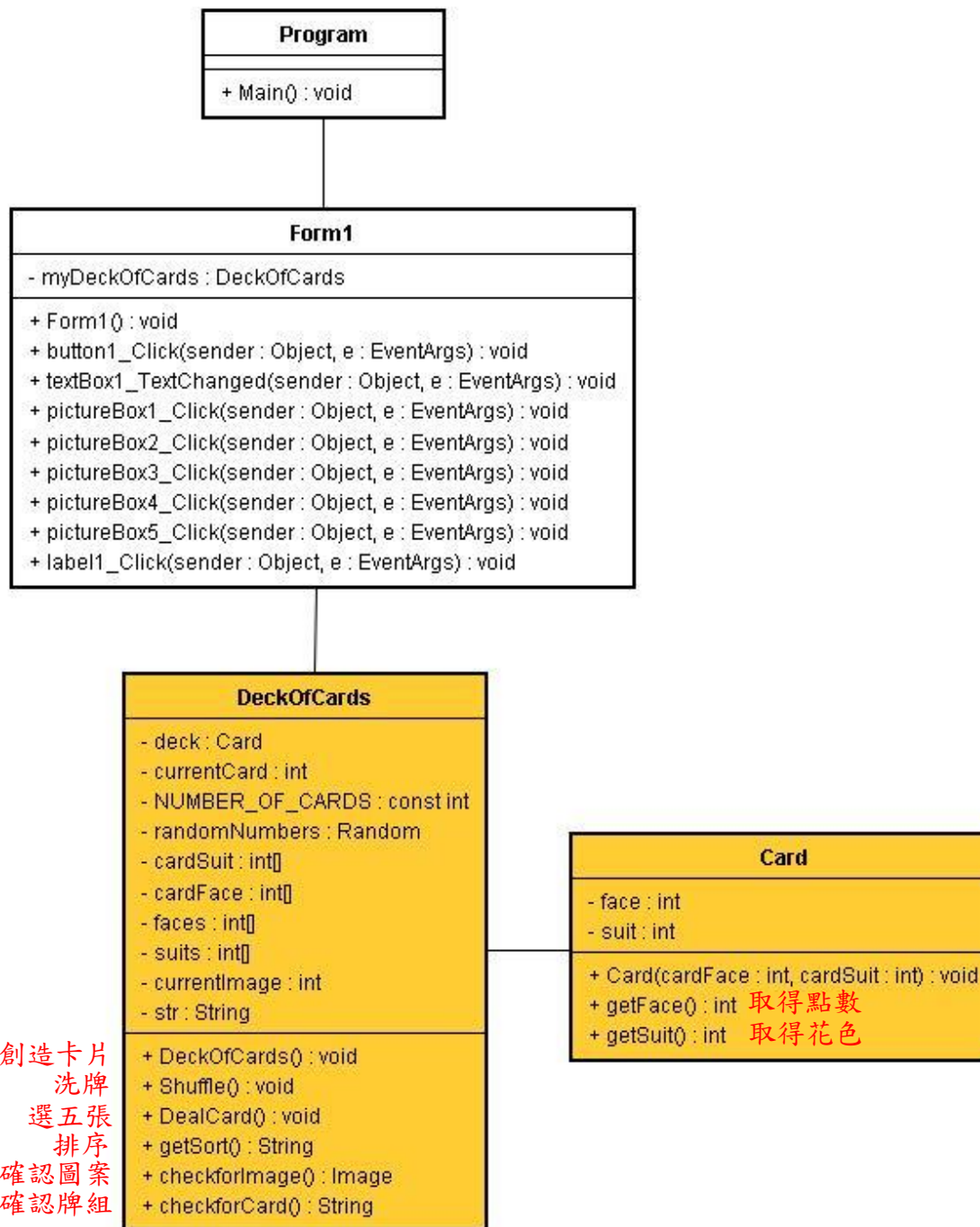
→ 將牌組排序A K Q J 10....大到小

→ 確認牌組的圖案

→ 判斷牌組的組合為?

