

資訊安全-隱藏學作業

2 個參考

class Codeing

{

1 個參考

public void

1 個參考

public void

0 個參考

static void M

```
public void Uncode()
{
    string content = File.ReadAllText(@"C:\Users\qoo09\OneDrive\program\C#\RGBT
    Bitmap image = new Bitmap(@"C:\Users\qoo09\OneDrive\program\C#\RGBTtransfo

    // 使用 LockBits 來增加效率
    BitmapData bmpData = image.LockBits(new Rectangle(0, 0, image.Width, image.
    IntPtr ptr = bmpData.Scan0;
    int bytes = Math.Abs(bmpData.Stride) * image.Height;
    byte[] rgbValues = new byte[bytes];
    Marshal.Copy(ptr, rgbValues, 0, bytes);
```

```
string sum = "";
int count = 0;
int x = 0, y = 0;
```

```
for (int i = 1; i
```

```
public void Encode()
{
    Bitmap image = new Bitmap(@"C
    Random rand = new Random();
    StringBuilder storage = new StringBuilder();
```

```
BitmapData bmpData = image.LockBits(new Rectangle(0, 0,
IntPtr ptr = bmpData.Scan0;
int bytes = Math.Abs(bmpData.Stride) * image.Height;
byte[] rgbValues = new byte[bytes];
Marshal.Copy(ptr, rgbValues, 0, bytes);
```

```
for (int y = 0; y < image.Height; y++)
```

```
{
    for (int x = 0; x < image.Width; x++)
```

```
{
    int position = (y * bmpData.Stride) + (x * 3);
    byte blue = rgbValues[position];
    byte green = rgbValues[position + 1];
    byte red = rgbValues[position + 2];
```

```
if (red <= 240 || green <= 240 || blue <= 240)
```

```
// 计算当前像素的位置
```

```
int position = (y * bmpData.Stride) + (x * 3)
rgbValues[position] = 0; // Blue
rgbValues[position + 1] = 0; // Green
rgbValues[position + 2] = 0; // Red
```

```
sum = "";
```

```
// 把 rgbValues 的值写回到图像
Marshal.Copy(rgbValues, 0, ptr, bytes);
image.UnlockBits(bmpData);
```

```
image.Save(@"C:\Users\qoo09\OneDrive\program\C#\RGBTtransfo
```

