

# SimpleC Incremental Editor Final Report

(New)

日期：2025-06-20

111550133 蔡宇炫

## 一、專案緣起與目標


本專案是為了簡化 C 語言新手的學習歷程，讓大家在寫 C code 的時候可以及時發現資料流異常、拼字錯誤、語法遺漏等小問題，也能快速補全關鍵字與變數、函數名稱。這樣寫程式就像用現代 IDE 一樣更有效率。

## 二、功能設計

### 1. 編輯器基本操作

- 多行文字編輯：支援貼上、刪除、複製、Undo/Redo (TextBox 內建)。
- 自動縮排：按 Enter 會延續前一行縮排，如果上一行有 {，新行會多縮排一層；輸入 } 會自動縮回。
- 行號顯示：左側 Template 區塊每一行前面都會顯示行號。

### 2. 語法高亮 & 行資訊

- 關鍵字高亮：Template 區塊會將 int、float、if、while 等 C 關鍵字用藍色字體顯示。
- 錯誤旗幟：有錯誤的行，行號前會有  紅色旗幟。
- 縮排保留：左邊展示與主編輯區一致的縮排，程式結構一目了然。

編輯區語法高亮功能（為何未實作）

雖然左側 Template 區塊有高亮，但主編輯區 (TextBox) 目前無法直接顯示語法顏色。原因主要是 Avalonia TextBox 不支援多色文字 (RichText)，要額外用第三方套件或自己處理複雜度較高，加上本次專案重心放在資料流異常檢查與自動補全，所以暫未優先實作此功能。若日後有需要會再補上。

### 3. 自動補全 (Auto-complete)

- C 關鍵字：只要輸入前綴兩個字母以上 (例如 fl)，下方就會跳出補全清單 (例如 float)。
- 變數、函數名稱：會自動偵測程式裡宣告過的變數/函數，也能補全 (例如 foo)。
- Tab、Enter 或滑鼠點擊選項都能補全。

### 4. 語法偵測與修正

- **拼字錯誤提示**：對關鍵字進行模糊比對（Levenshtein distance 1），如 `imt` 會提示「你是想輸入 `int` 嗎？」。
- **語法錯誤偵測**：
  - **分號檢查**：行末沒分號會警告。
  - **括號/大括號平衡**：缺少括號、大括號會提示缺幾個。
  - **不合法識別字**：像是 `2abc`、`a-bc`、`foo(int` 都會提示詞法錯誤。
  - **資料流異常**：
    - **UR (Use before Reference)**：未宣告直接用。
    - **DD (Double Declaration)**：重複宣告。
    - **DU (Defined but Unused)**：宣告但沒用過。
- **錯誤行同步**：所有錯誤都會顯示於右側異常列表，並同步於 **Template** 行旗幟。

## 5. 視覺化介面設計

- **三區佈局**：左邊是 **Template** 行資訊（含高亮/旗幟），中間是主要編輯 **TextBox**，右側是異常列表與 **AST** 樹狀結構。
- **分隔條可調整**：三區之間可以拖曳分隔線，自訂大小。
- **配色設計**：關鍵字藍色，行號灰色，旗幟紅色但顏色偏淡。

## 6. 即時 AST 樹狀結構顯示 (Bonus)

- **AST 樹**：按右側按鈕可以顯示/隱藏 **AST** 樹（程式結構），適合觀察簡單語法結構。
- **自動更新**：每次分析時會自動刷新 **AST**。

## 三、功能對需求對照

項目	是否完成	備註
文字編輯基本操作	✓	多行輸入/Undo/Redo
語法高亮	⚠	左側 <b>Template</b> 實作
自動縮排	✓	4空格一層
拼字錯誤提示	✓	關鍵字模糊比對
語法錯誤偵測	✓	分號、括號平衡
詞法錯誤提示	✓	識別字檢查
資料流異常 (UR/DD/DU)	✓	變數與函數皆檢查
自動補全（關鍵字）	✓	Tab/Enter/點擊可選
補全（變數/函數）	✓	已宣告的皆可補全
錯誤提示視覺化	✓	旗幟+右側異常清單

AST 即時顯示	✓	右側可開關 Bonus 功能
編輯區語法高亮	✗	TextBox 未做語法顏色

## 四、開發心得與挑戰

這次實作遇到的主要挑戰有三點：

1. **資料流異常檢查**：手動實作 DFA 判斷，如何簡化規則避免誤判同時能涵蓋常見情境，花了最多時間測試跟修正。
2. **高亮邏輯**：C# 的 TextBox 不支援 RichText，考量穩定性只在 Template 區塊實作高亮。
3. **自動補全**：要兼顧使用者習慣（隨打隨建議），同時能判斷已宣告過的變數/函式名稱也能即時建議，過程中調整了多次觸發時機與條件。

整體來說，功能上大多數需求都完成，也有實作 bonus AST，介面直觀、錯誤訊息明確。未來如果要做成商品化的 SDE，還能進一步支援複雜語法與程式片段語法高亮。

## 五、測試案例與功能驗證

以下為主要測資，可逐行展示對應功能：

```
int x;
float y

imt z;           // 拼字錯誤
int x;           // DD

y = foo(x);      // UR(foo)
if (y){          // 缺分號
    x = x + 1;
}

int foo(int a){
    a = a + 1;
}                // 括號/大括號平衡
```

- imt z; 會提示拼字錯誤「你是想輸入 int 嗎」。
- int x; 會標註 DD。
- y = foo(x); 會標註 UR(foo)。
- if (y){ 會警告缺分號。
- 括號/大括號數量不對會標註缺少數量。

- 每一個異常都會對應旗幟和右側列表，點行號也會自動跳到對應行。
- 

## 六、總結

整體來說，這份 SimpleC Incremental SDE 完成了幾乎所有需求，介面直覺，對於 C 語言新手練習非常實用。

過程中練習了資料流分析、自動補全、UI 設計、錯誤偵測等能力，也體會到設計一個好用的程式語言編輯器並不簡單。

希望未來有更多機會可以挑戰更完整的語法分析與即時高亮顯示。

FinalReport