

林林.台灣 | Linwebs - 課程

首頁 / 課程 / Qt / 跨平台圖形化程式開發 - (2)賽車遊戲開發-介面編排設計

跨平台圖形化程式開發 - (2)賽車遊戲開發-介面編排設計

本文章為嘉大資工讀書會第二期課程內容

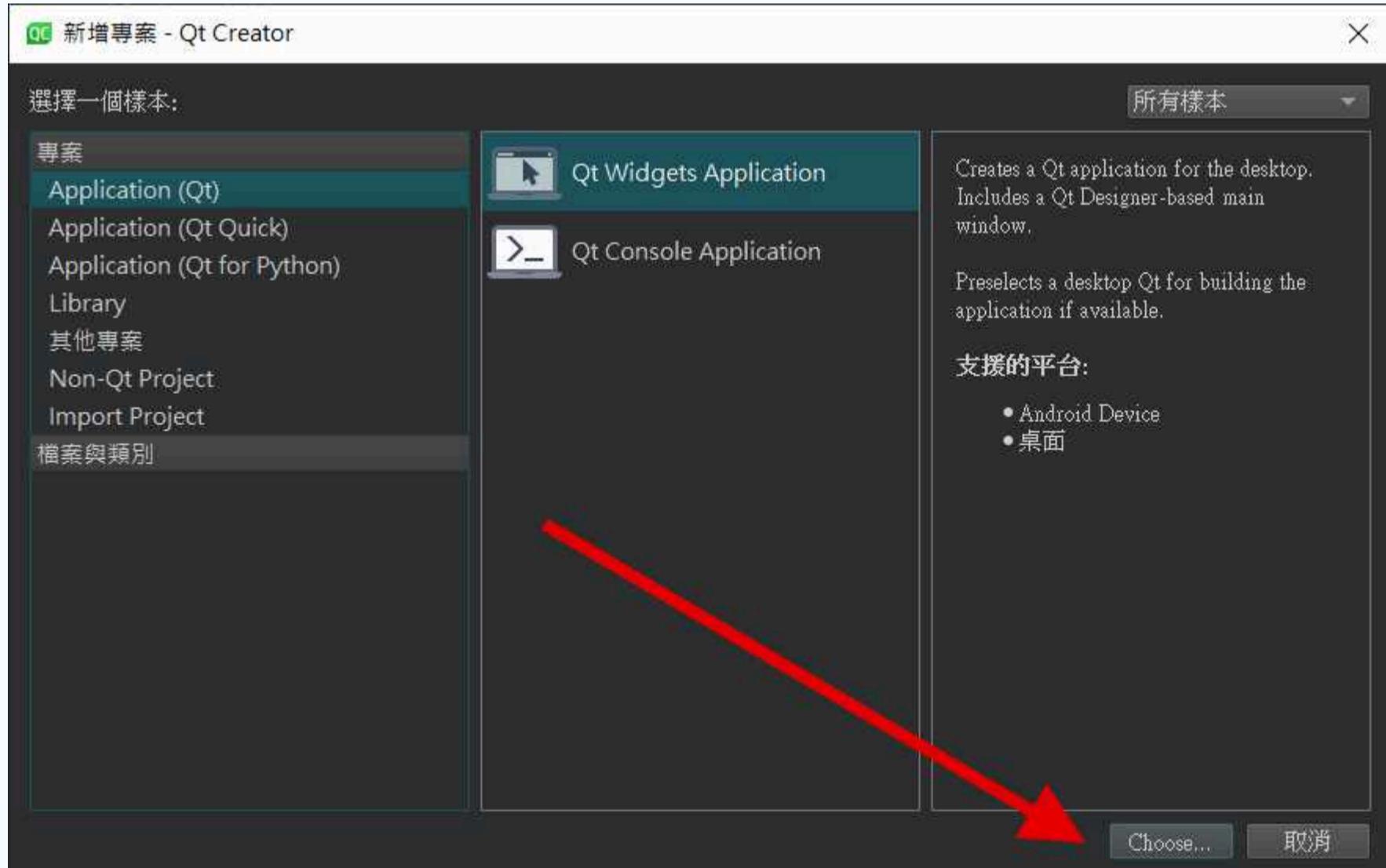
場次 4-1 【跨平台圖形化程式設計開發(二)】跨平台圖形化程式開發

本課程介紹 [【跨平台圖形化程式開發 - \(1\)賽車遊戲開發-介紹】](#)

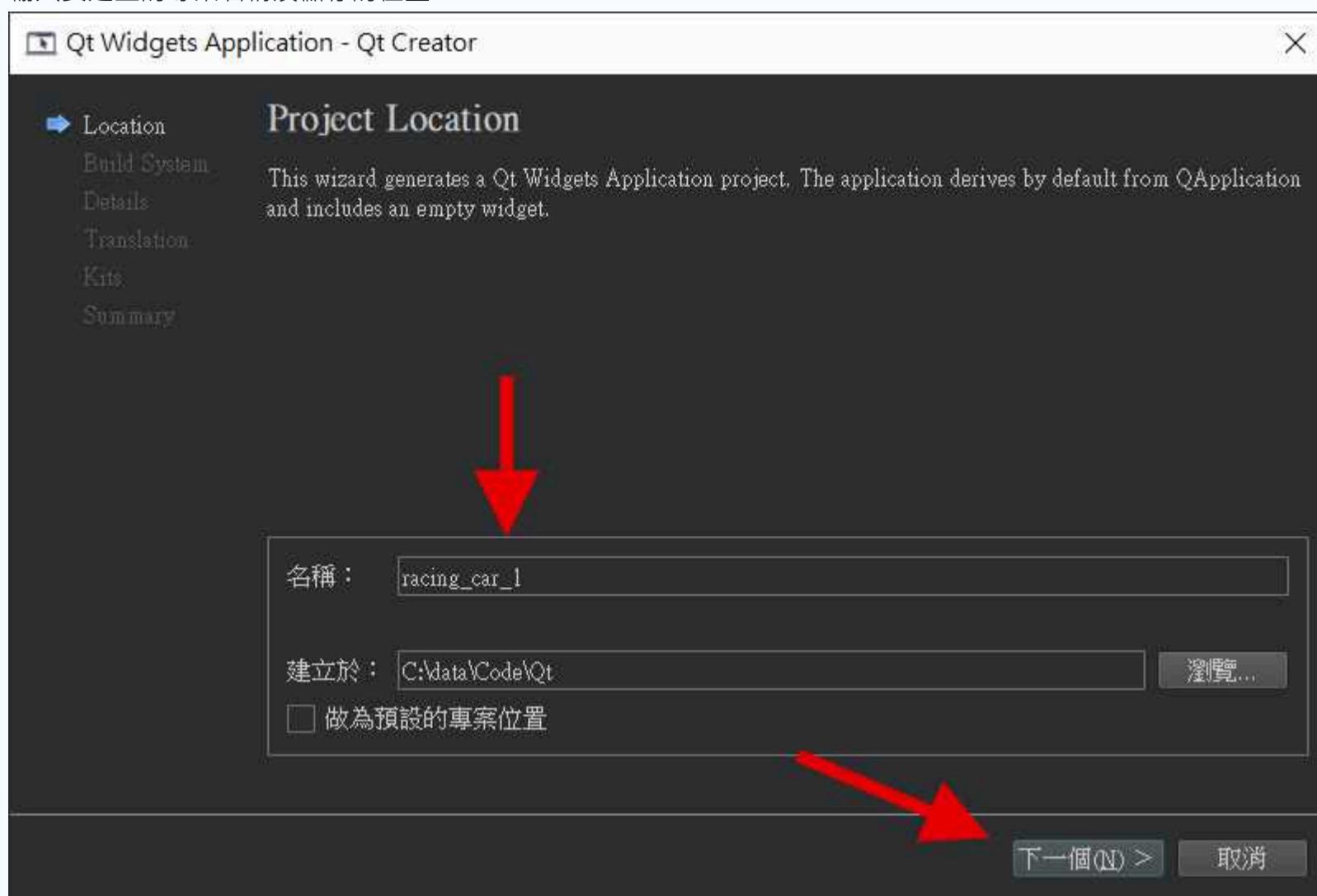
此次課程素材下載: [ncyu_study_racing_car_src.zip](#)

PS: 本課程圖片較多，此頁面圖片有經過壓縮處理，若圖片模糊不清，可點選圖片開啟圖片原始檔

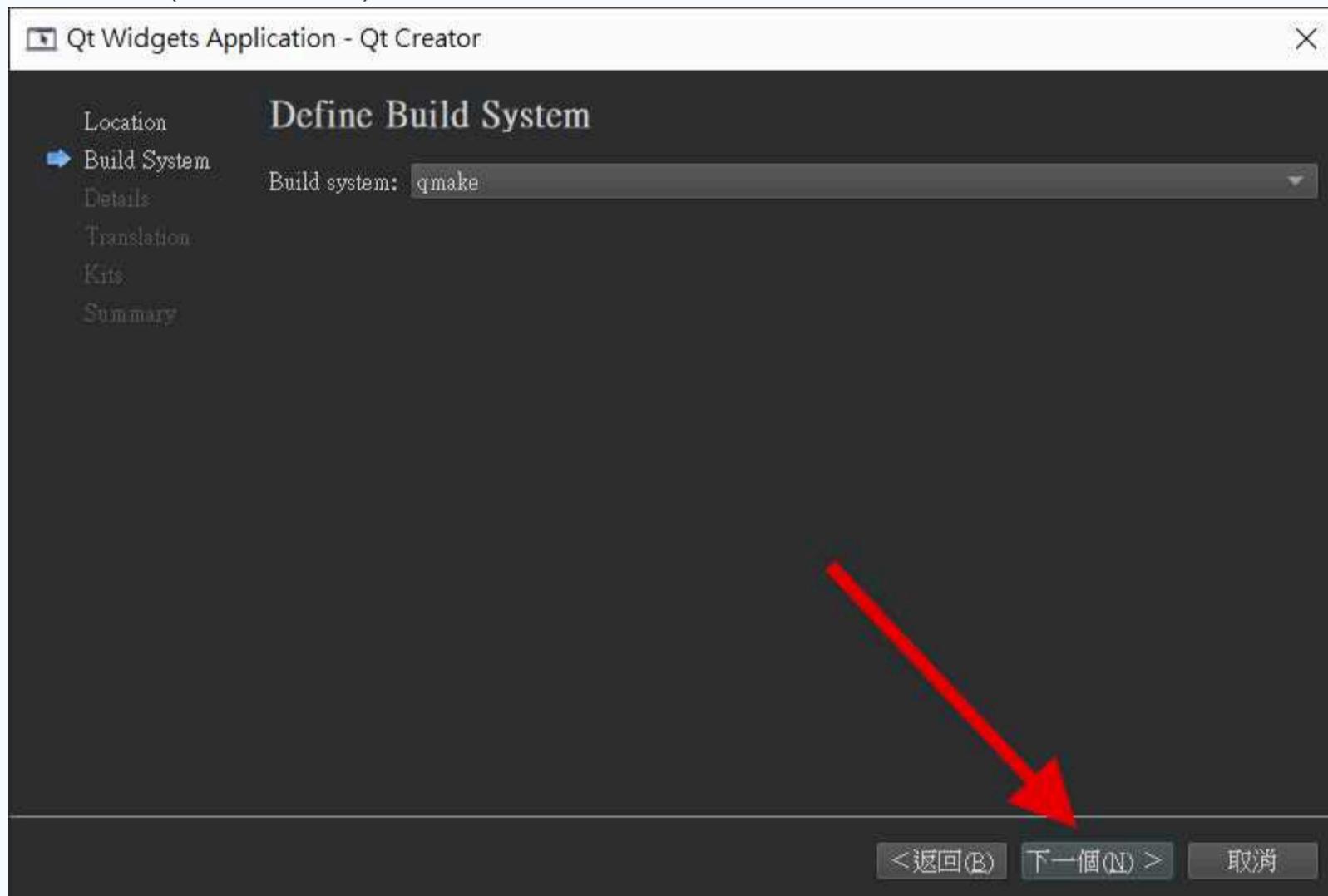
1. 開啟Qt Creator，新增一個專案，選擇【Qt Widgets Application】。