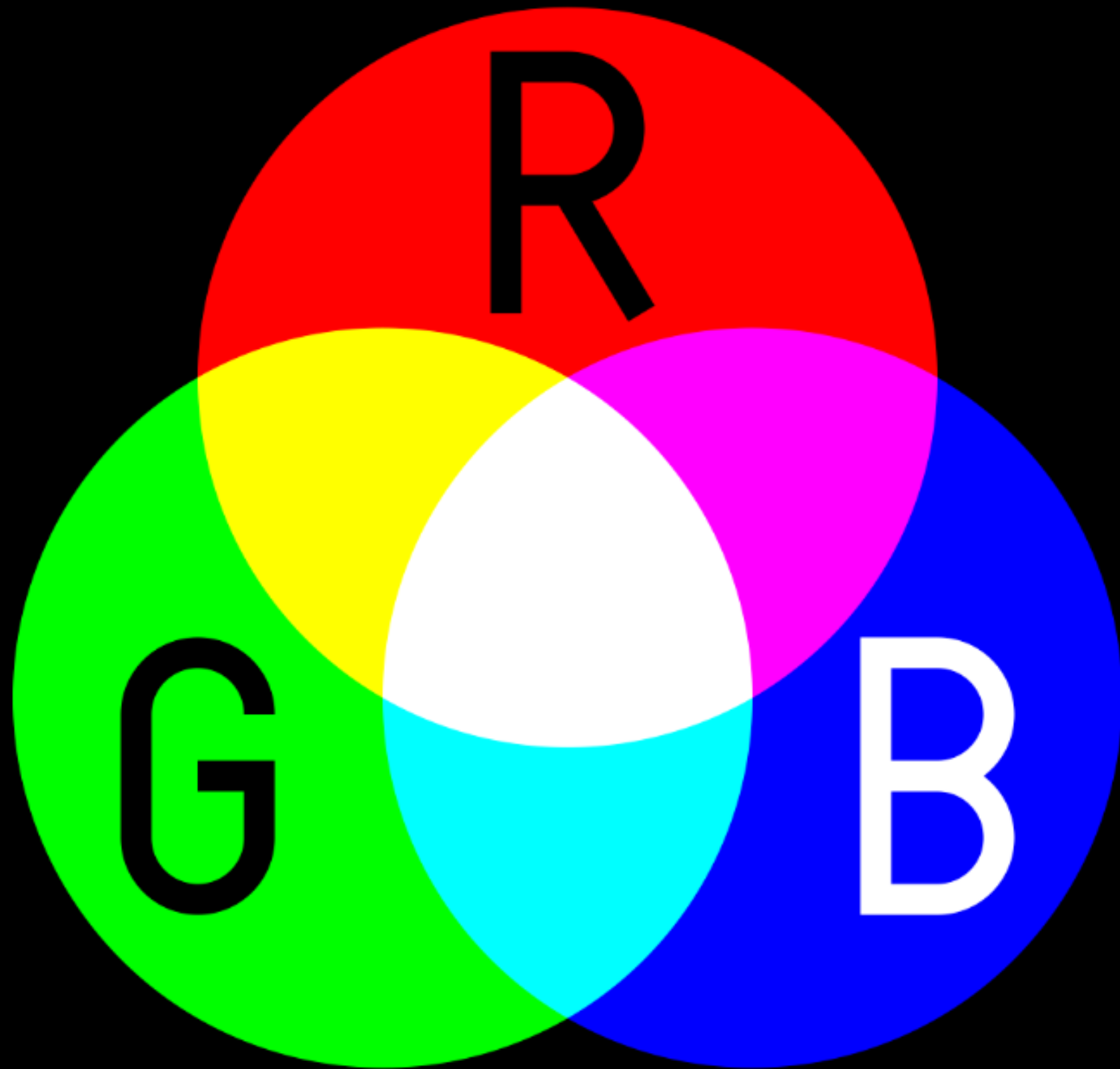


Code為Chatgpt-4協助編寫
因為實力不夠，想出方法後程式碼50%來自
gpt，50%為我簡略完成後丟給gpt優化。

製作者：
葉治嘉-AN4111071

簡報日期：
2023年9月16日

想法和目標



想法

- 在白紙上寫下想要傳遞的訊息後拍照儲存後丟入程式碼進行加密並傳遞。
- 利用C#讀取圖片後，一格一格象素打亂，並將想要傳遞的訊息的象素位置儲存形成鑰匙。
- 讀取鑰匙中的訊息後得到所有鑰轉變顏色的象素位置，將所有位置的象素一一轉變為相同顏色。

目標

- 傳遞訊息給特定人物。
- 避免個人資訊在傳遞時被竊取。

加密程式碼

```
public void Encode()
{
    Bitmap image = new Bitmap(@"C:\Users\qoo09\OneDrive\program\C#\RGBTransform\RGBTransform\Example.jpg"); // 加載圖像

    Random rand = new Random(); // 初始化隨機數生成器

    StringBuilder storage = new StringBuilder(); // 儲存需要的資訊

    // 鎖定圖像的部分，進行讀寫操作
    BitmapData bmpData = image.LockBits(new Rectangle(0, 0, image.Width, image.Height), ImageLockMode.ReadWrite, image.PixelFormat);
    IntPtr ptr = bmpData.Scan0;
    int bytes = Math.Abs(bmpData.Stride) * image.Height;
    byte[] rgbValues = new byte[bytes];
    Marshal.Copy(ptr, rgbValues, 0, bytes); // 複製圖像像素值到 byte 陣列

    for (int y = 0; y < image.Height; y++)
    {
        for (int x = 0; x < image.Width; x++)
        {
            int position = (y * bmpData.Stride) + (x * 3);
            byte blue = rgbValues[position];
            byte green = rgbValues[position + 1];
            byte red = rgbValues[position + 2];
        }
    }
}
```

加密程式碼

```
// 檢查像素值是否小於或等於240
if (red <= 240 || green <= 240 || blue <= 240)
{
    storage.Append("#").Append(x).Append("#").Append(y); // 如果是，則儲存該像素的座標
}

// 更新像素值為隨機色彩
rgbValues[position] = (byte)rand.Next(0, 256);
rgbValues[position + 1] = (byte)rand.Next(0, 256);
rgbValues[position + 2] = (byte)rand.Next(0, 256);
}
}

Marshal.Copy(rgbValues, 0, ptr, bytes); // 將更新後的 byte 陣列複製回 Bitmap
image.UnlockBits(bmpData); // 解鎖圖像資料

image.Save(@"C:\Users\qoo09\OneDrive\program\C#\RGBTransform\RGBTransform\obj\output.jpg"); // 儲存新圖像
File.WriteAllText(@"C:\Users\qoo09\OneDrive\program\C#\RGBTransform\RGBTransform\obj\Key.text", storage.ToString()); // 儲存關鍵資訊
}
```

解密程式碼

```
public void Uncode()
{
    string content = File.ReadAllText(@"C:\Users\qoo09\OneDrive\program\C#\RGBTransform\RGBTransform\obj\Key.text"); // 讀取儲存的關鍵資訊
    Bitmap image = new Bitmap(@"C:\Users\qoo09\OneDrive\program\C#\RGBTransform\RGBTransform\obj\output.jpg"); // 加載圖像

    BitmapData bmpData = image.LockBits(new Rectangle(0, 0, image.Width, image.Height), ImageLockMode.ReadWrite, image.PixelFormat);
    IntPtr ptr = bmpData.Scan0;
    int bytes = Math.Abs(bmpData.Stride) * image.Height;
    byte[] rgbValues = new byte[bytes];
    Marshal.Copy(ptr, rgbValues, 0, bytes);

    string sum = "";
    int count = 0;
    int x = 0, y = 0;

    for (int i = 1; i < content.Length; i++)
    {
        if (content[i] != '#')
        {
            sum += content[i]; // 如果當前字符不是'#'，將其添加到 sum
        }
        else
        {
            count++;

            if (count == 1)
```

