

國立臺灣大學

計算機程式設計(宜大)作業一

GRE 背單字之旅

Journey of the GRE Word Memorization

許哲榕

R07544033

授課教師：鄭士康 博士

July 2021

## 一、題目動機與發想

### （一）題目動機

題目的動機是因為近日花費不少時間在準備 GRE 留學考試，而 GRE 考試中的語文部分的單字又以其惡名昭彰地難以背誦聞名，這段時間每日都耗費時間和背單字的軟體及應用程式為伍。故由此發想，開始思考若是自己製作與開發一款背單字程式，那麼有哪些功能會是非常必須的？又有哪些功能有了會很方便、可以提升背單字效率？在有了這些開發的契機與動力後，便開始思考具體應該會有哪些內容涵蓋在一款合格的背單字軟體。

### （二）內容發想

首先，私以為一款合格的背 GRE 英文單字程式，一定要有可以自己輸入單字清單或自己鍵入單一單字的功能，因為既定的單字表不一定符合每一位使用者的英文背景與程度；其次，應該要有方便使用的鍵盤快捷鍵可以快速切換，如此一來便能迅速建立起對每一個英文單字的第一印象；第三，必須得有隨機的模式，才能夠方便確認自己有真實記憶單字，而非仰賴單字清單中的順序去建立的短期記憶；四，若有熟悉度的紀錄，則能準確地記錄自己對於不同單字當前的熟絡程度；最後，英文單字發音的功能必不可少，少了英文單字的發音，將全然仰賴閱讀字母的感覺，透過五感綜合起來學習英文單字應是較好的策略。綜上，對於自己想要在作業一中開發的 GRE 背單字之旅程式，應該基本具備這幾項功能及內容。

## 二、構想解說

### (一) 詳說構想及功能

以下透過列表之方式，一一說明打算構築的各項程式功能：

1. 背單字程式歡迎語的顯示	讓使用者可以快速進入背 GRE 單字的情境，感受此款背單字軟體的魅力。
2. 英文單字拼字的顯示	要有大尺寸的拼字顯示，方便使用者記憶單字。
3. 英文單字中文翻譯的顯示	盡量以簡潔的形式，提供最為必要的資訊，避免記憶力的多於浪費。
4. 輸入單字清單的選項	要可以讓使用者輸入自己所需要的單字清單，構想上以.txt 檔為理想，因為在處理檔案流的字串上應該會相對簡單。
5. 填寫鍵入單字的盒子	必須有地方讓使用者鍵入單個單字來輸入，以查缺補漏單字清單所沒有的單字，在輸入的格式上同樣要有所規範，以便以同樣的模式處理字串。
6. 輸入鍵入單字的選項	透過選項讓使用者輸入的字串得以進入資料結構中，構想上以 ArrayList 為較理想，因為處理簡單且不限定長度。
7. 回到前一個單字的選項	讓使用者回到前一個單字，隨機模式

	中的前一個單字也是隨機的。
8. 進入下一個單字的選項	讓使用者進入下一個單字，隨機模式中的前一個單字也是隨機的。
9. 顯示與隱藏中文翻譯的選項	若使用者是在復習階段，那麼可以先將中文隱藏起來的功能便將至關重要。
10. 單字熟悉度的顯示	透過熟悉度的紀錄，方便使用者知道自己對於這個單字的熟悉程度。構想上應該要有上下限，而非無限發散。
11. 確認熟悉單字的選項	讓使用者確認自己熟悉此單字，也將自動連接下一個單字的功能。
12. 確認對單字陌生的選項	讓使用者確認自己不熟悉此單字，也將自動連接下一個單字的功能。
13. 發音模式的勾選盒子	勾選後，每次進入不同單字，將有英文語音的單字示範。
14. 隨機模式的勾選盒子	勾選後，可以使單字的排序成為亂序，構想上應該要排除每一輪中已經有背過的單字，但在此次的作業一中暫以完全隨機的模式來設計。
15. 單字輸入的規範說明	規範使用者要以@來分隔英文單字與中文解釋，且不同單字間以換行來做為區隔，在檔案流的處理中，要特別注意最後一個分隔單位有時候會是空白的