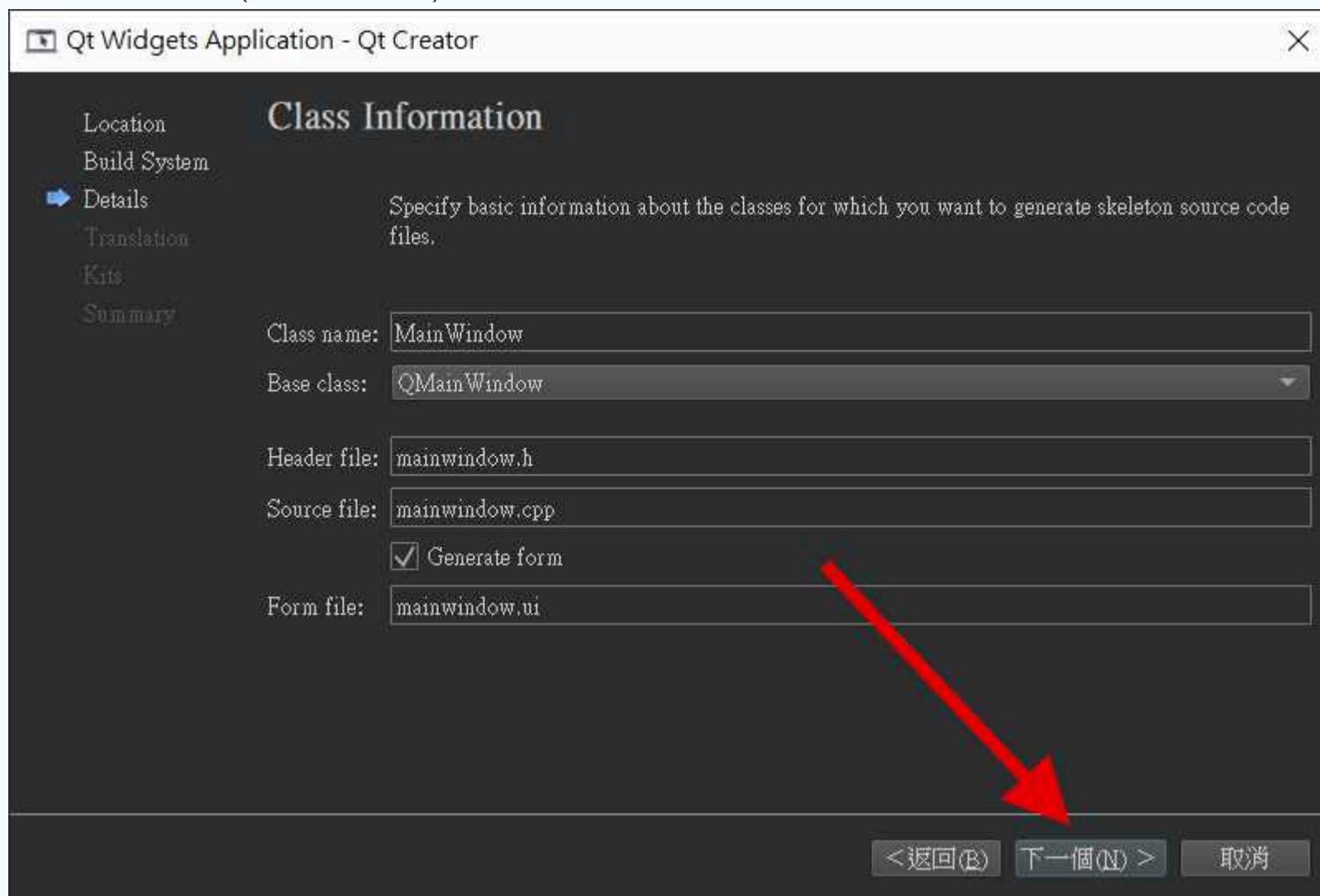
2. 輸入要建立的專案名稱及儲存的位置。



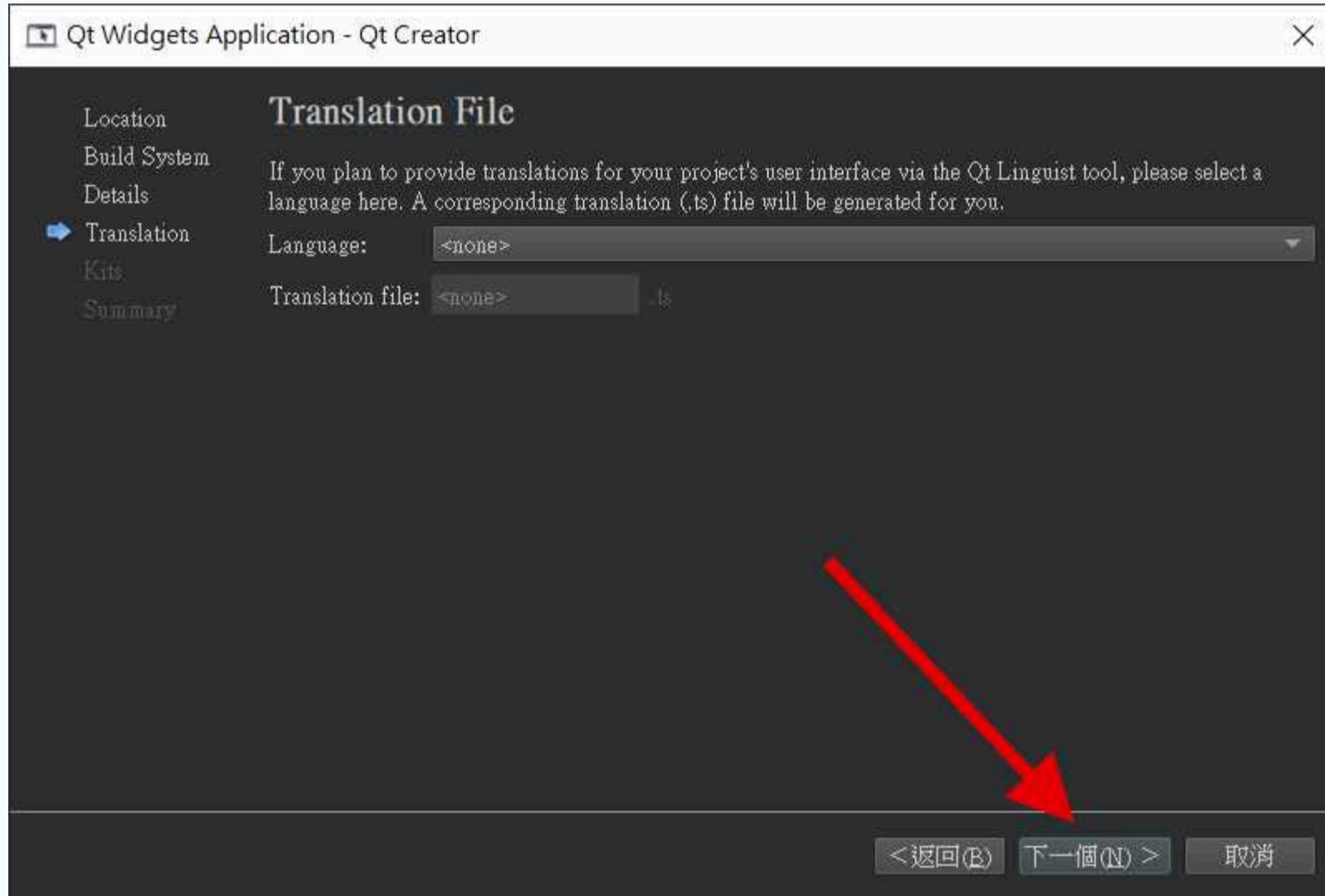
3. 選擇建構系統(在此採用預設值)。



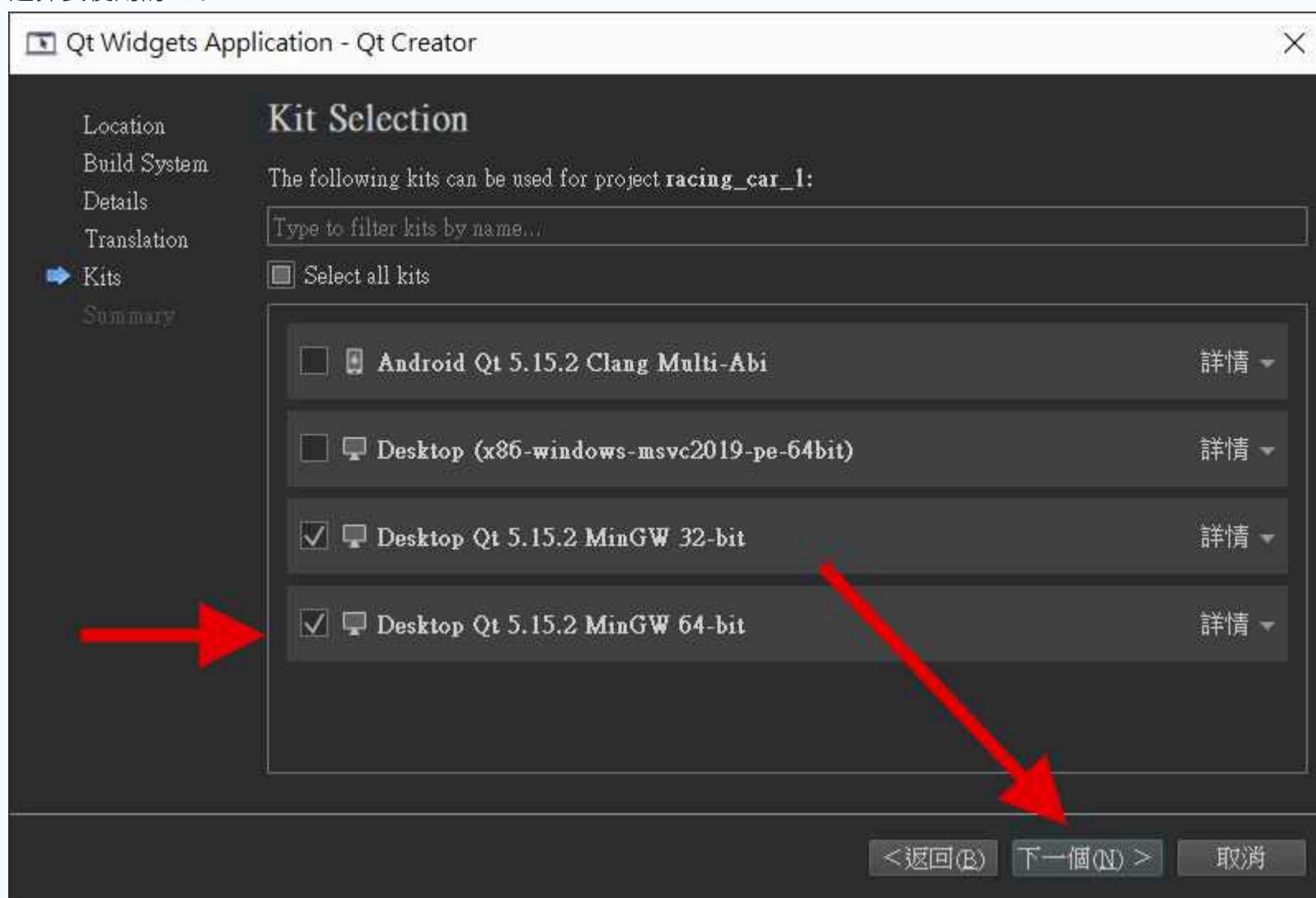
4. 設定初始 class 資訊(在此採用預設值)。



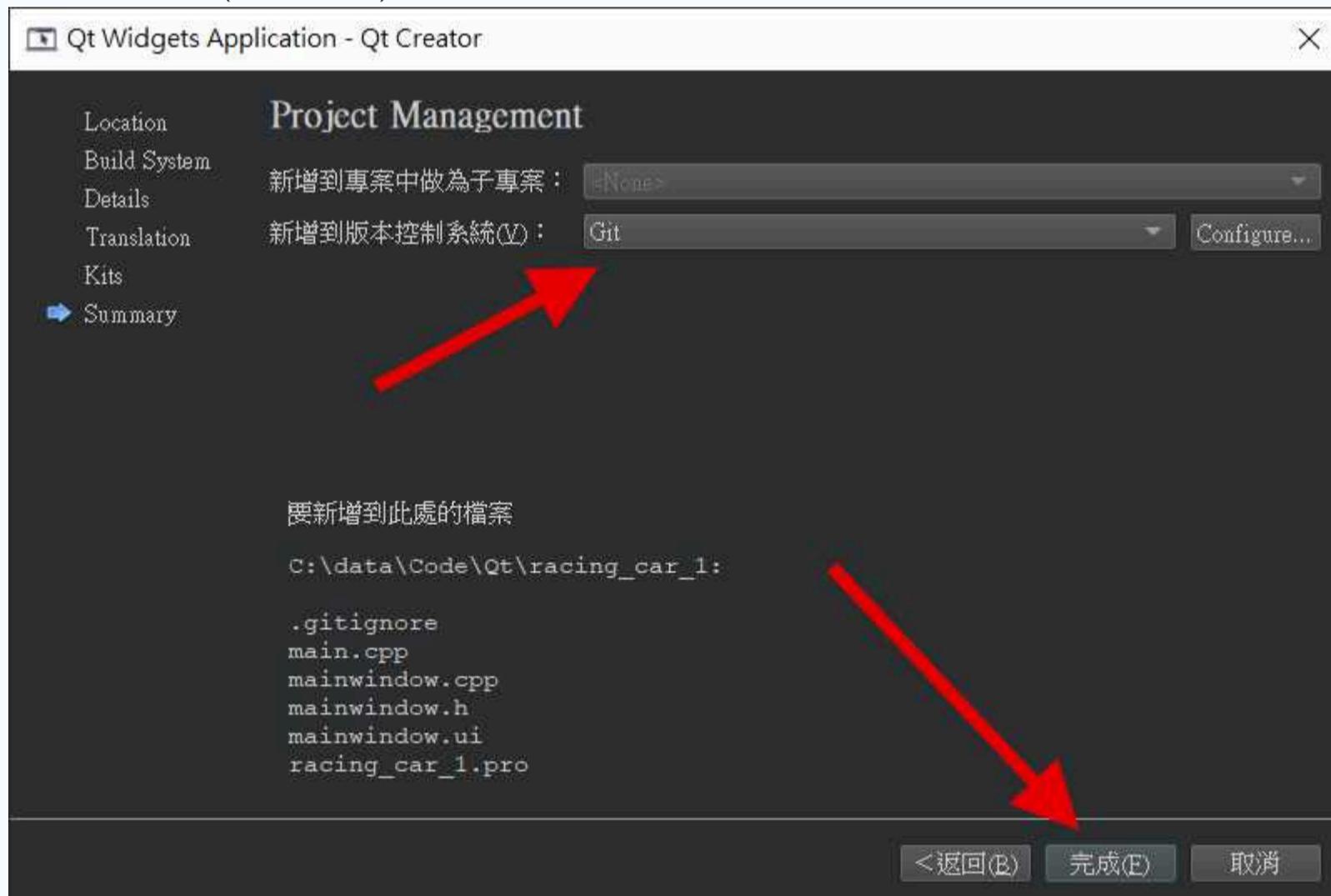
5. 設定翻譯檔(在此不做設定)。



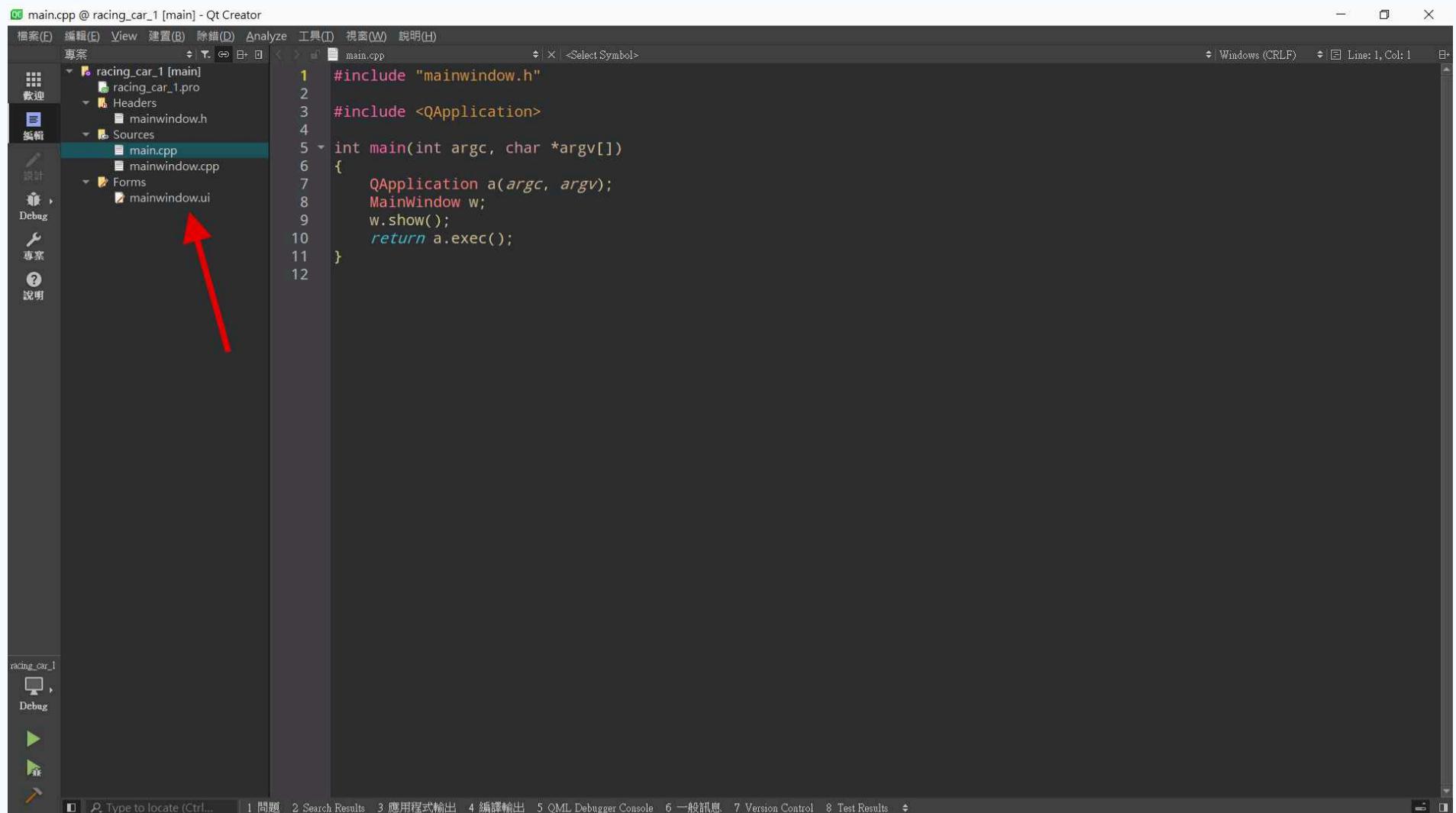
6. 選擇要使用的 Kit。



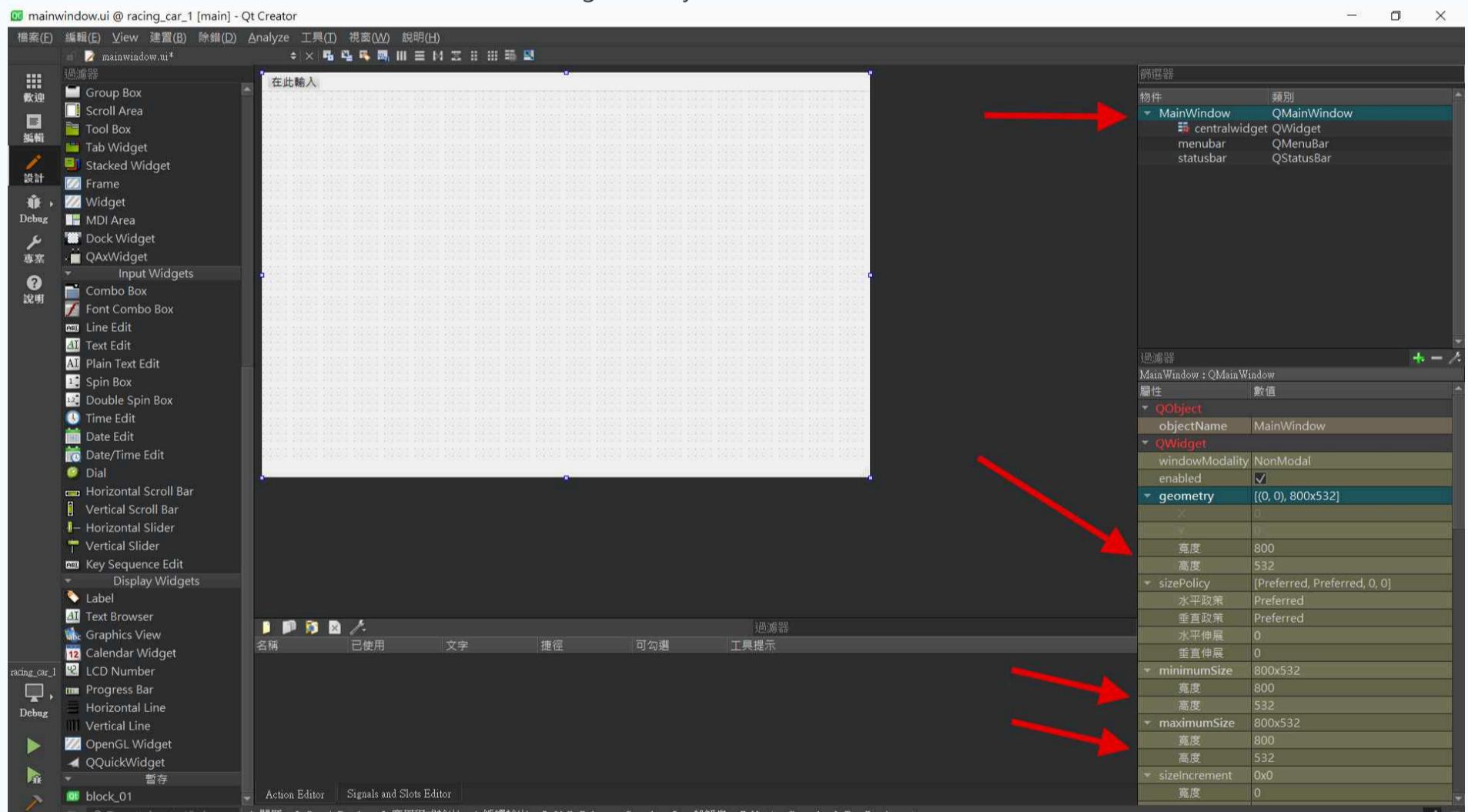
7. 選擇版本控制系統(在此採用 Git)。



8. 選擇【mainwindow.ui】進行介面設計。

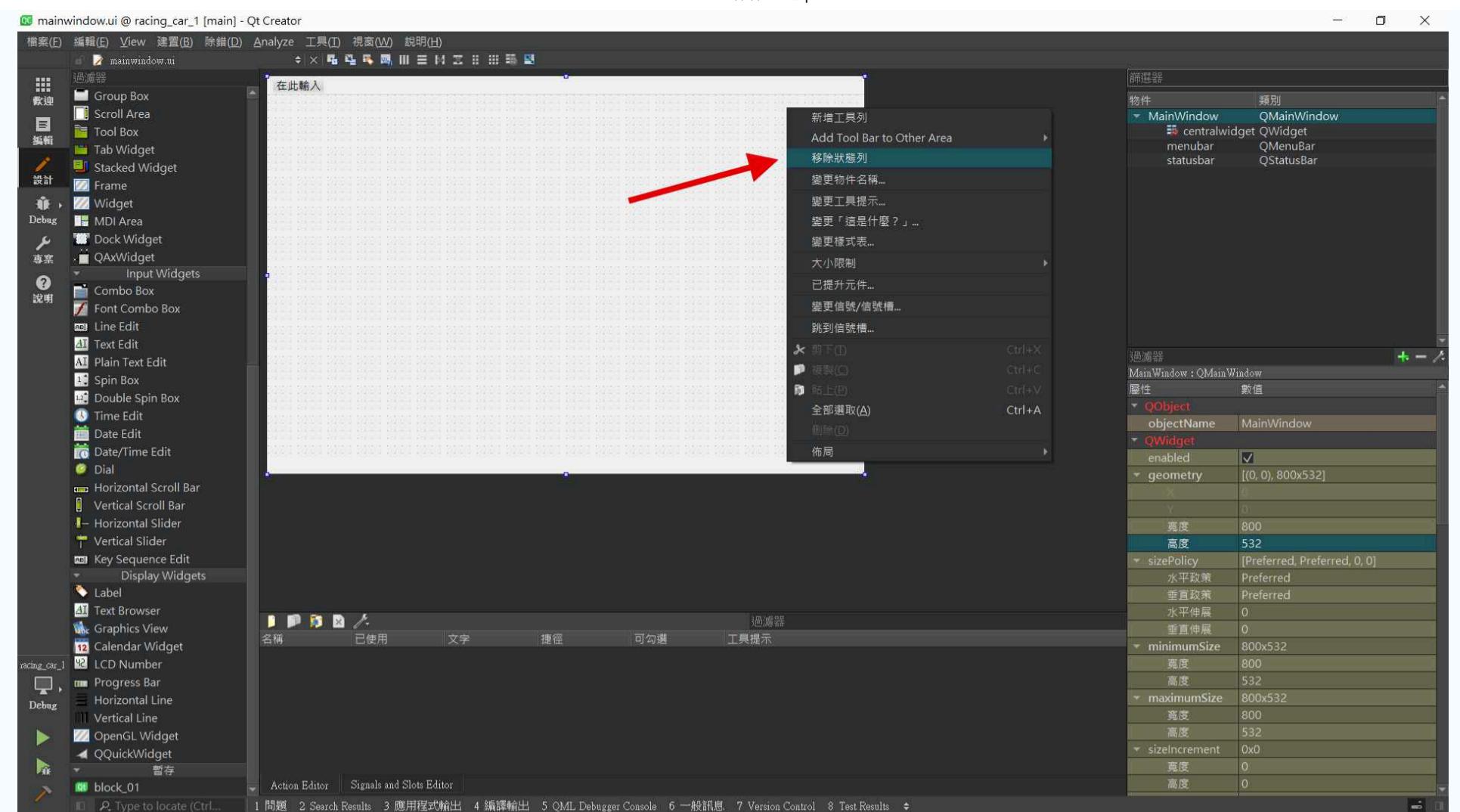


9. 點選【MainWindow】調整視窗大小，因為才關係，geometry、minimumSize、maximumSize 皆設定為寬度 800 和高度 532。