解密程式碼

```
        if (count == 1)
            x = int.Parse(sum); // 解析 X 座標
        else if (count == 2)
        {
            y = int.Parse(sum); // 解析 Y 座標
            count = 0;

            int position = (y * bmpData.Stride) + (x * 3);
            rgbValues[position] = 0;
            rgbValues[position + 1] = 0;
            rgbValues[position + 2] = 0; // 將相應的像素設置為黑色
        }

        sum = "";
    }
}

Marshal.Copy(rgbValues, 0, ptr, bytes);
image.UnlockBits(bmpData);

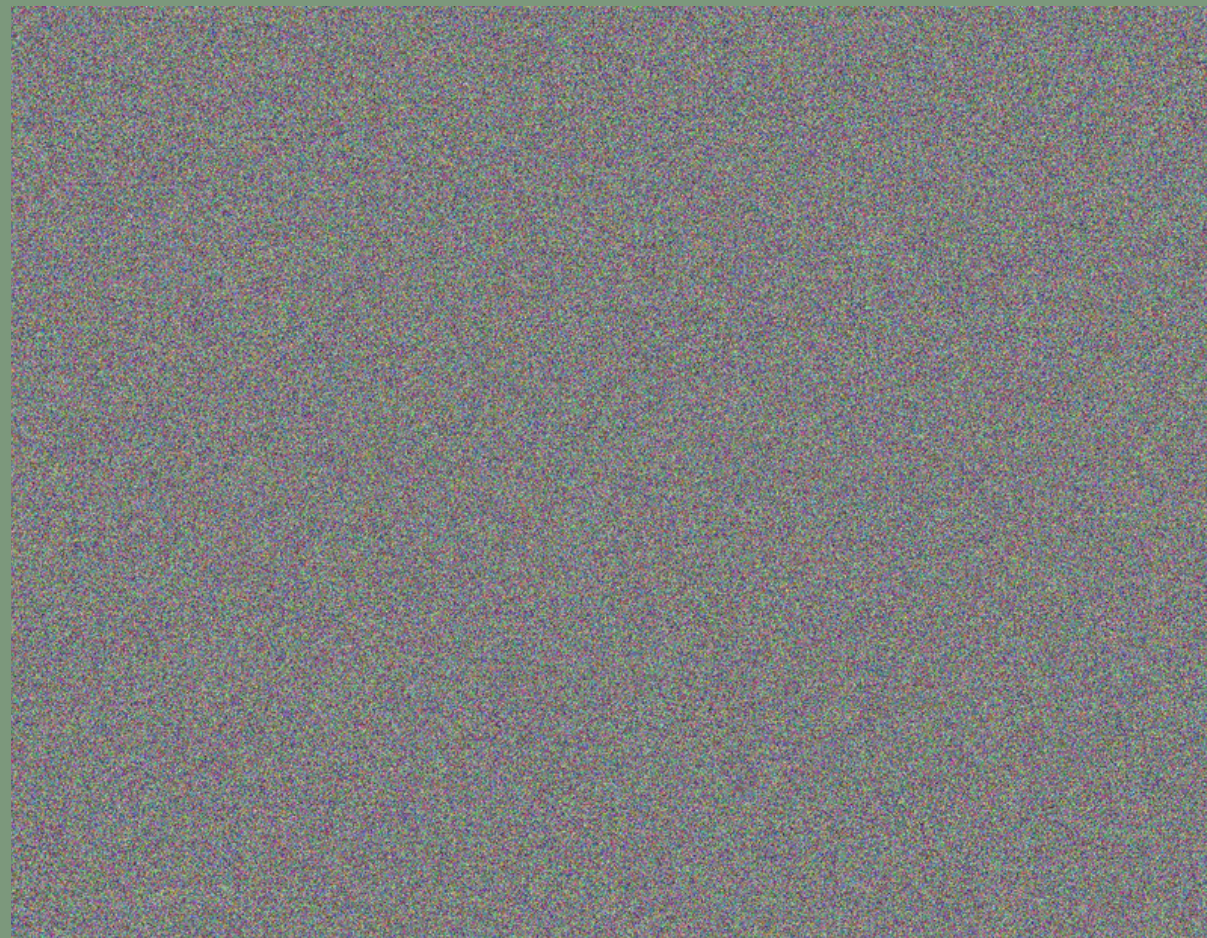
image.Save(@"C:\Users\qoo09\OneDrive\program\C#\RGBTransform\RGBTransform\output.jpg"); // 儲存解碼後的圖像
}
```


結果展示

原圖



加密後



解密



程式碼下載:https://drive.google.com/file/d/1A80iPJqZHzK-qcLkj0rr-YLSouV_6Okf/view?usp=sharing