	字串。
--	-----

### 三、程式測試規劃

以下同樣透過列表之方式，一一說明打算測試規劃的各點：

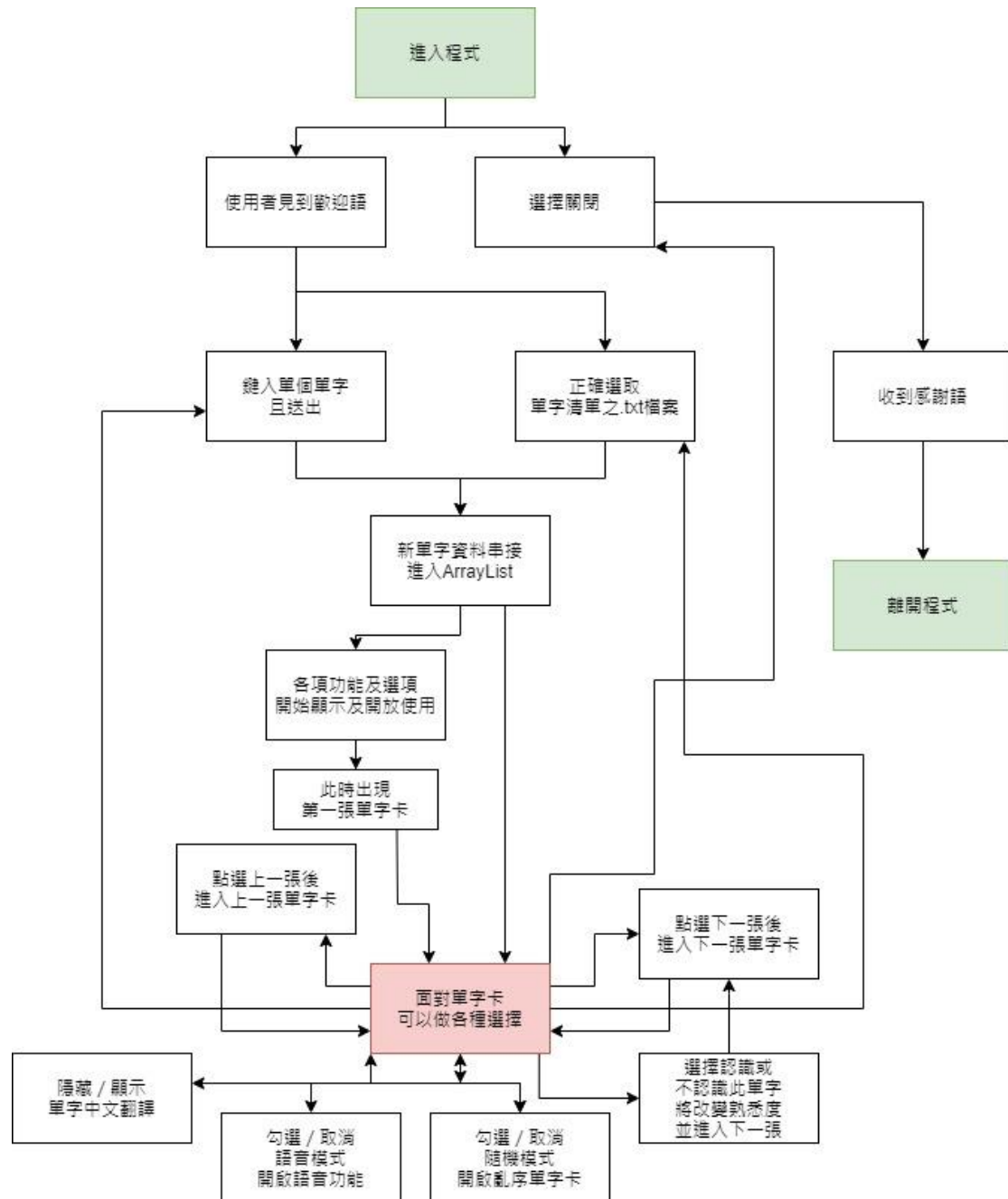
1. 背單字程式歡迎語的顯示	注意顯示時不要超出版面，且在歡迎語顯示的階段應該隱藏部分顯示和選項，方便使用者直觀地去點選輸入單字清單抑或是鍵入單字。
2. 英文單字拼字的顯示	注意放入 ArrayList 後，顯示是否正確，應考量 index 有沒有放錯，以及邊緣資料的處理。
3. 英文單字中文翻譯的顯示	同樣應注意放入 ArrayList 後，顯示是否正確，應考量 index 有沒有放錯，以及邊緣資料的處理。且要注意使用隱藏功能時，能否正確隱藏／顯示。
4. 輸入單字清單的選項	注意使用者有沒有成功選取資料夾，若沒有則不應有反應；且注意是否選取規範的檔案格式，不過本次作業一並未處理後者，僅以文字說明指引使用者。
5. 填寫鍵入單字的盒子	透過預設的字元來指引使用者，填入符合規範的單字，且點選輸入後應檢查是