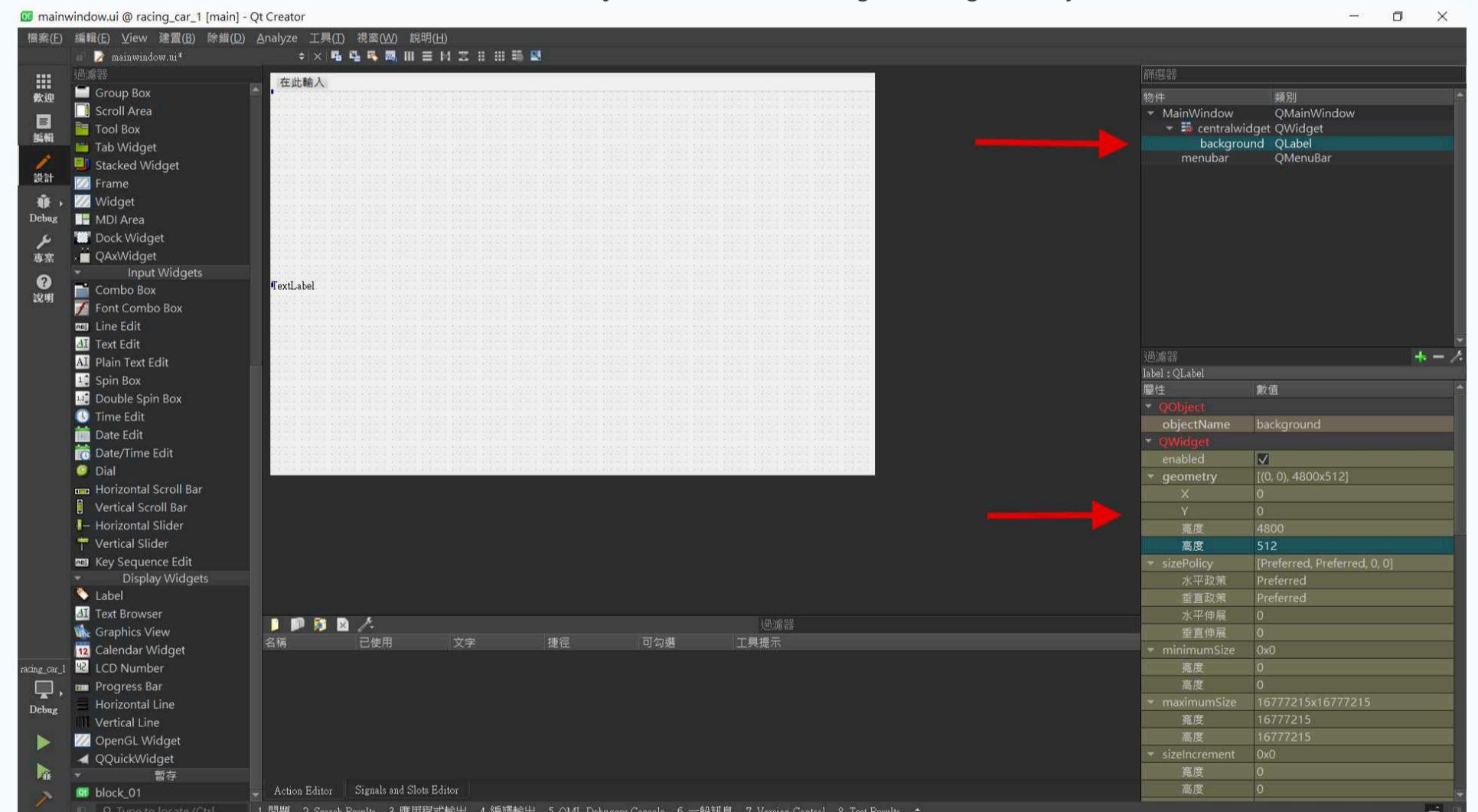


10. 此次專案不須使用狀態列，故將其移除。

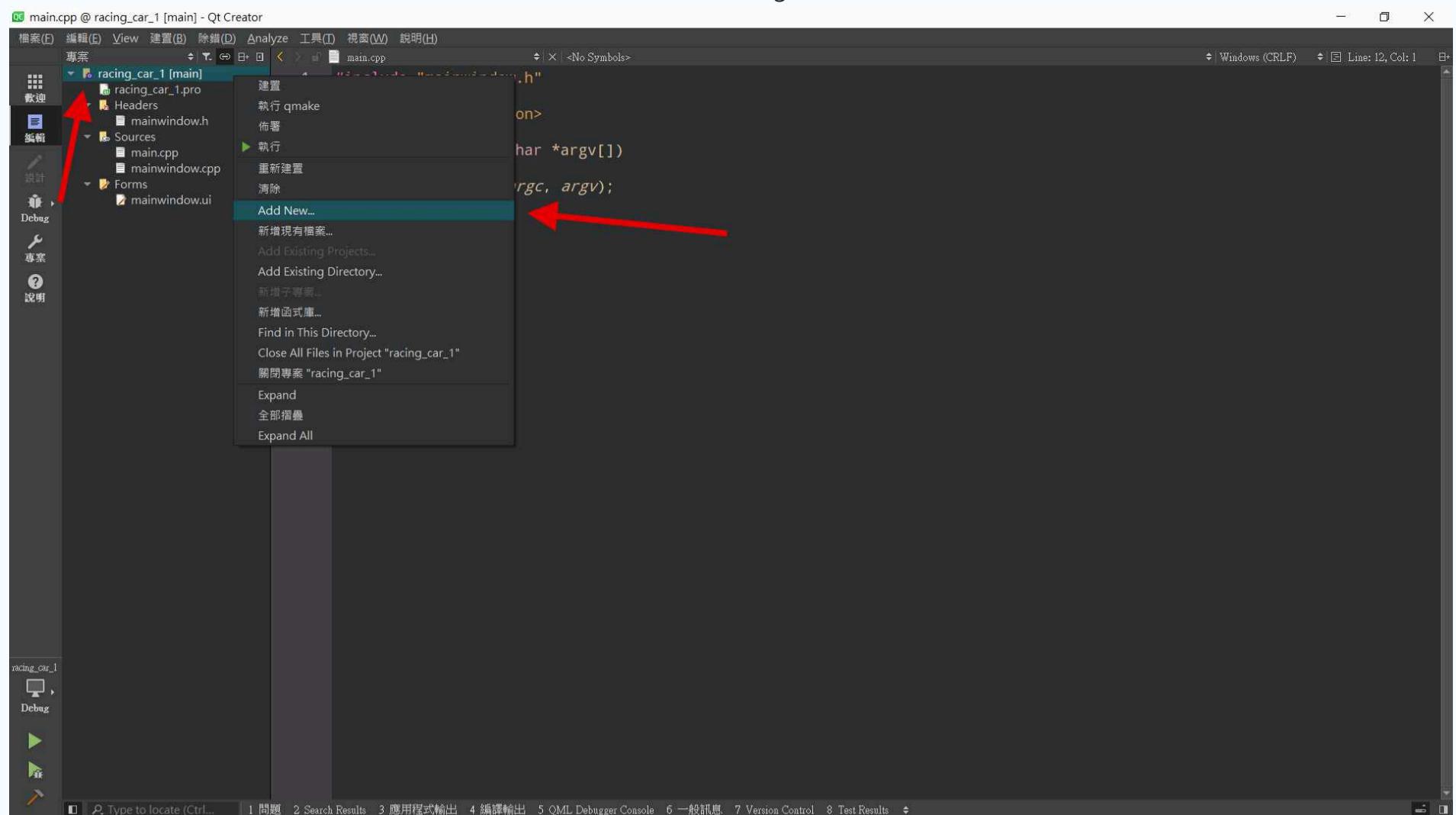
在畫面中點選滑鼠右鍵，點選【移除狀態列】。



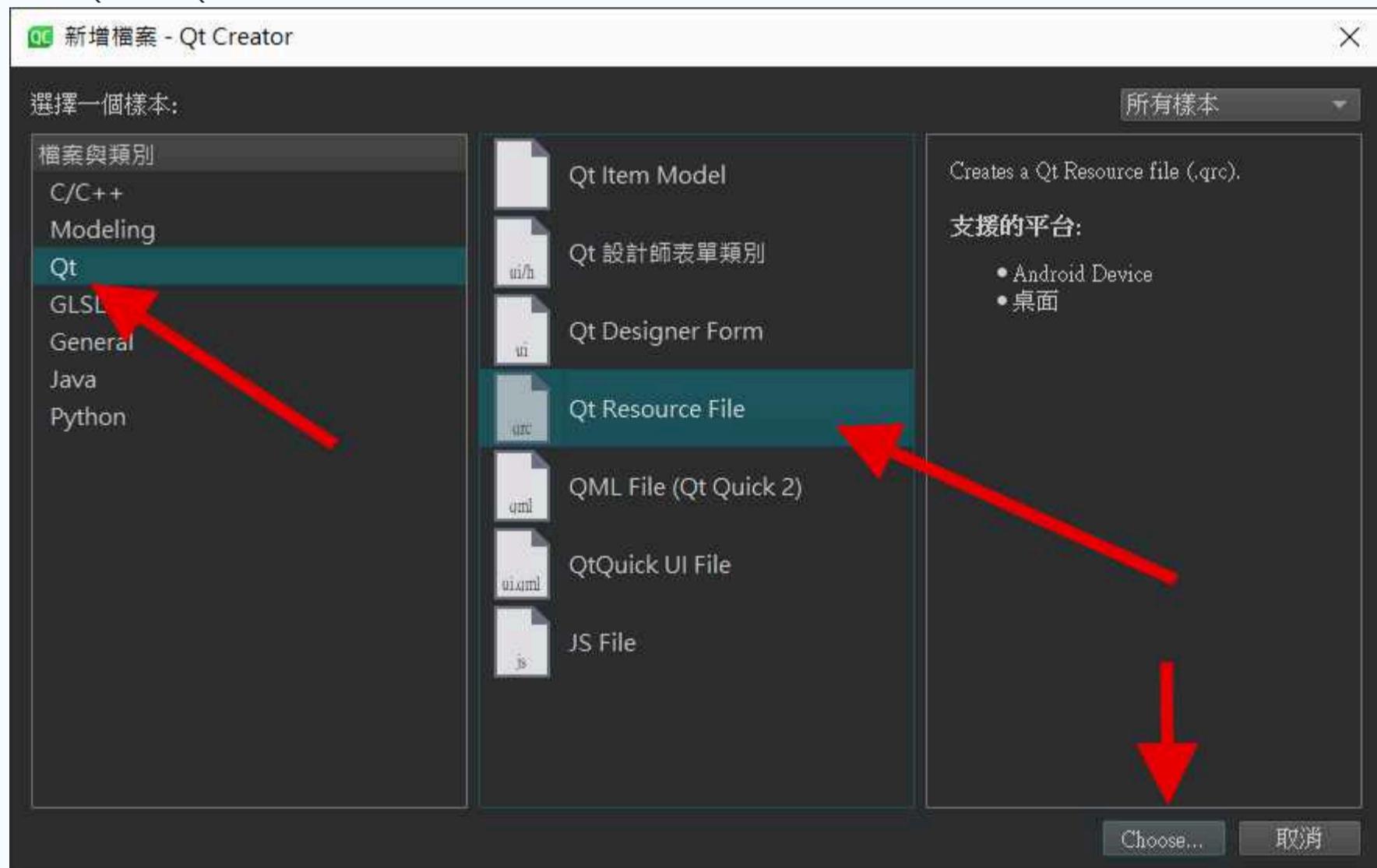
11. 接下來，製作遊戲背景，新增一個【Label】，將其 `objectName` 命名為 `background`，`geometry` 的寬度為 4800，高度為 512。



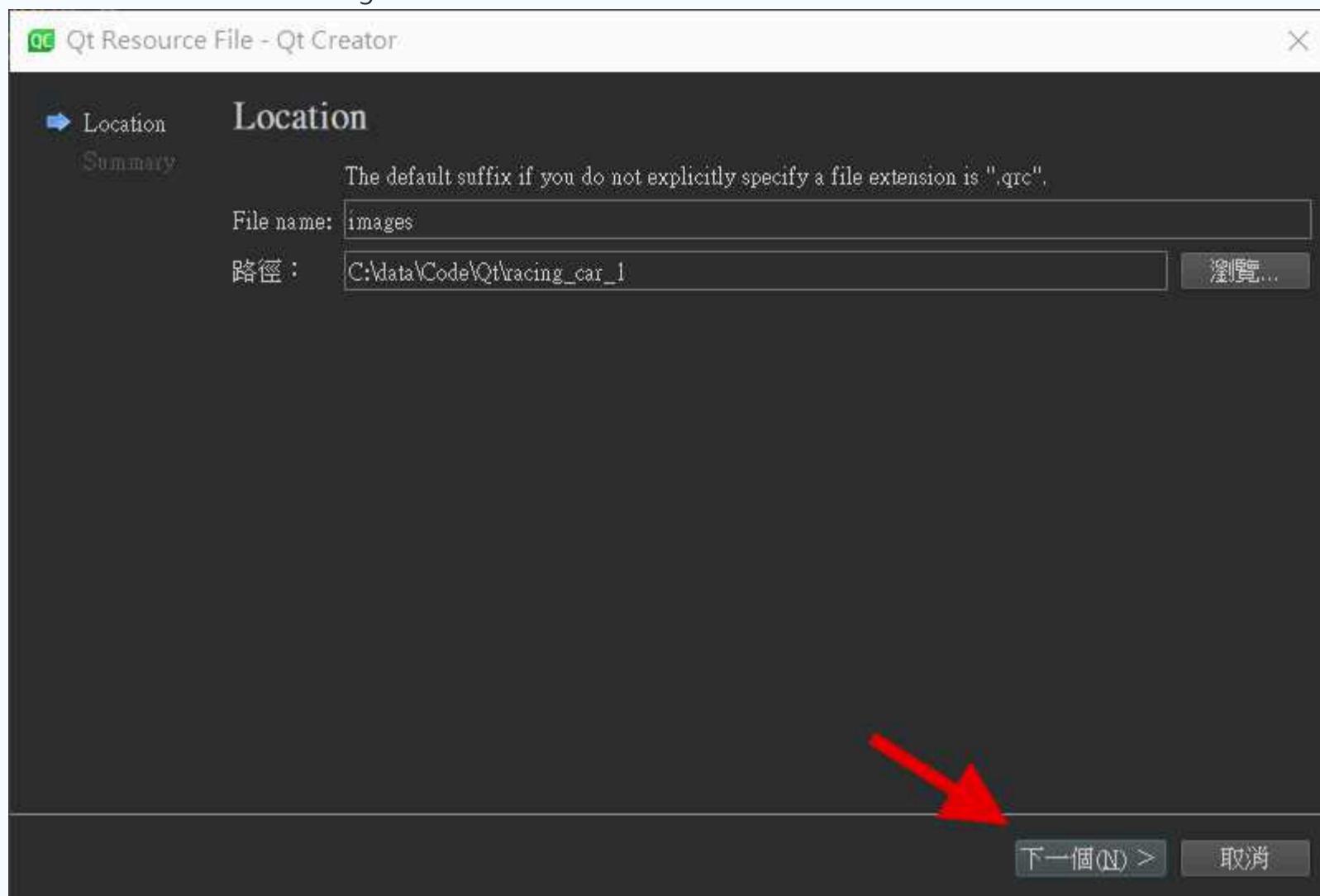
12. 接下來，須建立一個資源檔，將所有的圖片放入，點選專案名稱【racing_car_1】，點選【Add New...】。