Poker Hand Rankings

Royal Flush					
Straight Flush					
4-of-a-Kind					
Full House					
Flush					
Straight					
3-of-a-Kind					
Two Pair					
One Pair					
High Card					



創造卡片
洗牌
選五張
排序
確認圖案
確認牌組

```

public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
        textBox1.Visible = false;
    }
    DeckOfCards myDeckOfCards = new DeckOfCards();
    public void button1_Click(object sender, EventArgs e)
    {
        myDeckOfCards.Shuffle(); // place Cards in random order

        // display all 52 Cards in the order in which they are dealt
        for (int i = 0; i < 5; ++i)
        {
            //選擇卡片5張
            myDeckOfCards.DealCard();

        } // end for
        //排序
        myDeckOfCards.getSort();

        //textBox1.Visible = true; //需要結果可解開

        for (int i = 0; i < 5; i++)
        {
            switch (i)
            {
                case 0:
                    pictureBox1.Image = myDeckOfCards.checkforImage();
                    break;
                case 1:
                    pictureBox2.Image = myDeckOfCards.checkforImage();
                    break;
                case 2:
                    pictureBox3.Image = myDeckOfCards.checkforImage();
                    break;
                case 3:
                    pictureBox4.Image = myDeckOfCards.checkforImage();
                    break;
                default:
                    pictureBox5.Image = myDeckOfCards.checkforImage();
                    break;
            }
        }
    }
}

```

```

public class DeckOfCards
{
    private Card[] deck; // array of Card objects
    private int currentCard; // index of next Card to be dealt (0-51)
    private const int NUMBER_OF_CARDS = 52; // constant number of Cards
    private Random randomNumbers; // random number generator
    private int[] cardSuit = new int[5];
    private int[] cardFace = new int[5];
    private int[] faces = { 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1 };
    private int[] suits = { 0, 1, 2, 3 };
    private int currentImage;
    String str;
}

```

用數字代表牌組

```

public void Shuffle()
{
    // after shuffling, dealing should start at deck[ 0 ] again
    currentCard = -1; // reinitialize currentCard
    currentImage = 0;
    // for each Card, pick another random Card and swap them
    for (int first = 0; first < deck.Length; ++first)
    {
        // select a random number between 0 and 51
        int second = randomNumbers.Next(NUMBER_OF_CARDS);

        // swap current Card with randomly selected Card
        Card temp = deck[first];
        deck[first] = deck[second];
        deck[second] = temp;
    } // end for
} // end method Shuffle