	否再次跳回預設的字元。要注意測試在填寫單字拼寫時，要先暫時關閉所有有以英文字母為快捷鍵的功能。
6. 輸入鍵入單字的選項	注意測試確認在點選後，鍵入盒子有沒有回到原初的狀態，且各項被暫時關閉的選項有沒有回復；另外，應注意是否有正確地讓單字加入 ArrayList 結構中。
7. 回到前一個單字的選項	注意測試在首張單字卡的情況下，使用此選項是否逸出 index。也要測試隨機模式和發音模式下此選項能否正確反映單字出來。
8. 進入下一個單字的選項	注意測試在末張單字卡的情況下，使用此選項是否逸出 index。也要測試隨機模式和發音模式下此選項能否正確反映單字出來。
9. 顯示與隱藏中文翻譯的選項	注意顯示和隱藏功能是否良善，不論在不同模式下。
10. 單字熟悉度的顯示	注意熟悉度的上下界，注意是否有正確搭上該份單字，還是次序有所亂掉。
11. 確認熟悉單字的選項	注意特殊模式下，使用此功能能否正確連接此選項隱含的下一張功能。

12. 確認對單字陌生的選項	測試特殊模式下，使用此功能能否正確 連接此選項隱含的下一張功能。
13. 發音模式的勾選盒子	確認是否是正確的對應該單字。
14. 隨機模式的勾選盒子	確認排序是否真正隨機，且有沒有在 index 計算中出現問題，導致某些單字 永遠不會輪到。
15. 單字輸入的規範說明	測試相關說明是否足夠直觀，得以減少 使用者進行設計意圖外的行為。

## 四、流程圖

以下透過圖片之方式，說明 GRE 背單字之旅程式之流程圖：





## 五、程式列表

以下主要針對 WindowsFormsApp 部分的程式進入列表與說明，多為自動生成的 Designer.cs 與 Program.cs 略過不提。程式之功能以圖片中的綠色註釋字句說明：



```

namespace WindowsFormsApp1
{
    3 個參考
    public partial class GRE : Form
    {
        //首先是一些全域都可以用的變數或物件
        int intCount1 = 0;
        int intCount2 = 1;
        string strList = ""; //處理檔案流
        string[] strarrayWords; //處理檔案流
        public ArrayList words = new ArrayList(); //放單字卡資料
        public ArrayList familiar = new ArrayList(); //放單字卡熟悉度資料
        public SpeechSynthesizer sayEng = new SpeechSynthesizer(); //用來提供語音功能

        1 個參考
        public GRE()
        {
            InitializeComponent();
        }
    }
}

```

1 個參考

```
private void Form1_Load(object sender, EventArgs e)
{
    //先隱藏還不想干擾使用者的部分選項和標籤
    button1.Enabled = false; //下一個單字
    button2.Enabled = false; //上一個單字
    button5.Enabled = false; //顯示與隱藏中文翻譯
    button6.Enabled = false; //學會了
    button7.Enabled = false; //還不會
    label11.Visible = false; //英文拼字
    label12.Visible = false; //中文翻譯
    label13.Visible = false; //當前單字序
    label14.Visible = false; //單字序的橫槓
    label15.Visible = false; //總輸入單字數
    label16.Visible = true; //顯示：歡迎語
    label18.Visible = false; //熟悉度文字
    label19.Visible = false; //熟悉度數值
    checkBox1.Visible = false; //隨機模式
    checkBox2.Visible = false; //發音模式
    sayEng.Rate = 4; //確認英文發音語速，4為略快
}
```

```
1 個參考
private void button1_Click(object sender, EventArgs e)
{
    if (checkBox1.Checked == false) //非隨機模式中的下一張
    {
        if (intCount2 + 2 < words.Count) //計算下一張的index，注意邊緣逸出
        {
            intCount1 += 2;
            intCount2 += 2;
        }

        label11.Text = Convert.ToString(words[intCount1]); //顯示英文拼字
        if (checkBox2.Checked == true){ //語音模式的話，則放送該單字語音
            sayEng.Speak(label11.Text);
        }
        label12.Text = Convert.ToString(words[intCount2]); //顯示中文翻譯
        label13.Text = Convert.ToString((intCount2 + 1) / 2); //處理當前單字序
        label19.Text = Convert.ToString(familiar[intCount1]); //顯示熟悉度數值
    }
    else { //隨機模式中的下一張
        Random ran = new Random();
        int intRan = ran.Next(1, (words.Count / 2)+1);
        intCount2 = (intRan * 2 - 1);
        intCount1 = intCount2 - 1;
        label11.Text = Convert.ToString(words[intCount1]);
        if (checkBox2.Checked == true)
        {
            sayEng.Speak(label11.Text);
        }
        label12.Text = Convert.ToString(words[intCount2]);
        label13.Text = Convert.ToString((intCount2 + 1) / 2);
        label19.Text = Convert.ToString(familiar[intCount1]);
    }
}
```