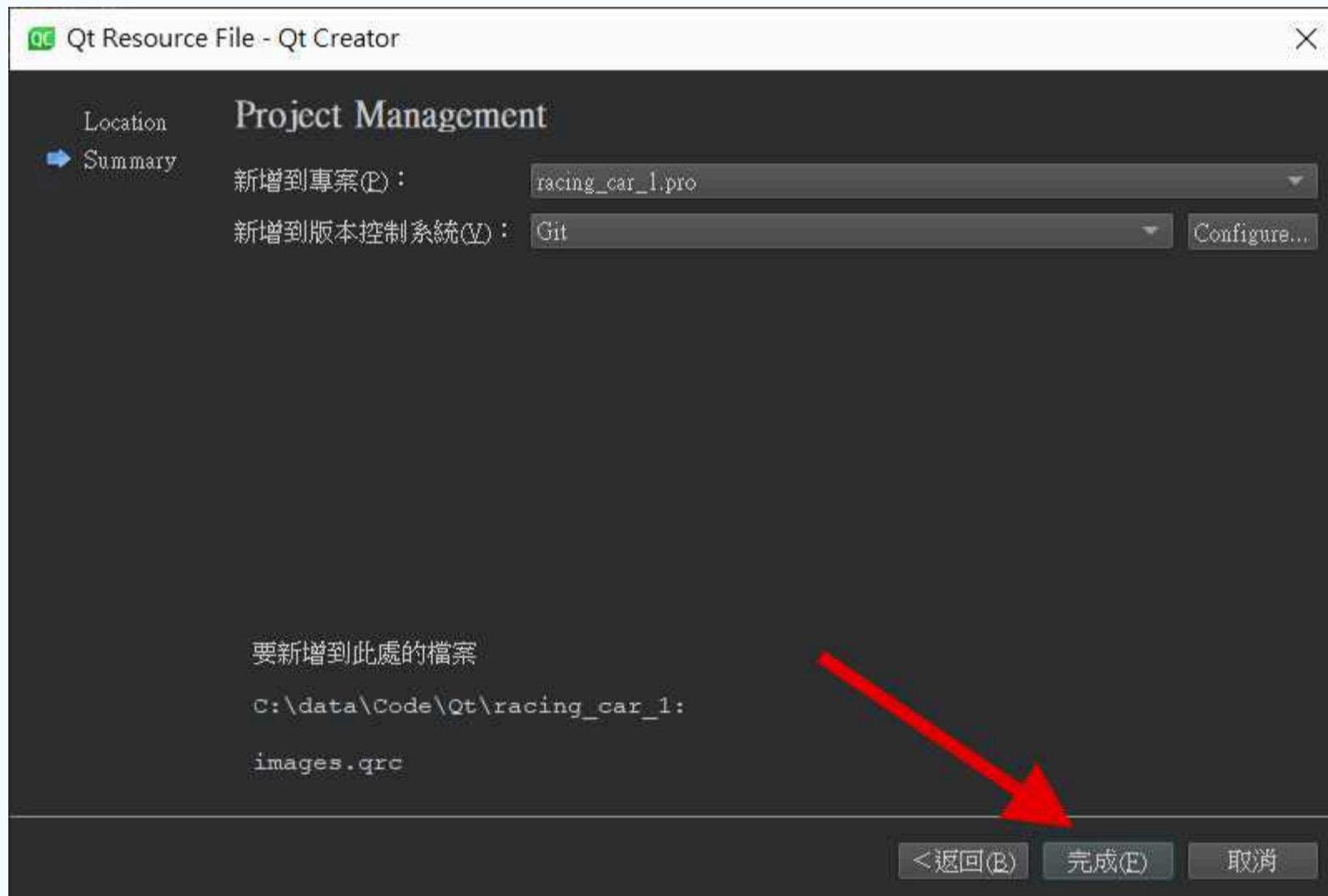
13. 選擇【Qt】>【Qt Resource File】。



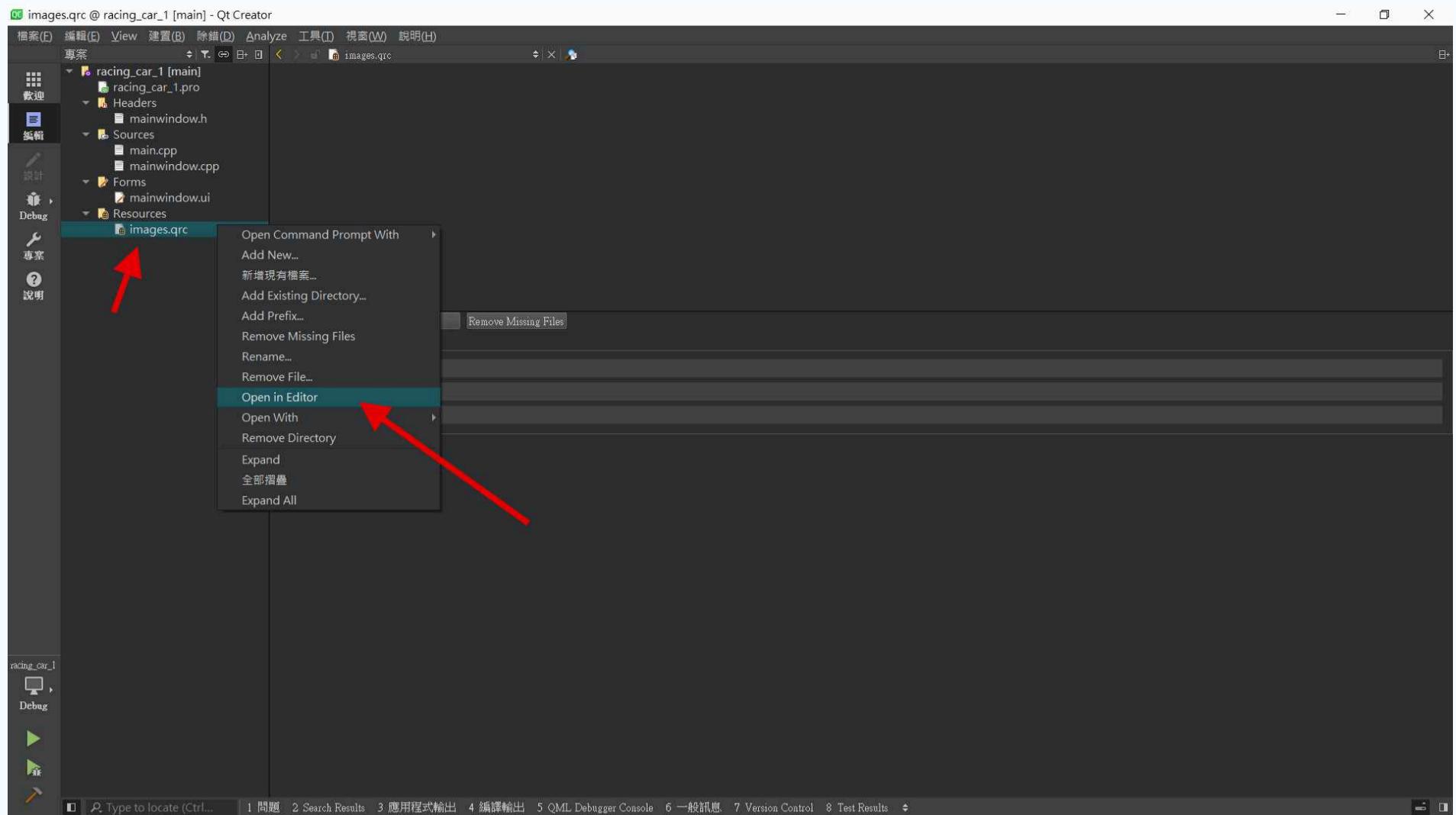
14. 將【File name】命名為【images】。



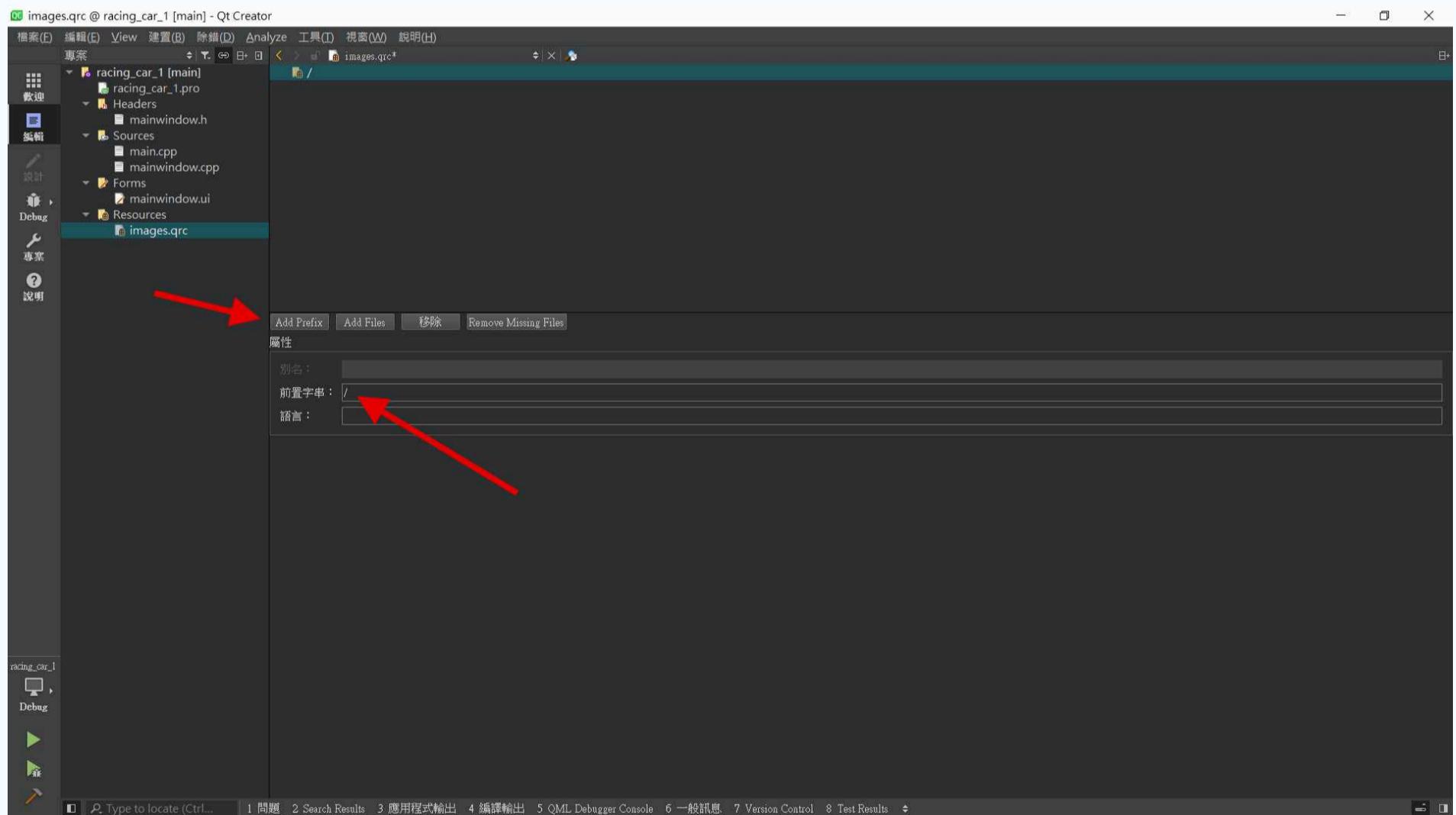
15. 選擇版本控制系統【在此選擇 Git】。



16. 若要編輯資源檔，需在資源檔上點選滑鼠右鍵，選擇【Open in Editor】。



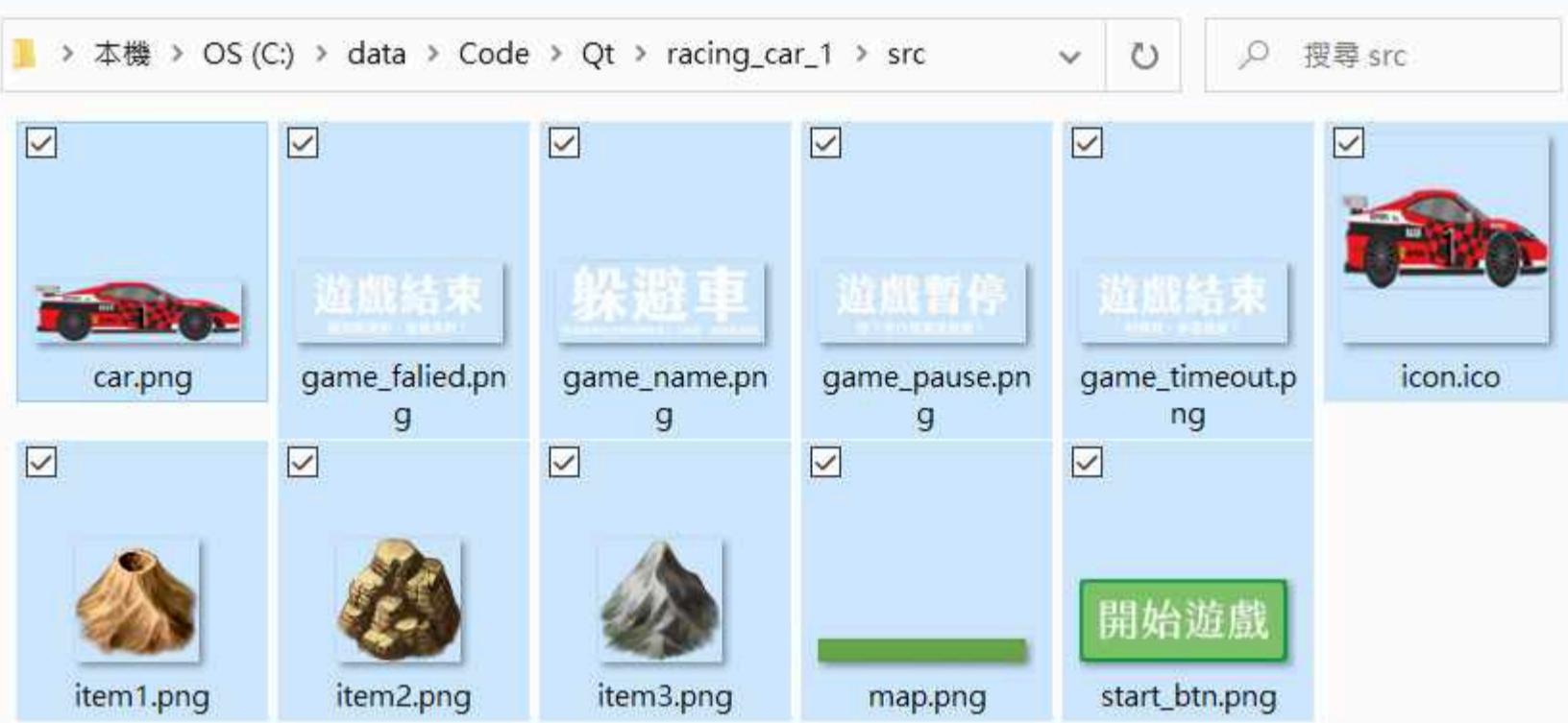
17. 在資源檔編輯器中點選【Add Prefix】，輸入前字字串【/】。



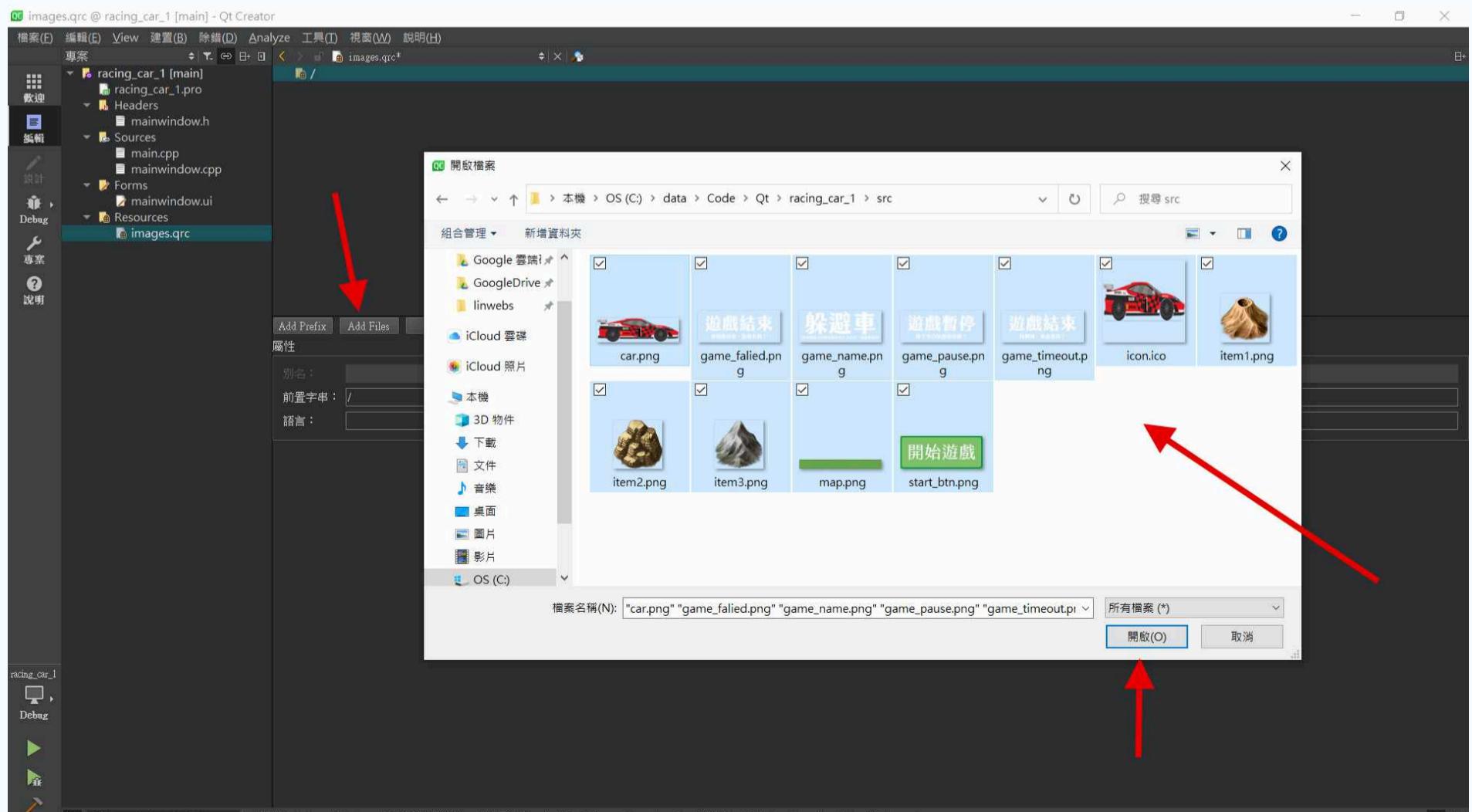
18. 將下載來的素材 [ncyu_study_racing_car_src.zip](#) 複製到專案的資料夾中，並將其解壓縮，可看到 src 資料夾。

| 名稱 | 修改日期 | 類型 | 大小 |
|---|--------------------|----------------------|--------|
| <input checked="" type="checkbox"/> src | 2021/5/24 上午 01:41 | 檔案資料夾 | |
| <input type="checkbox"/> .gitignore | 2021/5/25 上午 09:54 | GITIGNORE 檔案 | 1 KB |
| <input type="checkbox"/> images.qrc | 2021/5/25 上午 10:05 | QRC 檔案 | 1 KB |
| <input type="checkbox"/> images.qrc.autosave | 2021/5/25 上午 10:08 | AUTOSAVE 檔案 | 1 KB |
| <input type="checkbox"/> main.cpp | 2021/5/25 上午 09:54 | CPP 檔案 | 1 KB |
| <input type="checkbox"/> mainwindow.cpp | 2021/5/25 上午 09:54 | CPP 檔案 | 1 KB |
| <input type="checkbox"/> mainwindow.h | 2021/5/25 上午 09:54 | H 檔案 | 1 KB |
| <input type="checkbox"/> mainwindow.ui | 2021/5/25 上午 09:58 | Qt UI file | 1 KB |
| <input checked="" type="checkbox"/> ncyu_study_racing_car_src.zip | 2021/5/25 上午 10:08 | WinRAR ZIP 壓縮檔 | 147 KB |
| <input type="checkbox"/> racing_car_1.pro | 2021/5/25 上午 10:05 | Qt Project file | 1 KB |
| <input type="checkbox"/> racing_car_1.pro.user | 2021/5/25 上午 09:54 | Per-User Project ... | 37 KB |