```

創造52張牌

```

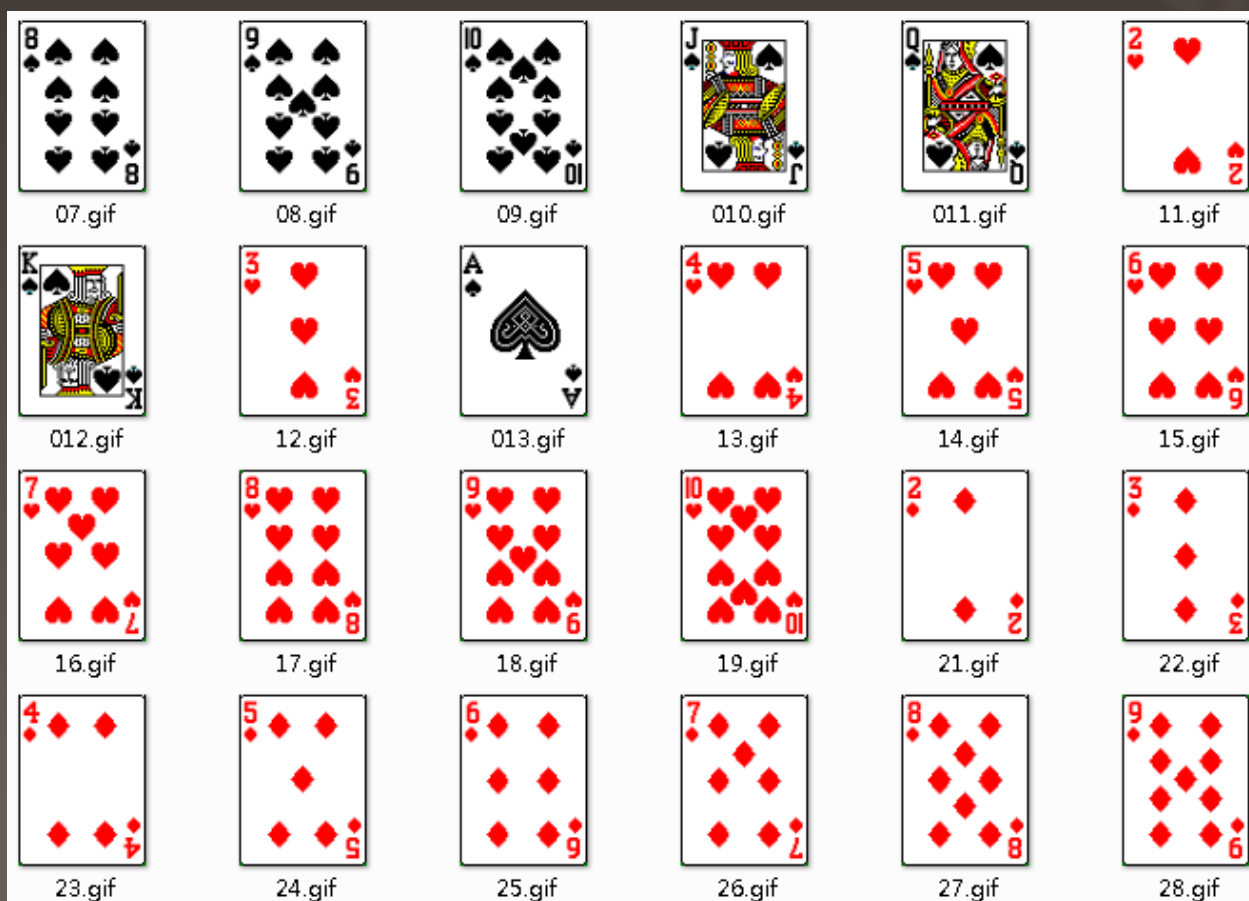
public void DealCard()
{
    // determine whether Cards remain to be dealt
    if (currentCard < 5)
    {
        currentCard++;

        cardSuit[currentCard] = deck[currentCard].getSuit();
        cardFace[currentCard] = deck[currentCard].getFace();
        // return cardFace[currentCard] + " of " + cardSuit[currentCard]
    }
} // end method DealCard

```

挑出5張牌組成牌組

數字	A	K	Q	J	10	9	8	7	6	5	4	3	2
桃	013	012	011	010	09	08	07	06	05	04	03	02	01
紅	113	112	111	110	19	18	17	16	15	14	13	12	11
菱	213	212	211	210	29	28	27	26	25	24	23	22	21
梅	313	312	311	310	39	38	37	36	35	34	33	32	31



```

public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
        textBox1.Visible = false;
    }
    DeckOfCards myDeckOfCards = new DeckOfCards();
    public void button1_Click(object sender, EventArgs e)
    {

        myDeckOfCards.Shuffle(); // place Cards in random order

        // display all 52 Cards in the order in which they are dealt
        for (int i = 0; i < 5; ++i)
        {
            //選擇卡片5張
            myDeckOfCards.DealCard();

        } // end for
        //排序
        myDeckOfCards.getSort();

        //textBox1.Visible = true; //需要結果可解開

        for (int i = 0; i < 5; i++)
        {
            switch (i)
            {
                case 0:
                    pictureBox1.Image = myDeckOfCards.checkforImage();
                    break;
                case 1:
                    pictureBox2.Image = myDeckOfCards.checkforImage();
                    break;
                case 2:
                    pictureBox3.Image = myDeckOfCards.checkforImage();
                    break;
                case 3:
                    pictureBox4.Image = myDeckOfCards.checkforImage();
                    break;
                default:
                    pictureBox5.Image = myDeckOfCards.checkforImage();
                    break;
            }
        }
    }
}