1 個參考

```

private void button3_Click(object sender, EventArgs e) //列表輸入
{
    if (openFileDialog1.ShowDialog() == DialogResult.OK) //確認檔案選取
    {
        var wordsList = new StreamReader(openFileDialog1.FileName); //檔案流輸入處理
        strList = wordsList.ReadToEnd();
        strarrayWords = strList.Split('@', '\n'); //字串分割處理
        for (int i=0;i<strarrayWords.Length;i++) { //array轉arraylist
            words.Add(strarrayWords[i]);
            familiar.Add(0);
        }
        if ((string)words[(words.Count - 1)] == "") { //處理檔案末尾空白字串
            words.RemoveAt((words.Count - 1));
            familiar.RemoveAt((words.Count - 1));
        }
        intCount1 = 0;
        intCount2 = 1;
        label1.Text = Convert.ToString(words[intCount1]);
        label2.Text = Convert.ToString(words[intCount2]);
        button1.Enabled = true; //以下開始顯示各種功能之開放
        button2.Enabled = true;
        button5.Enabled = true;
        button6.Enabled = true;
        button7.Enabled = true;
        label1.Visible = true;
        label2.Visible = true;
        label3.Visible = true;
        label4.Visible = true;
        label5.Visible = true;
        label6.Visible = false;
        label8.Visible = true;
    }
}

```

```

private void GRE_KeyDown(object sender, KeyEventArgs e) //處理各選項之快捷
{

```

```

    if (e.KeyCode == Keys.D) {
        button1.PerformClick();
    }
    if (e.KeyCode == Keys.A)
    {
        button2.PerformClick();
    }
    if (e.KeyCode == Keys.S) {
        button5.PerformClick();
    }
    if (e.KeyCode == Keys.K)
    {
        button6.PerformClick();
    }
    if (e.KeyCode == Keys.N)
    {
        button7.PerformClick();
    }
}

```

```

1 個參考
private void button6_Click(object sender, EventArgs e) //熟悉此單字功能。處理熟悉度
{
    int intFamiliar =(int) familiar[intCount1]; //計算熟悉度數值
    if (intFamiliar < 5) {
        familiar[intCount1] = ++intFamiliar;
    }
    label9.Text= Convert.ToString(familiar[intCount1]);
    button1.PerformClick(); //確認熟悉後自動跳轉下一張
}

```

## 六、程式測試執行結果

程式在非語音模式下運行十分順利，儘管在語音模式下略感卡頓。不過各項功能在各種模式下，都同樣運作良好。尚待修正與改進之處主要在於檔案的匯入，當前在檔案的選取上尚未設定讓使用者無從選擇錯誤之檔案類型，所提供的單字清單模式目前也侷限在以@符號來分割一種；也未在鍵入單字處，設置檢查，倘若使用者輸入不合規格之單字，程式仍然可以正常運作，但也就無法正確顯示應該的單字卡樣貌，難以達致正確使用的學習效果。綜合考量上述幾點，未來在作業二應改進在檔案流的處理，並且在使用者介面上做出更新，當前作業一的版本長相過於陽春。

## 參考文獻

- Microsoft Docs (2018), 'How to separate strings using String.Split in C#',  
<https://docs.microsoft.com/en-us/dotnet/csharp/how-to/parse-strings-using-split>.
- Microsoft Docs (2019), 'How to: Open files with the OpenFileDialog',  
<https://docs.microsoft.com/en-us/dotnet/desktop/winforms/controls/how-to-open-files-using-the-openfiledialog-component?view=netframeworkdesktop-4.8>.
- Microsoft Docs (N/A), 'SpeechSynthesizer Class',  
<https://docs.microsoft.com/en-us/dotnet/api/system.speech.synthesis.speechsynthesizer?view=netframework-4.8>.