19. 確認 src 資料夾內有這些圖片檔案。

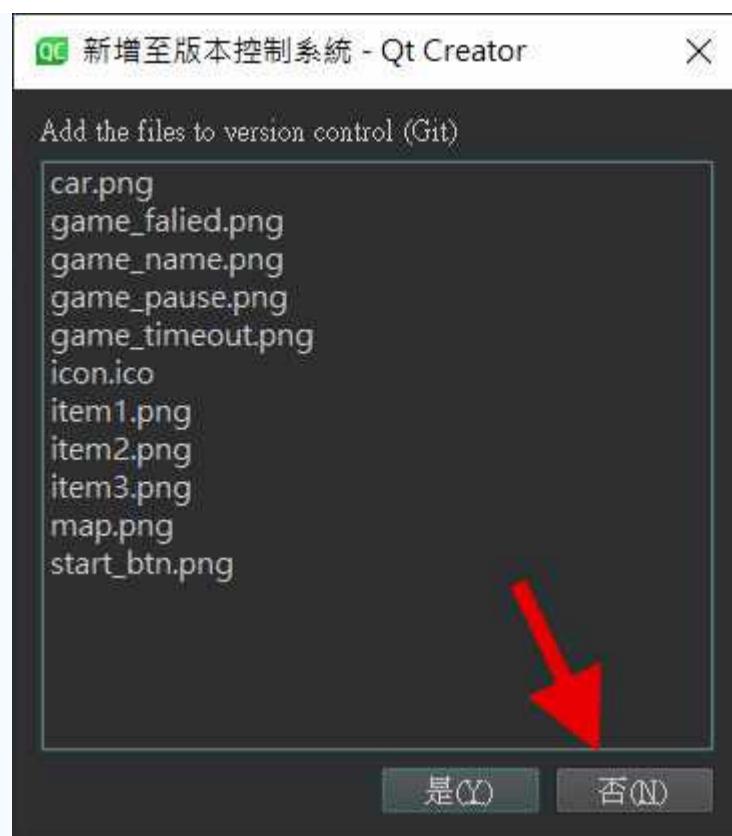


20. 回到 Qt Creator，點選資源檔的【Add Files】，選擇所有的圖片後按【開啟】。

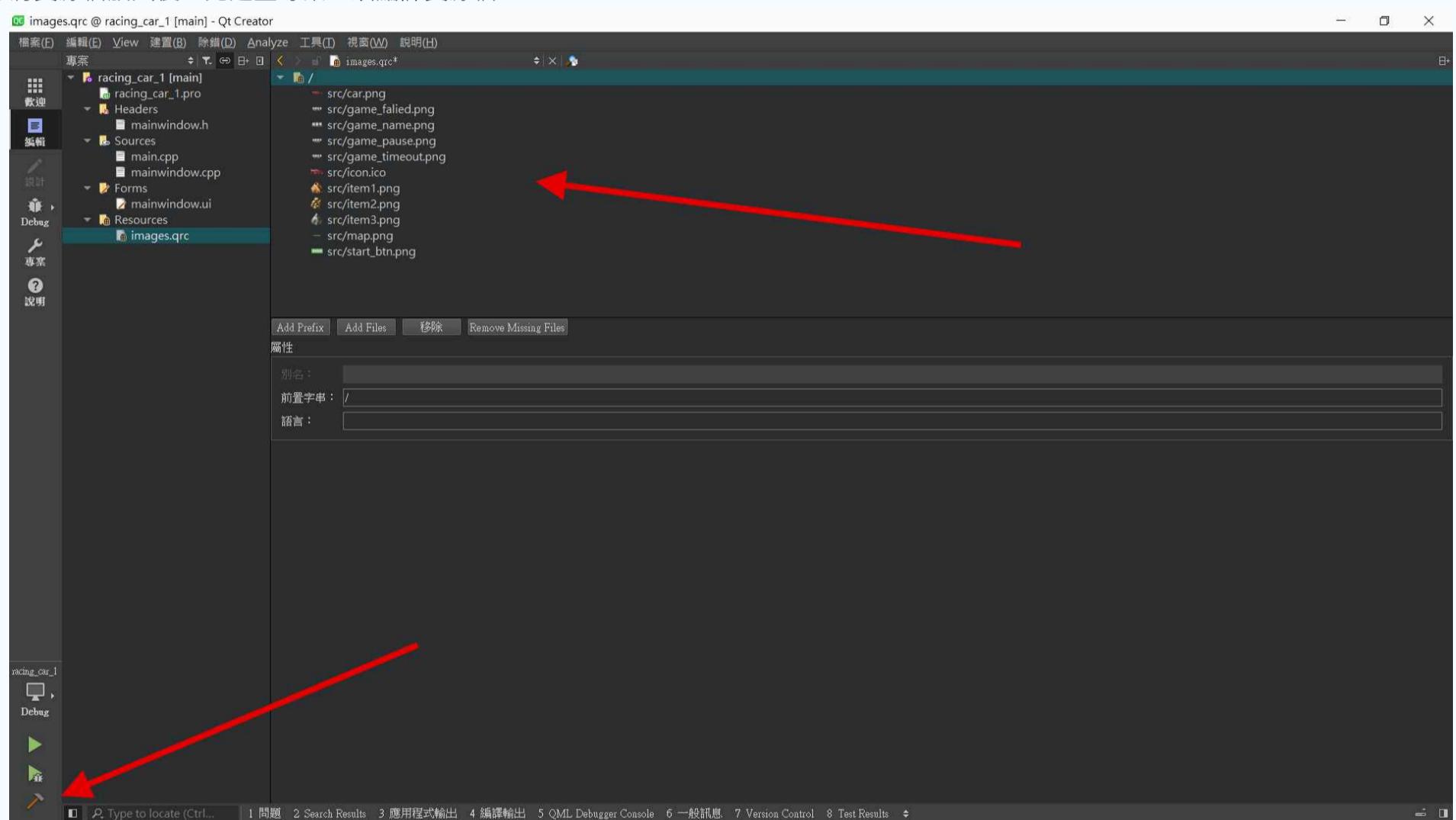


21. 是否將其新增到版本控制。

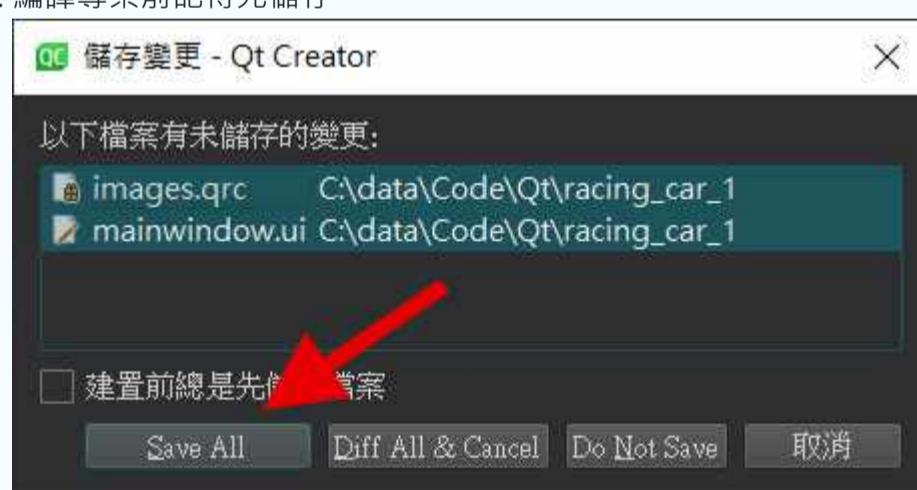
由於此版本的 Qt Creator 有問題，無法正常執行，故先選擇【否】，待稍後手動加入版本控制。



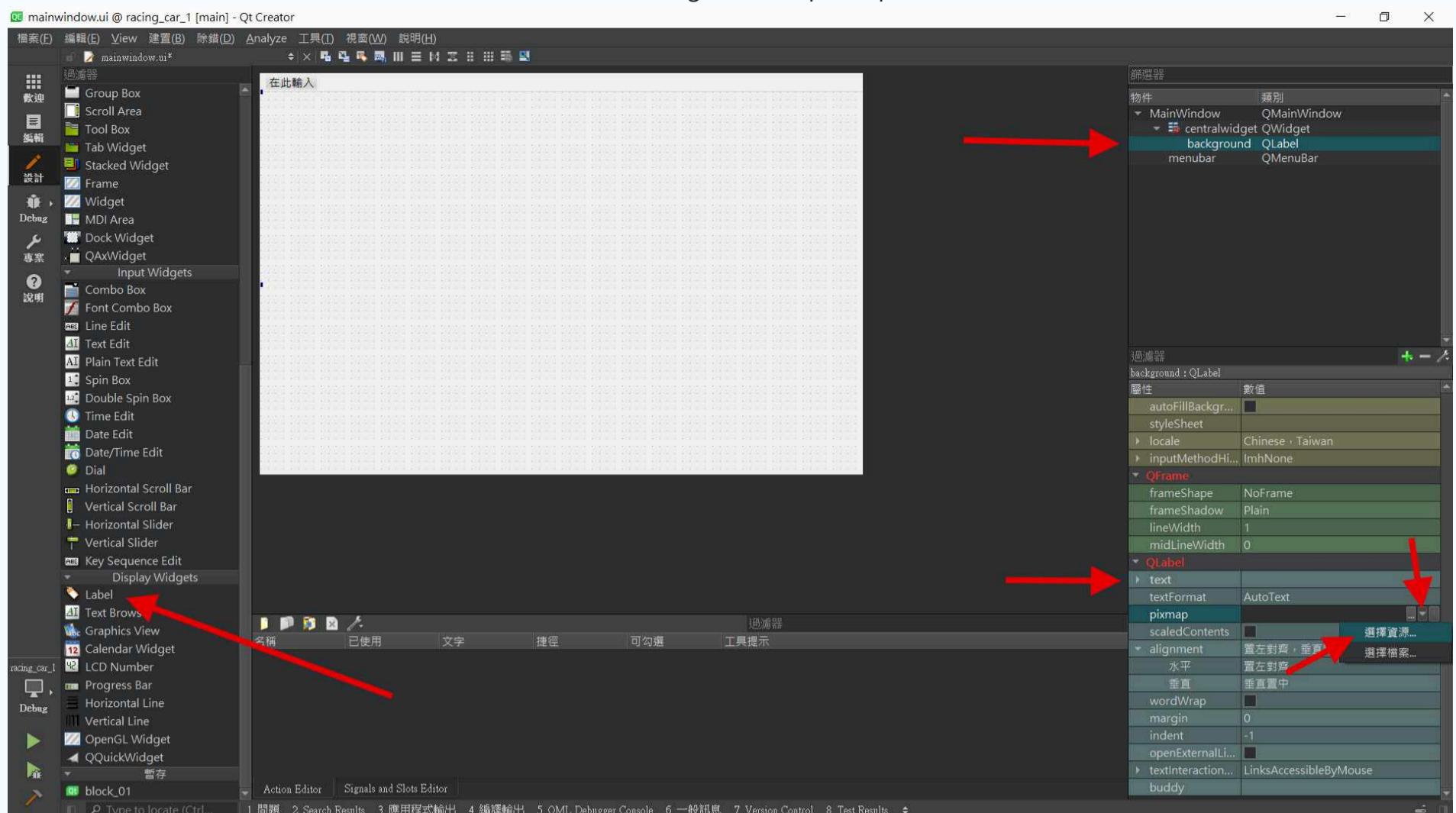
22. 將資源檔放入後，先建置專案，以編譯資源檔。



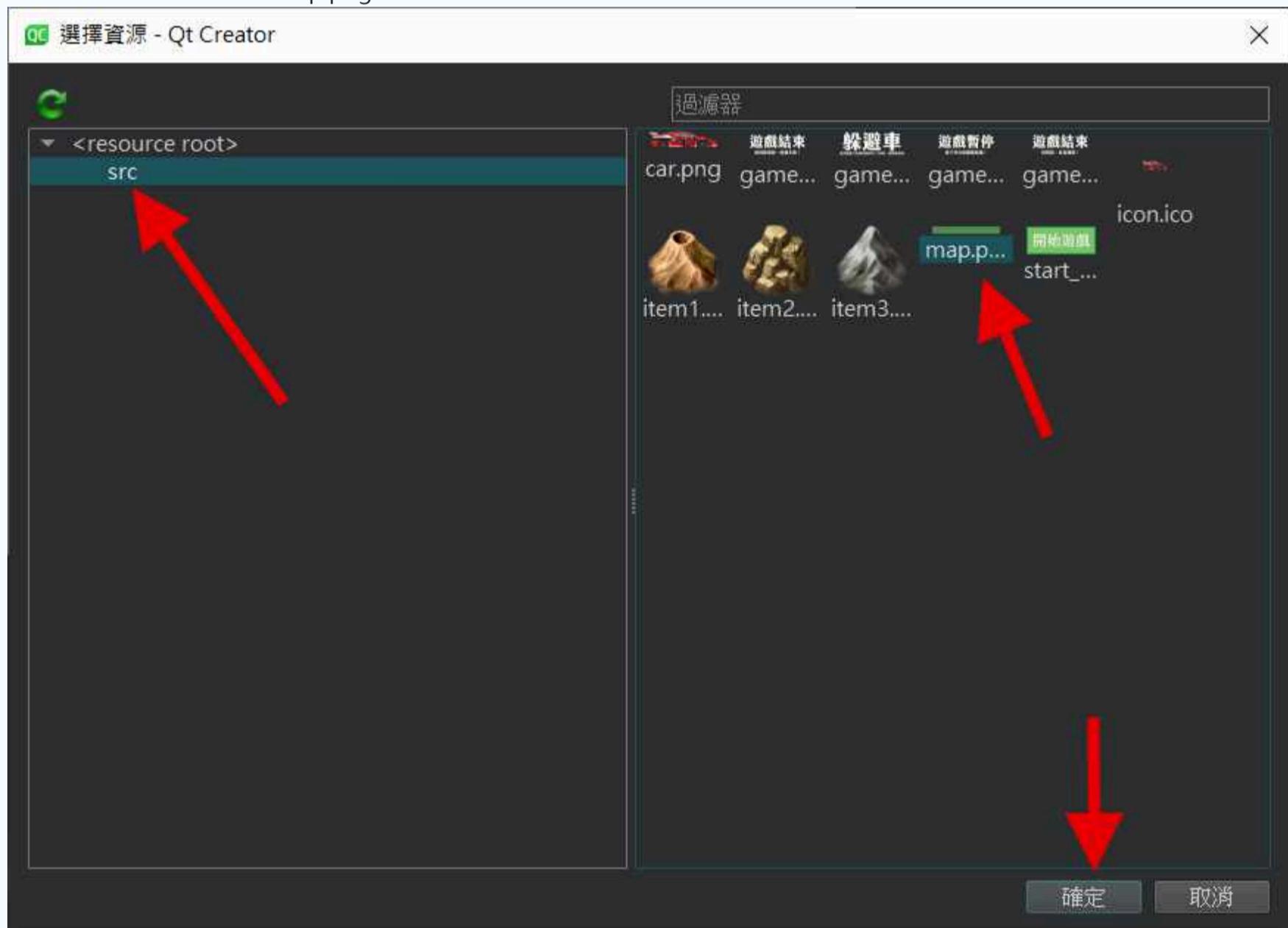
23. 編譯專案前記得先儲存。



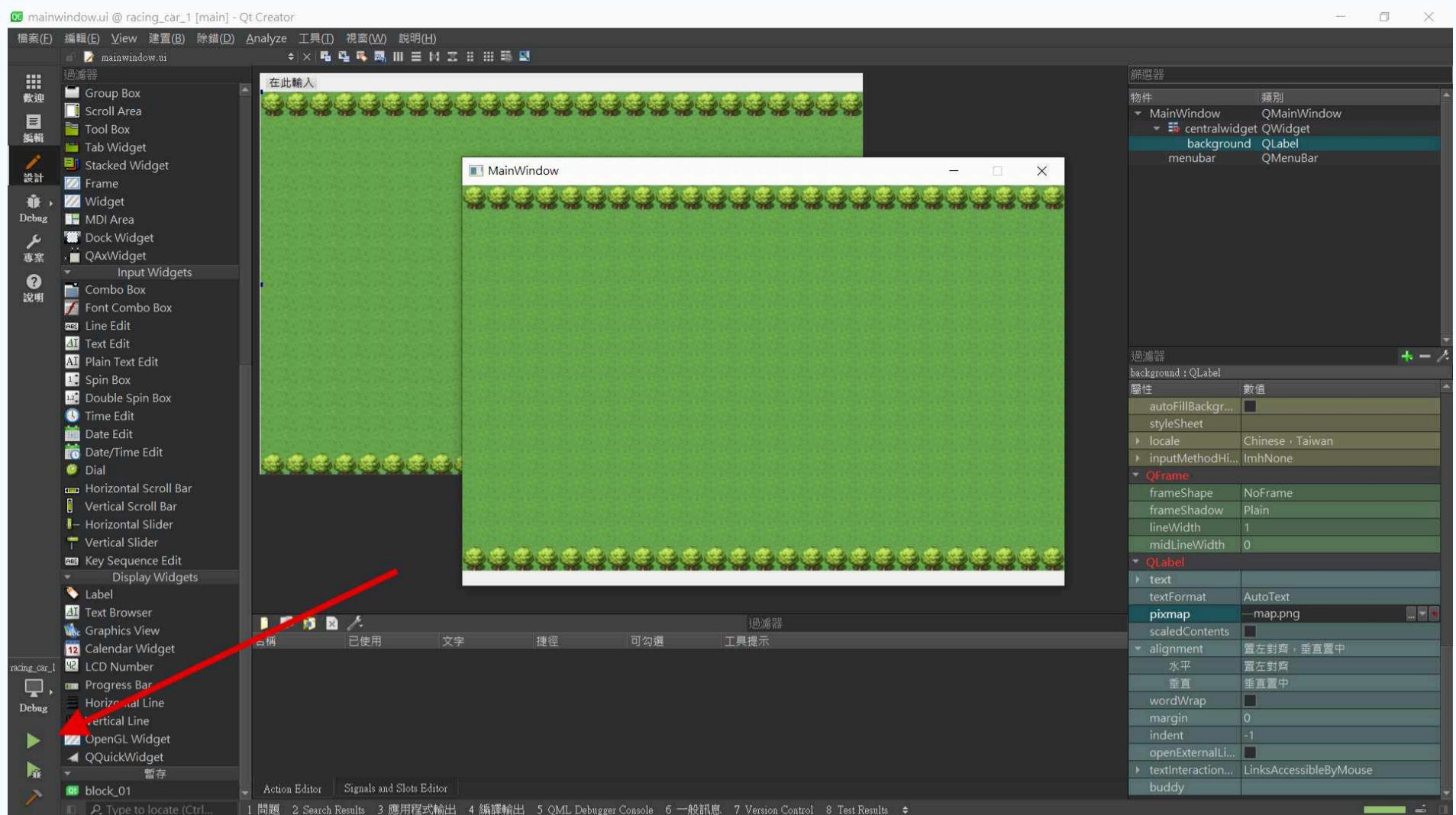
24. 再次回到 ui 編輯頁面，將背景套上剛才匯入的圖片，點選 background 的 pixmap 後方的箭頭【▼】，點選【選擇資源...】。