```

```

public string getSort()
{
    //排序
    for (int allcheck = 0; allcheck < cardSuit.Length; allcheck++)
    {
        for (int sort = 0; sort < cardSuit.Length - 1; sort++)
        {
            if (cardFace[sort] < cardFace[sort + 1])
            {
                int temp = 0;
                int temp1 = 0;

                temp = cardFace[sort + 1];
                cardFace[sort + 1] = cardFace[sort];
                cardFace[sort] = temp;

                temp1 = cardSuit[sort + 1];
                cardSuit[sort + 1] = cardSuit[sort];
                cardSuit[sort] = temp1;
            }
        }
    }

    String sttr = "";
    for (int o = 0; o < cardSuit.Length; o++)
    {
        sttr += cardFace[o] + " of " + cardSuit[o] + "\r\n";
    }
    return sttr;
}

```

排序以便後續判斷


```

for (int i = 0; i < 5; i++)
{
    switch (i)
    {
        case 0:
            pictureBox1.Image = myDeckOfCards.checkforImage();
            break;
        case 1:
            pictureBox2.Image = myDeckOfCards.checkforImage();
            break;
        case 2:
            pictureBox3.Image = myDeckOfCards.checkforImage();
            break;
        case 3:
            pictureBox4.Image = myDeckOfCards.checkforImage();
            break;
        default:
            pictureBox5.Image = myDeckOfCards.checkforImage();
            break;
    }
}

```

判斷每一張牌的花色數字，組成檔名
讀取圖片，回傳Image物件

```

public Image checkforImage()
{
    currentImage++;

    if (currentImage == 1)
    {
        str = cardSuit[currentImage - 1].ToString() + cardFace[currentImage - 1].ToString();
        return Image.FromFile(str + ".gif");
    }
    else if (currentImage == 2)
    {
        str = cardSuit[currentImage - 1].ToString() + cardFace[currentImage - 1].ToString();
        return Image.FromFile(str + ".gif");
    }
    else if (currentImage == 3)
    {
        str = cardSuit[currentImage - 1].ToString() + cardFace[currentImage - 1].ToString();
        return Image.FromFile(str + ".gif");
    }
}

```

```
textBox1.Text += myDeckOfCards.checkforCard() + "\r\n";  
label1.Text = myDeckOfCards.checkforCard();
```



進行判斷，有優先順序

```
public string checkforCard()  
{
```

```
    String answer = "";
```

```
    if ((cardSuit[0] == cardSuit[1] && cardSuit[1] == cardSuit[2] && cardSuit[2] == cardSuit[3] && cardSuit[3] == cardSuit[4]) && (cardFace[0] == 13 && cardFace[1] == 12 && cardFace[2] == 11 && cardFace[3] == 10 && cardFace[4] == 9))
```

```
    {  
        answer = "Royal Straight";
```

```
    }
```

```
    else
```

```
    {
```

```
        if ((cardSuit[0] == cardSuit[1] && cardSuit[1] == cardSuit[2] && cardSuit[2] == cardSuit[3] && cardSuit[3] == cardSuit[4]) && ((cardFace[0] - cardFace[1]) == 1 && (cardFace[1] - cardFace[2]) == 1 && (cardFace[2] - cardFace[3]) == 1 && (cardFace[3] - cardFace[4]) == 1))
```

```
        {  
            answer = "Straight Flush";
```

```
        }
```

```
    else
```

```
    {
```

```
        if ((cardFace[0] == cardFace[1] && cardFace[1] == cardFace[2] && cardFace[2] == cardFace[3]) | (cardFace[4] == cardFace[3] && cardFace[3] == cardFace[2] && cardFace[2] == cardFace[1]))
```

```
        {  
            answer = "Four of a kind";
```

```
        }
```

```
    else
```



Poker Hand Rankings

Royal Flush 

Straight Flush 

4-of-a-Kind 

Full House 

Flush 

Straight 

3-of-a-Kind 

Two Pair 

One Pair 

High Card 