25. 點選【src】資料夾中的【map.png】。



26. 編譯並執行專案，可看到背景已成功顯示。

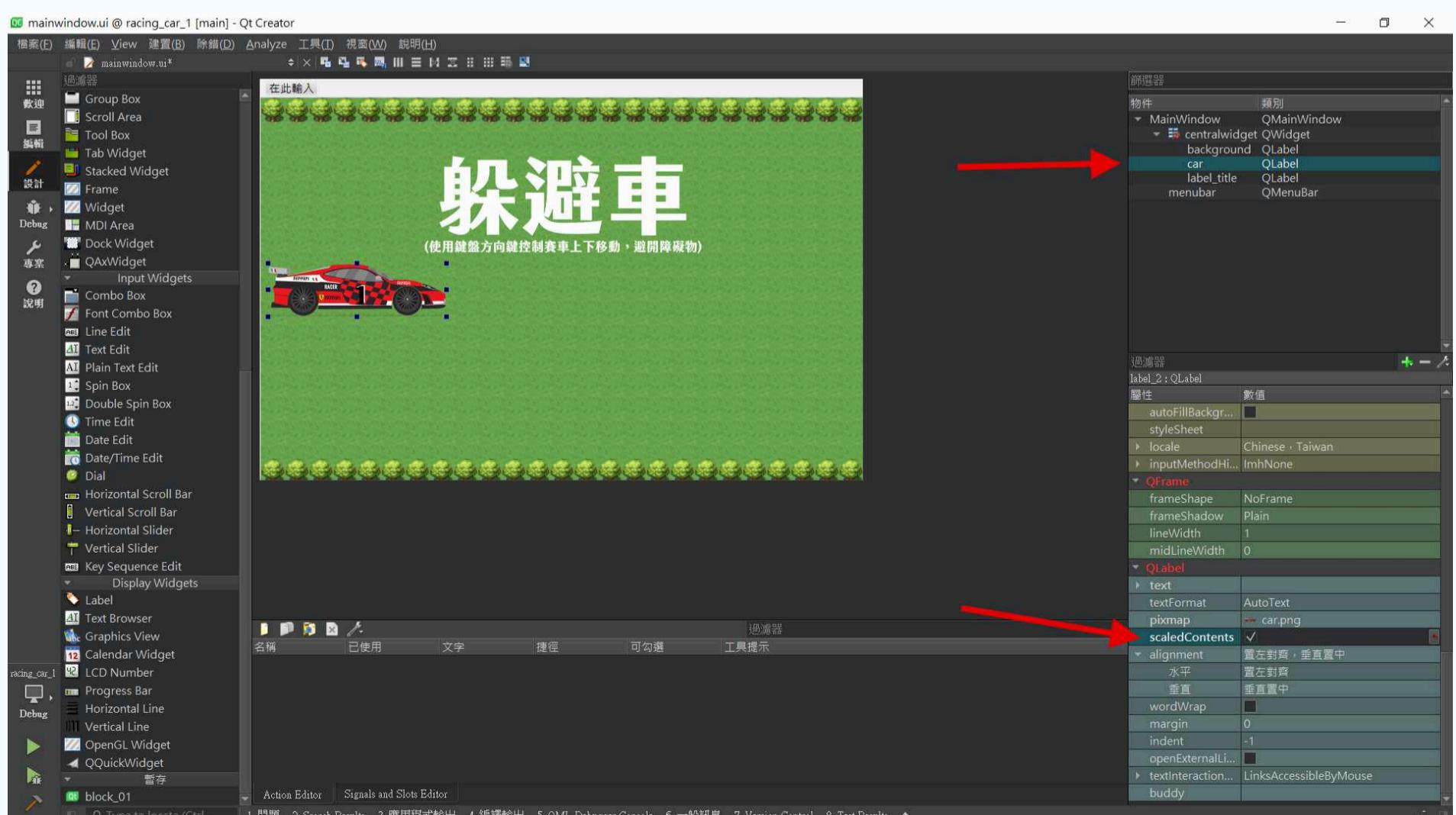


27. 接下來以同樣的方式加入兩個 Label，分別為當作畫面標題及車子。

並將車子的 scaledContents 打勾，將其圖片大小符合框架大小。

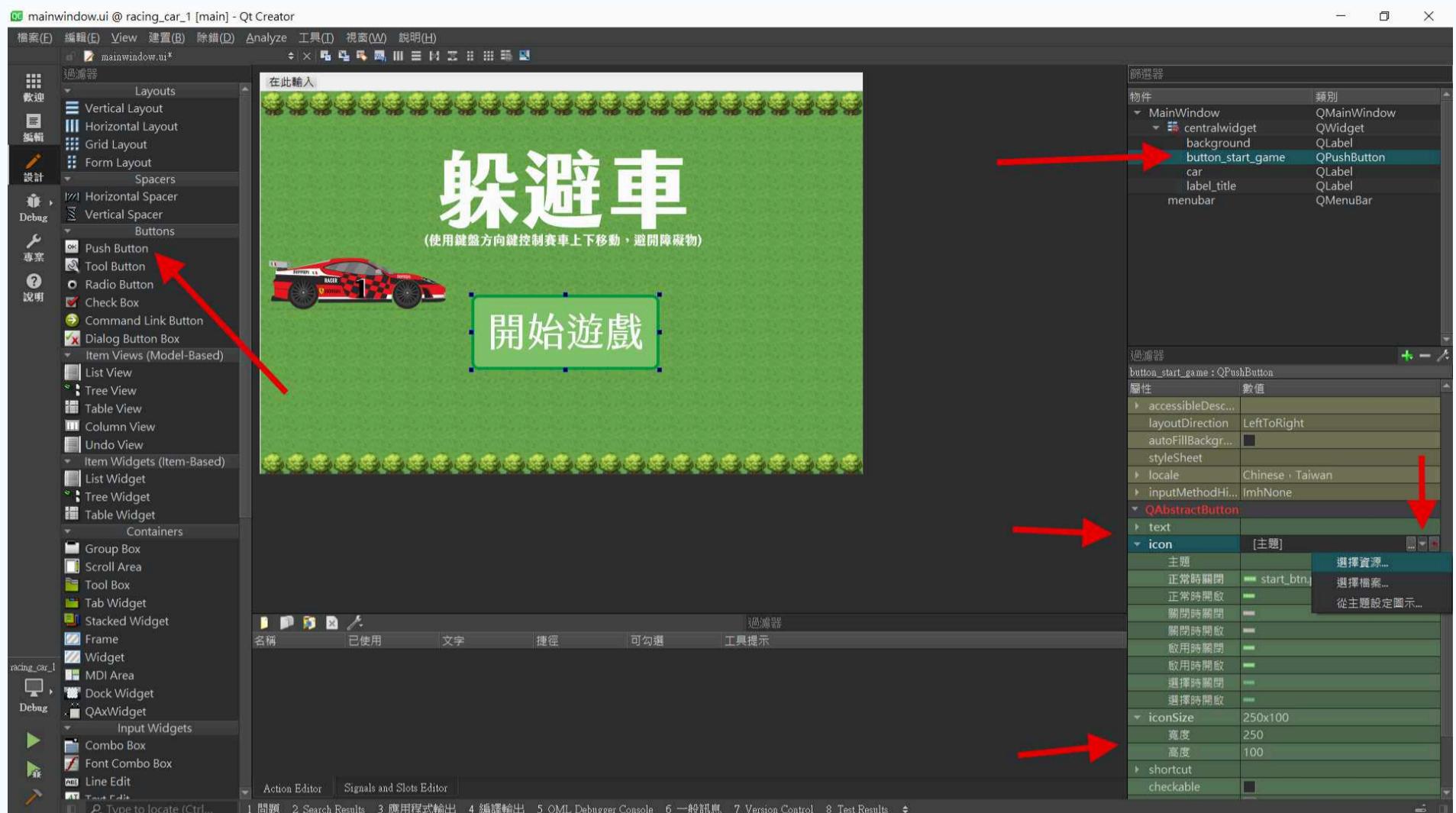
```
label_title = {
    objectName: label_title,
    geometry: [x: 210, y: 70, 寬度: 384, 高度: 150],
    pixmap: game_name.png
}
```

```
car = {
    objectName: car,
    geometry: [x: 10, y: 220, 寬度: 237, 高度: 71],
    scaledContents: true,
    pixmap: car.png
}
```



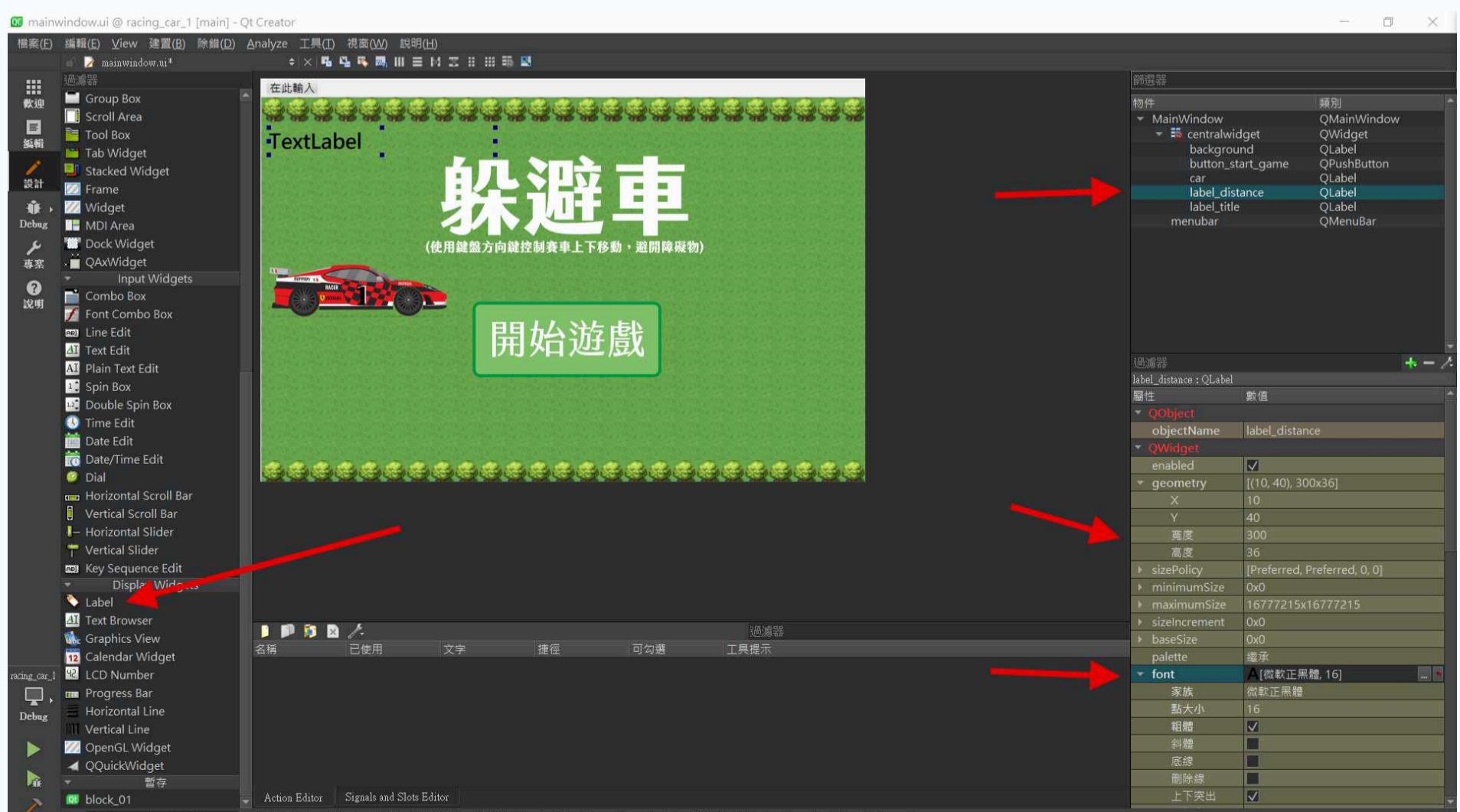
28. 再加入一個 PushButton，點選【icon】設定圖片背景，並將 iconSize 改為按鈕的寬度和高度。

```
button_start_game = {
    objectName: button_start_game,
    geometry: [x: 280, y: 270, 寬度: 250, 高度: 100],
    text: ,
    icon: start_btn.png,
    iconSize: [寬度: 250, 高度: 100]
}
```



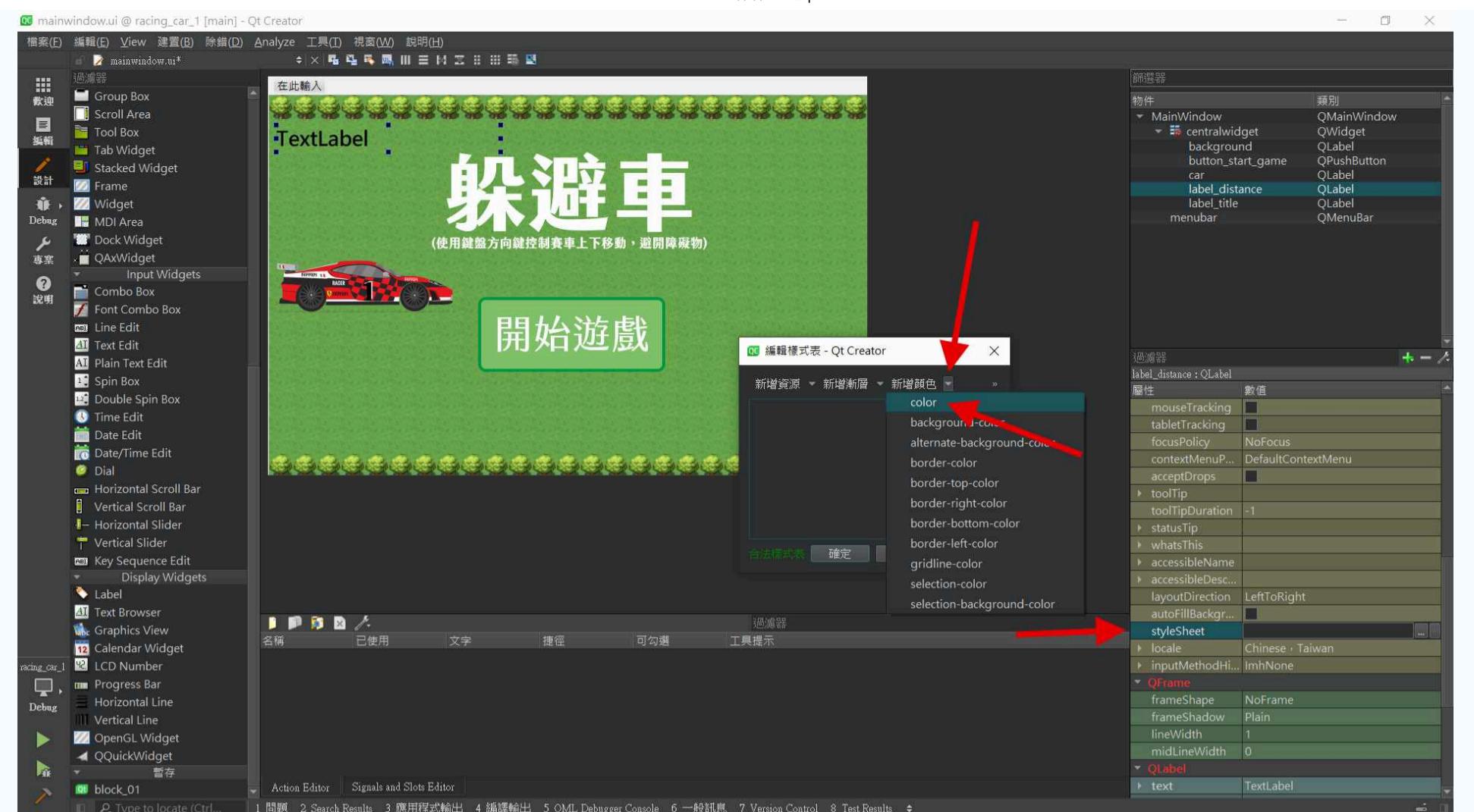
29. 再加入一個 Label 顯示車子的行駛距離。

```
label_distance = {
    objectName: label_distance,
    geometry: [x: 10, y: 40, 寬度: 300, 高度: 36],
    text: 行駛距離: 0公尺
}
```



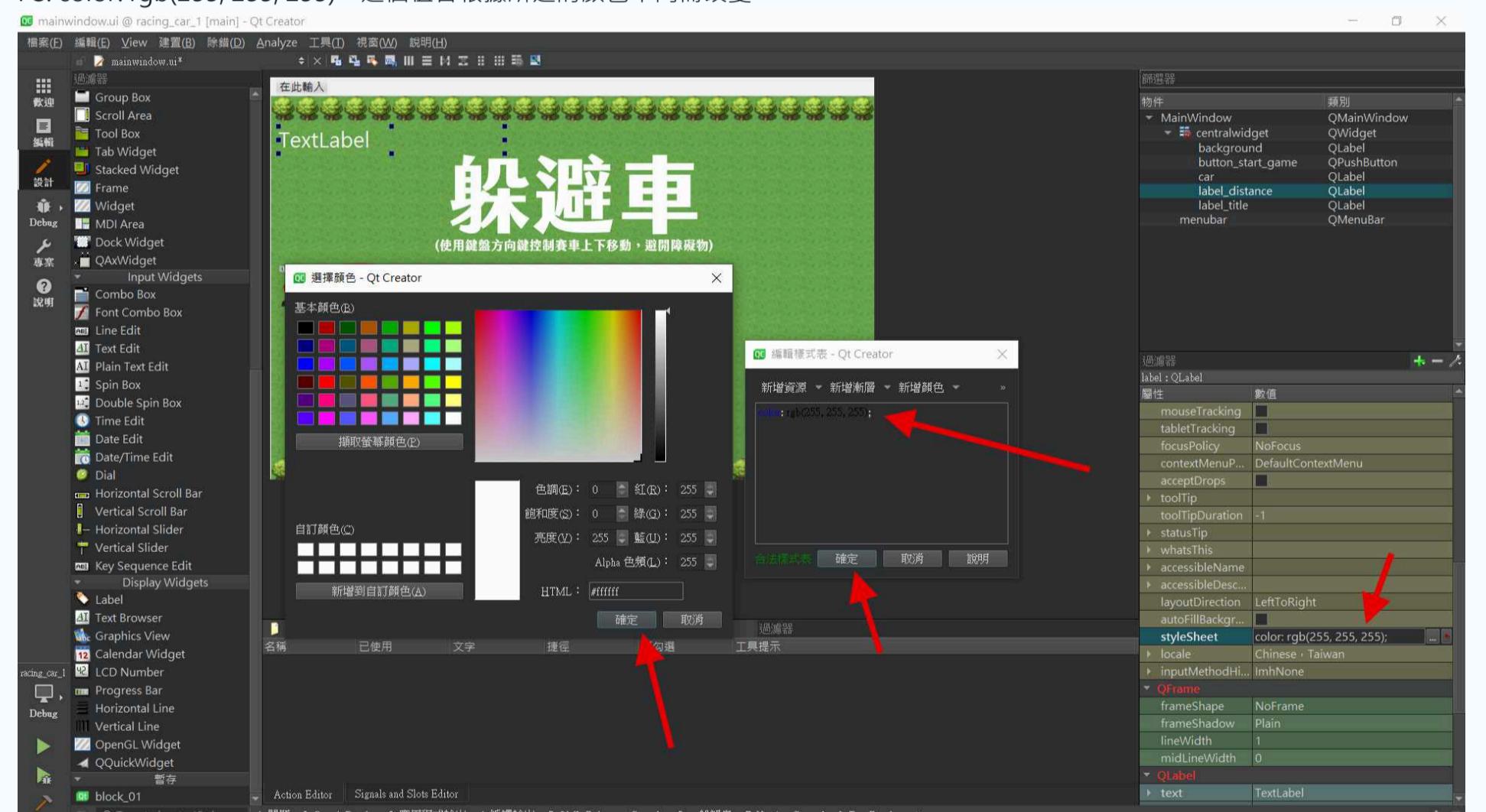
30. 將行駛距離的文字改變顏色。

點選【styleSheet】，點選新增顏色旁的【▼】，點選【color】。

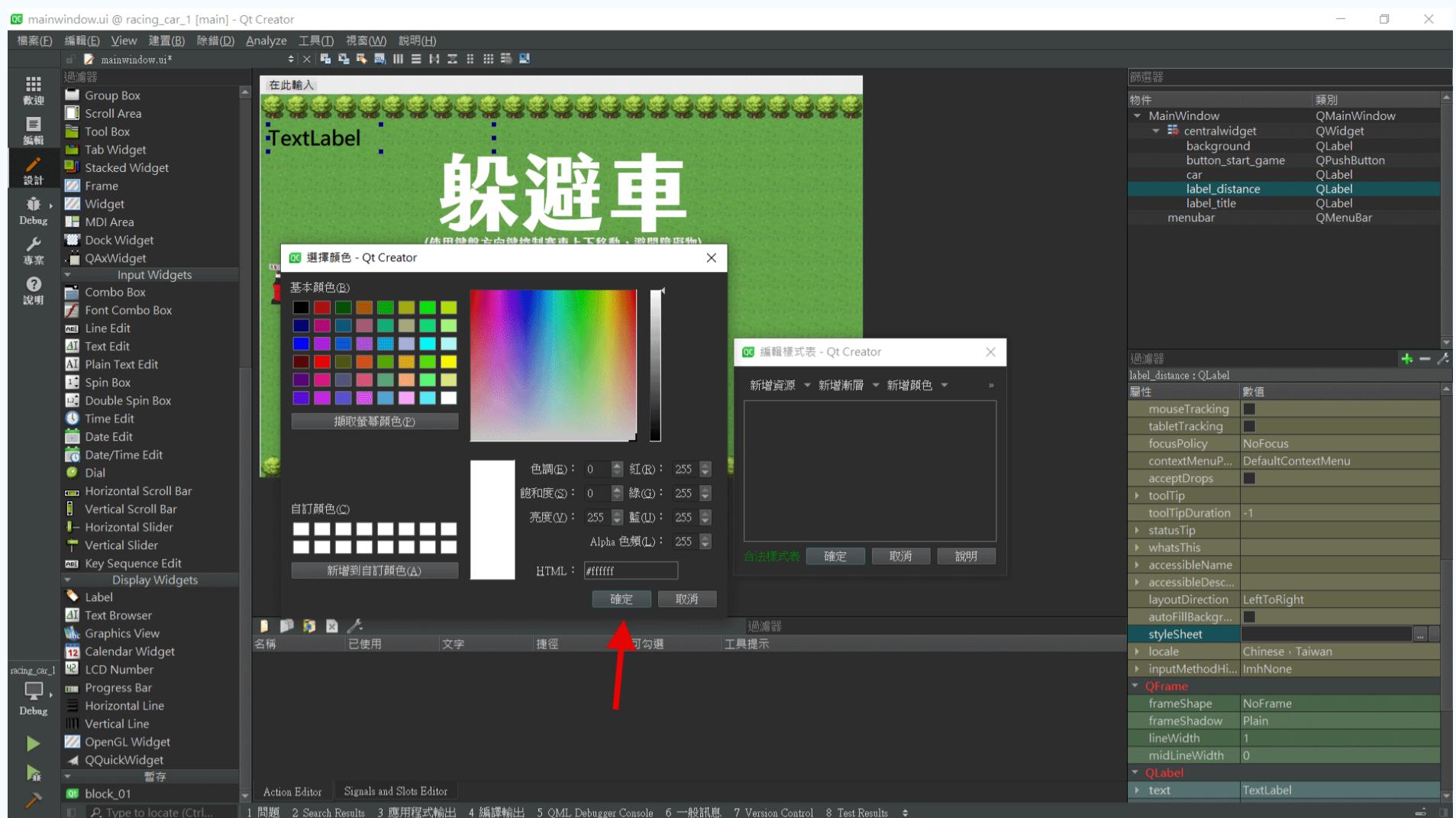


31. 可在此處選擇顏色，選擇完畢，點選確定，確認 `styleSheet` 的值為 `color: rgb(255, 255, 255)`。

PS: `color: rgb(255, 255, 255)`，這個值會根據所選的顏色不同而改變。

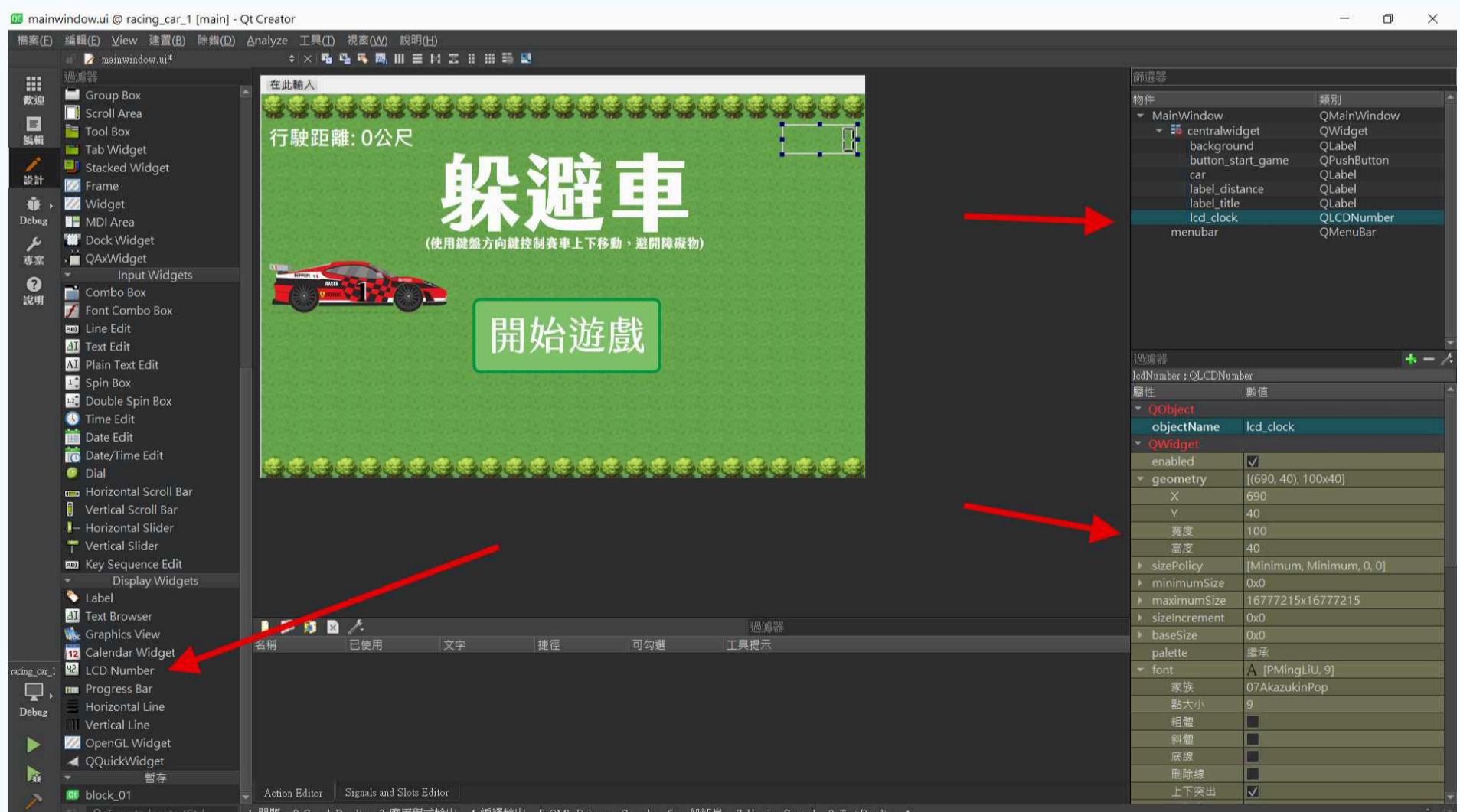


32. 步驟如下。



33. 再新增一個顯示遊戲時間的【LCD Number】。

```
lcd_clock = {
    objectName: lcd_clock,
    geometry: [x: 690, y: 40, 寬度: 100, 高度: 10],
    value: 0.000000,
    intValue: 0
}
```



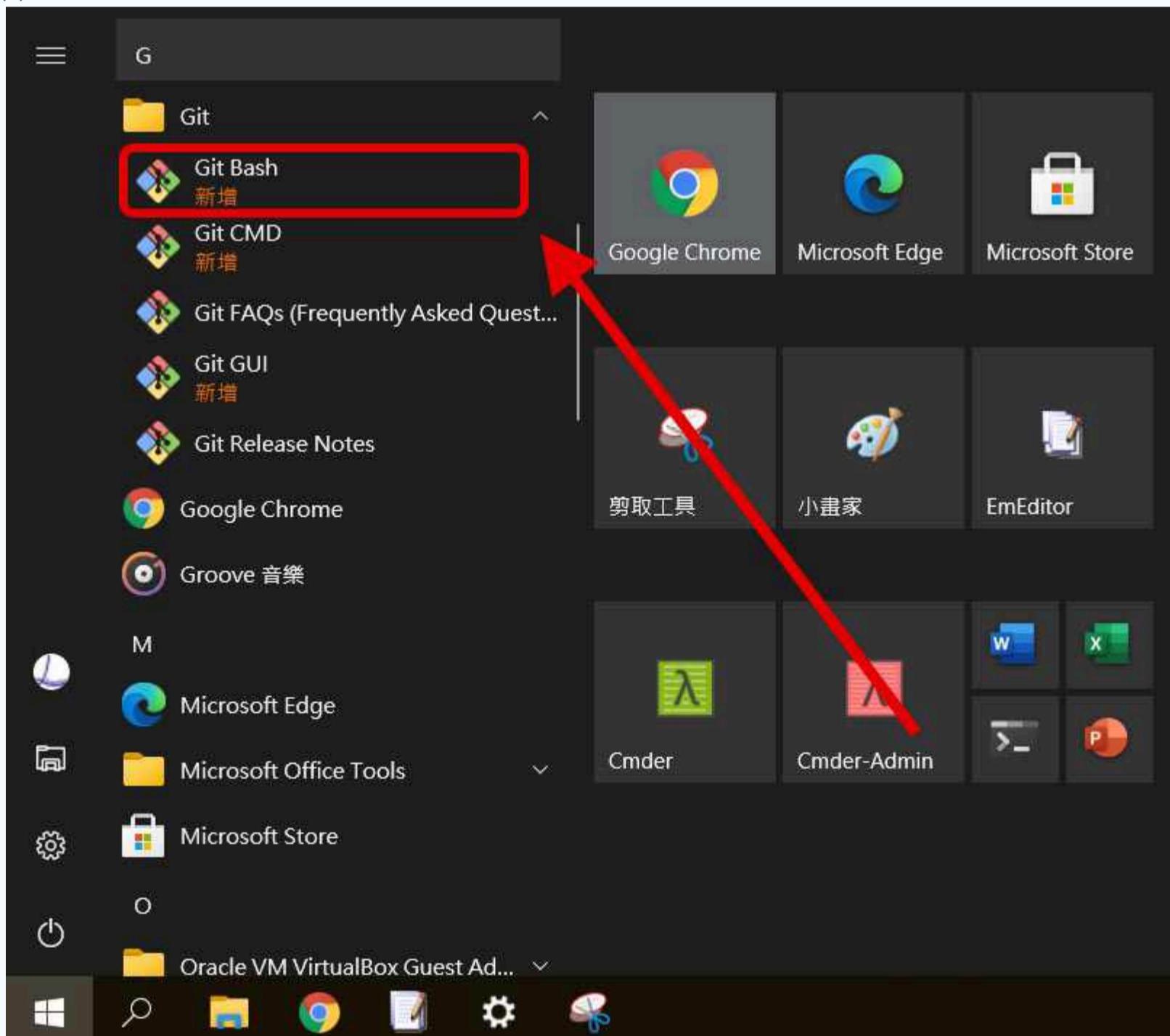
34. 建置並執行專案查看畫面編排結果。



35. 將此專案的程式碼交由 Git 管理。

可使用自己常用的終端機介面執行 Git 指令，如 Git 預設的 Git Bash，或使用 PowerShell(跳至第37步)、或 Cmder (跳至第38步)等。

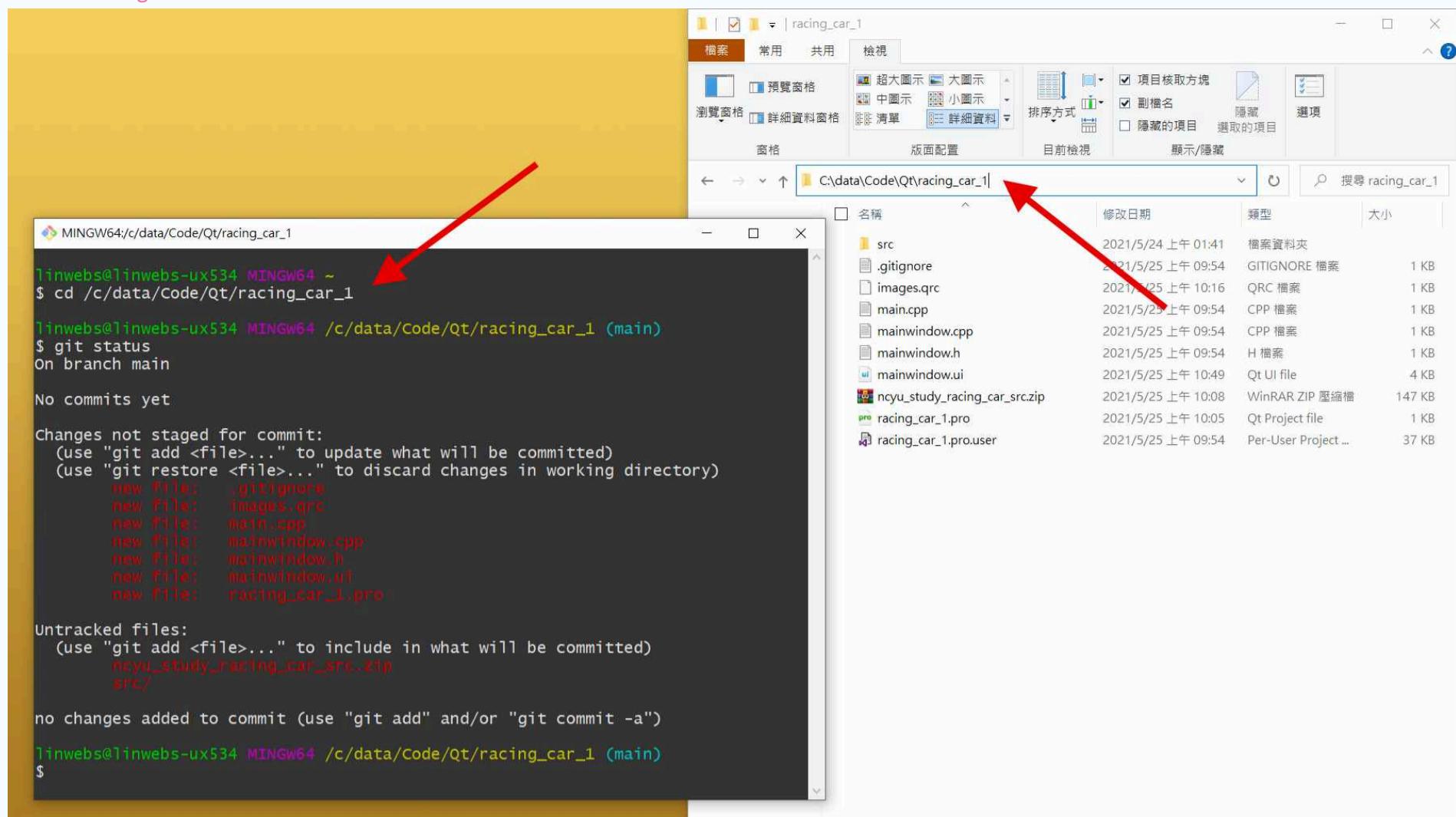
(1) 在此先以 Git Bash 做示範



36. 打開 Git Bash 後，使用 cd 指令前往專案資料夾的路徑。

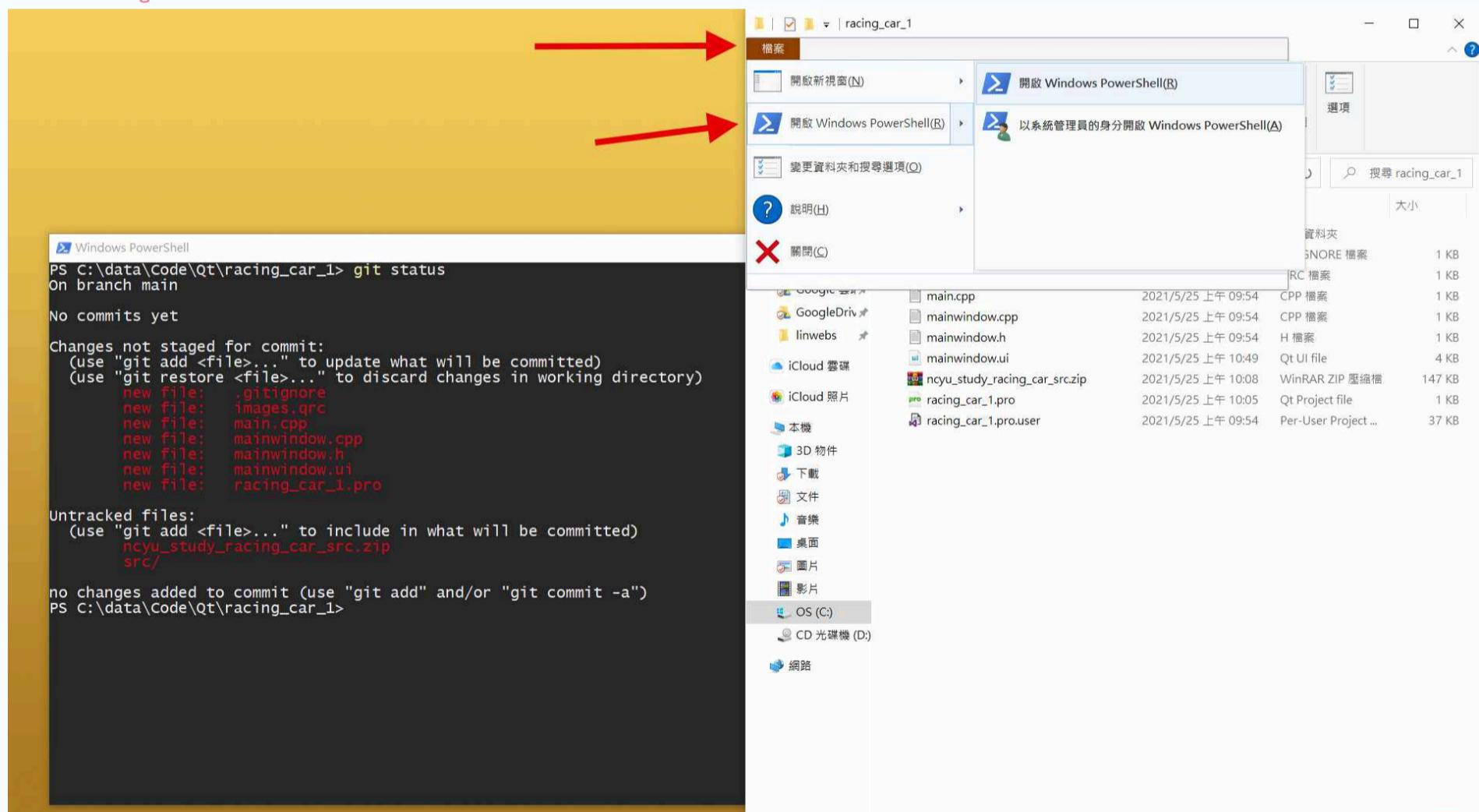
若專案資料夾位於 C:\data\Code\Qt\racing_car_1，則須輸入 cd /c/data/Code/Qt/racing_car_1

接下來使用 `git status` 查看目前程式碼狀態，跳至第 39 步。



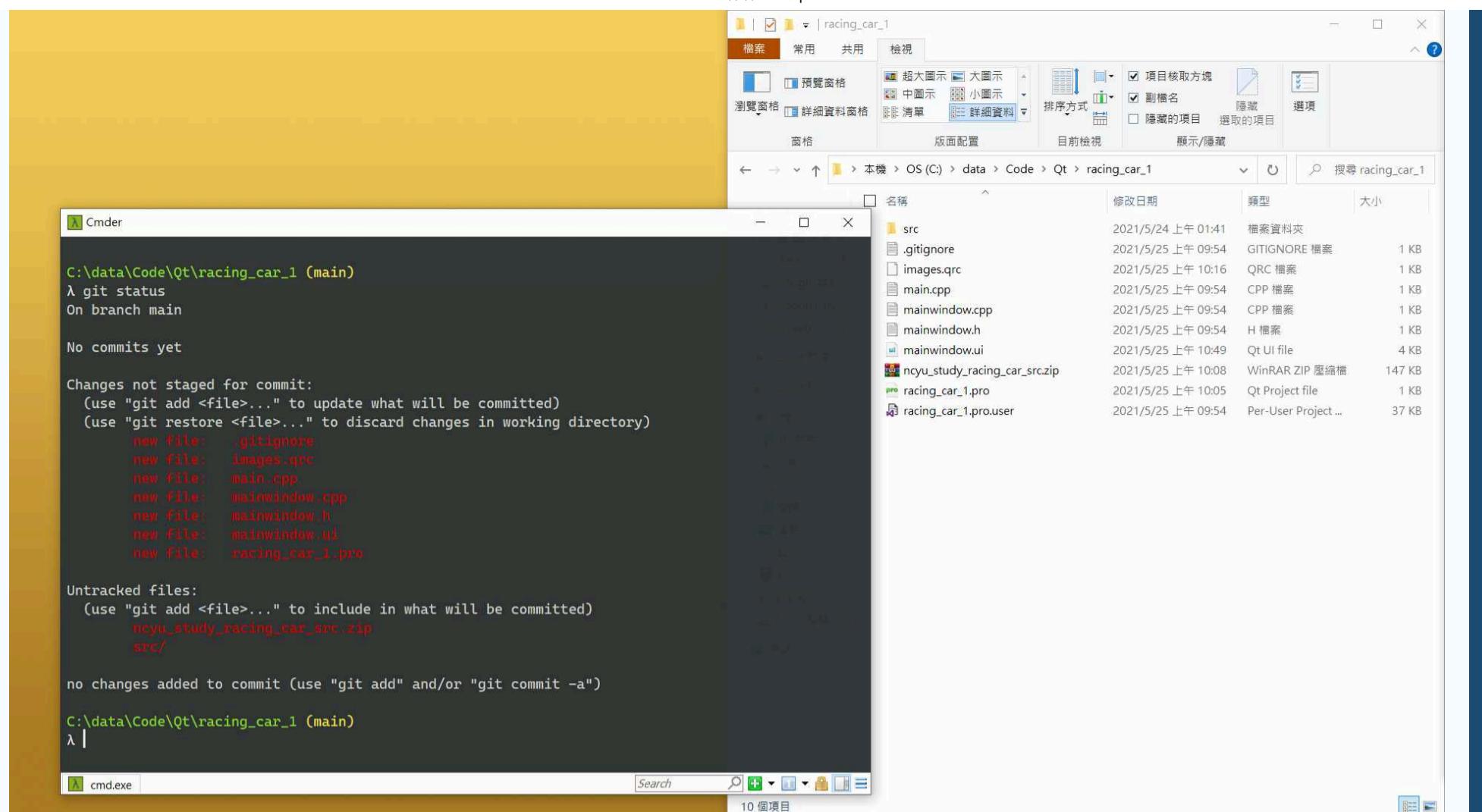
37. (2) 在此使用 PowerShell 做示範，檔案總管中的專案資料夾上方選單選擇【檔案】>【開啟 Windows PowerShell】。

接下來使用 `git status` 查看目前程式碼狀態，跳至第 39 步。



38. (3) 在此使用 Cmder 做示範，開啟 Cmder 使用 `cd` 指令前往專案資料夾的路徑。

接下來使用 `git status` 查看目前程式碼狀態。



39. 將所有檔案交由 git 管理，輸入 `git add .` 指令，再次使用 `git status` 查看目前程式碼狀態。

The screenshot shows a Cmder terminal window with the following command history and output:

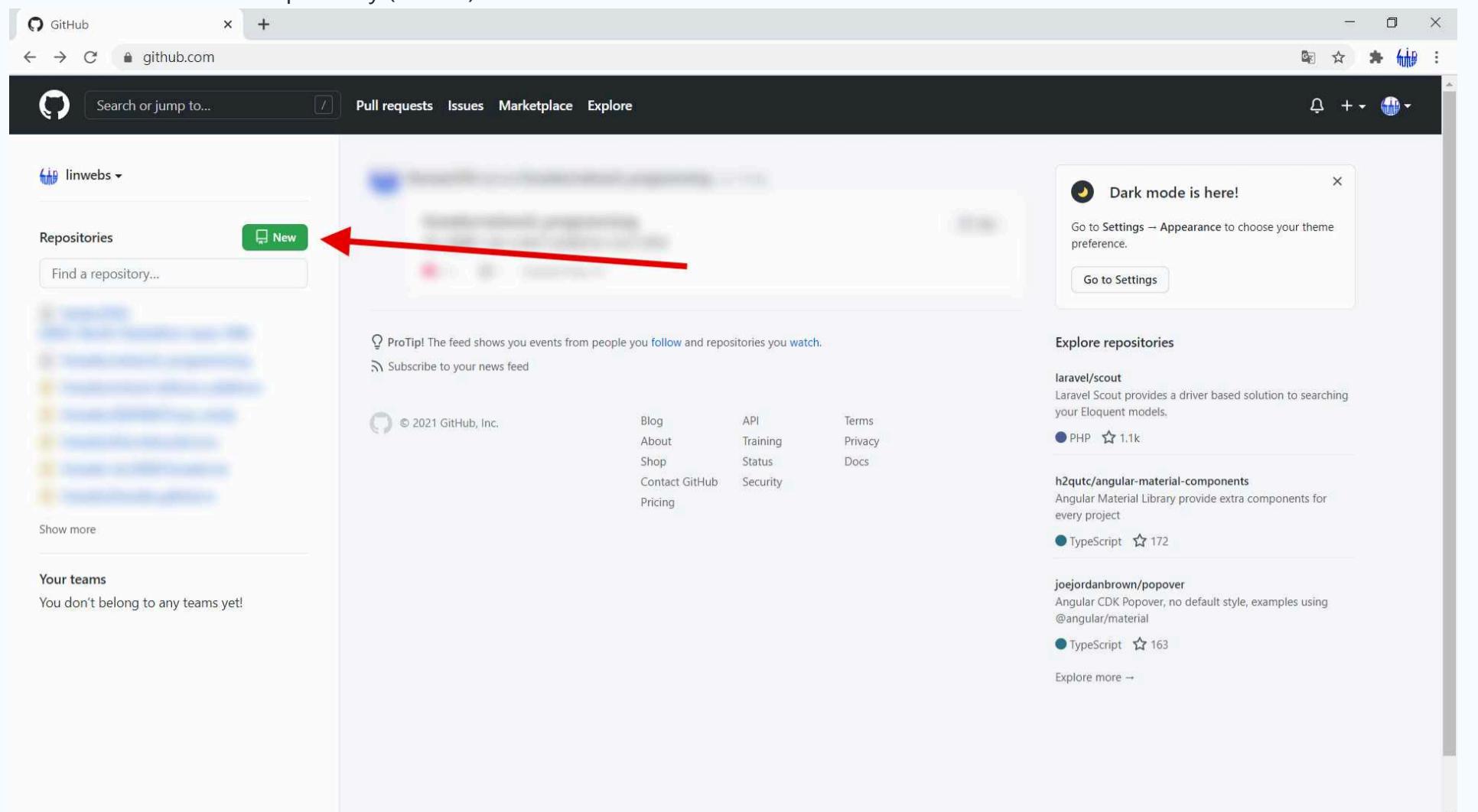
```
C:\data\Code\Qt\racing_car_1 (main)
λ git add .
warning: CRLF will be replaced by LF in .gitignore.
The file will have its original line endings in your working directory
warning: CRLF will be replaced by LF in images.qrc.
The file will have its original line endings in your working directory
warning: CRLF will be replaced by LF in main.cpp.
The file will have its original line endings in your working directory
warning: CRLF will be replaced by LF in mainwindow.cpp.
The file will have its original line endings in your working directory
warning: CRLF will be replaced by LF in mainwindow.h.
The file will have its original line endings in your working directory
warning: CRLF will be replaced by LF in mainwindow.ui.
The file will have its original line endings in your working directory
warning: CRLF will be replaced by LF in racing_car_1.pro.
The file will have its original line endings in your working directory

C:\data\Code\Qt\racing_car_1 (main)
λ git status
On branch main

No commits yet

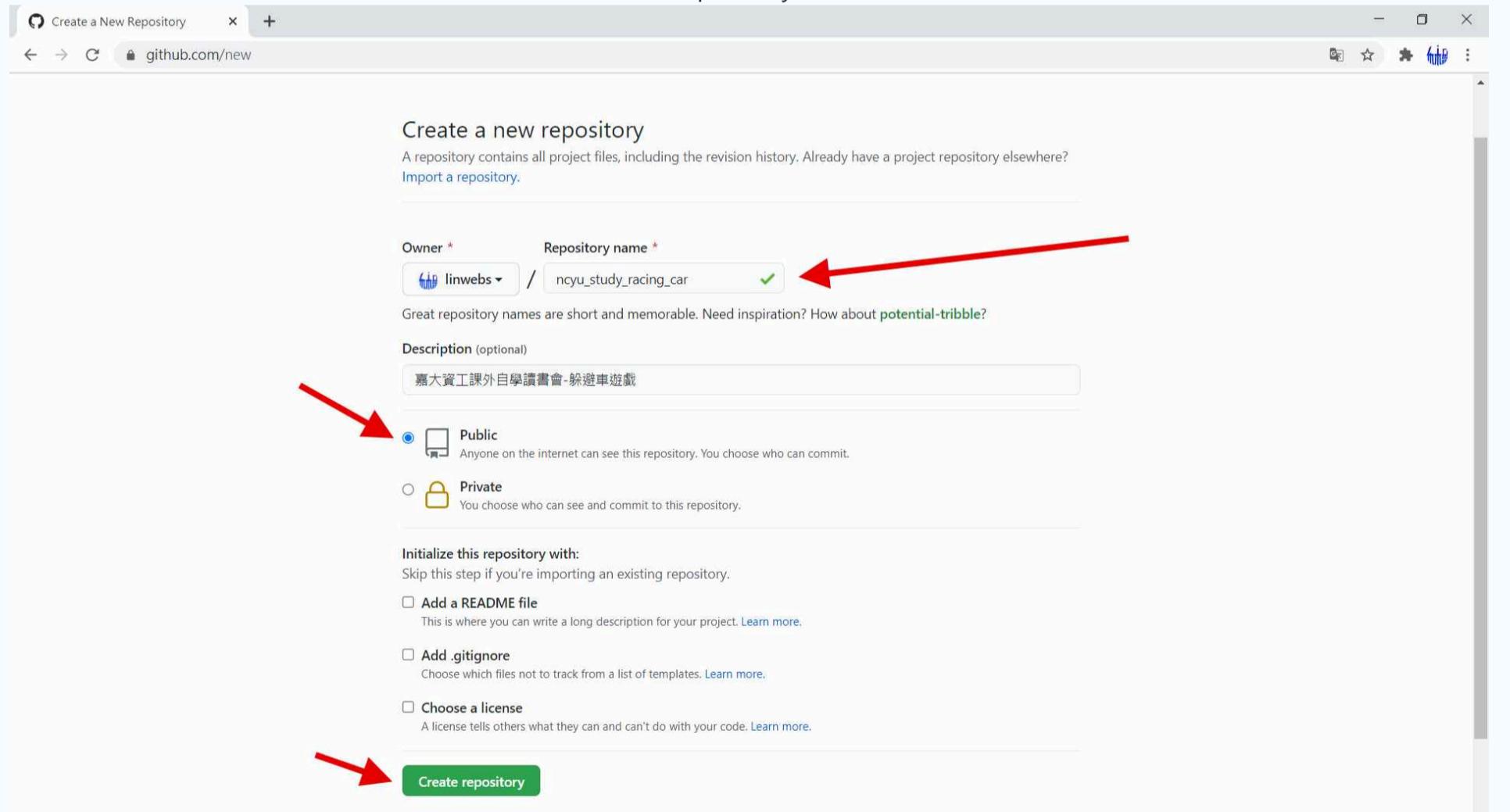
Changes to be committed:
(use "git rm --cached <file>..." to unstage)
  new file:   .gitignore
  new file:   images.qrc
  new file:   main.cpp
  new file:   mainwindow.cpp
  new file:   mainwindow.h
  new file:   mainwindow.ui
  new file:   ncyu_study_racing_car_src.zip
  new file:   racing_car_1.pro
  new file:   src/car.png
  new file:   src/game_failed.png
  new file:   src/game_name.png
  new file:   src/game_pause.png
  new file:   src/game_timeout.png
  new file:   src/icon.ico
```

40. 至 GitHub 新建立一個 Repository (儲存庫)。



The screenshot shows the GitHub homepage. A red arrow points to the green 'New' button located at the top right of the main repository list area. The page includes a search bar, navigation links for Pull requests, Issues, Marketplace, and Explore, and a sidebar for repositories and teams.

41. 輸入儲存庫名稱，選擇儲存庫可見度，接下來點選【Create Repository】。



The screenshot shows the 'Create a new repository' form. A red arrow points to the 'Repository name' field containing 'ncyu_study_racing_car'. Another red arrow points to the 'Public' visibility option, which is selected. A third red arrow points to the 'Create repository' button at the bottom.

Create a new repository
A repository contains all project files, including the revision history. Already have a project repository elsewhere?
[Import a repository](#)

Owner * linwebs / **Repository name *** ncyu_study_racing_car

Great repository names are short and memorable. Need inspiration? How about [potential-tribble](#)?

Description (optional)
嘉大資工課外自學讀書會-躲避車遊戲

Public
Anyone on the internet can see this repository. You choose who can commit.

Private
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

Add a README file
This is where you can write a long description for your project. [Learn more](#).

Add .gitignore
Choose which files not to track from a list of templates. [Learn more](#).

Choose a license
A license tells others what they can and can't do with your code. [Learn more](#).

Create repository

42. 將以下指令貼至終端機介面執行(`git init` 除外，因為 Qt Creator 已幫我們完成此步驟)。

The screenshot shows a GitHub repository page for 'linwebs/ncyu_study_racing_car'. It displays several command-line instructions for initializing a new repository or pushing an existing one. A red arrow points to the first command under the '...or create a new repository on the command line' section:

```
echo "# nceu_study_racing_car" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin git@github.com:linwebs/ncyu_study_racing_car.git
git push -u origin main
```

43. 輸入 `echo xxxxxxx >> README.md` 作為 Readme 文件的標題 (若使用 Cmder 則不須加上引號)。

輸入 `git add README.md` 將檔案交由 git 管理。

輸入 `git commit -m "xxx"` 提交目前 git 暫存區的程式碼，xxx 為 commit 訊息。

The screenshot shows a Cmder terminal window on the left and a GitHub repository page for 'linwebs/ncyu_study_racing_car' on the right. The terminal window displays the following command-line session:

```
C:\data\Code\Qt\racing_car_1 (main)
λ echo # nceu_study_racing_car >> README.md

C:\data\Code\Qt\racing_car_1 (main)
λ git add README.md
warning: CRLF will be replaced by LF in README.md.
The file will have its original line endings in your working directory

C:\data\Code\Qt\racing_car_1 (main)
λ git commit -m "介面設計"
[main (root-commit) 99b59f7] 介面設計
 20 files changed, 316 insertions(+)
 create mode 100644 .gitignore
 create mode 100644 README.md
 create mode 100644 images.qrc
 create mode 100644 main.cpp
 create mode 100644 mainwindow.cpp
 create mode 100644 mainwindow.h
 create mode 100644 mainwindow.ui
 create mode 100644 nceu_study_racing_car_src.zip
 create mode 100644 racing_car_1.pro
 create mode 100644 src/car.png
 create mode 100644 src/game_failed.png
 create mode 100644 src/game_name.png
 create mode 100644 src/game_pause.png
 create mode 100644 src/game_timeout.png
 create mode 100644 src/icon.ico
 create mode 100644 src/item1.png
 create mode 100644 src/item2.png
 create mode 100644 src/item3.png
 create mode 100644 src/map.png
 create mode 100644 src/start_btn.png
```

The GitHub repository page shows the 'Quick setup' section with the same command-line steps. The 'git commit -m "first commit"' line is highlighted in green.

44. 輸入 `git branch -M main` 切換到 main 分支。

輸入 `git remote add origin xxx` 設定 git 遠端儲存庫。

輸入 `git push -u origin main` 將程式碼 push 到遠端儲存庫(GitHub) 上的main 分支。

```

create mode 100644 images.qrc
create mode 100644 main.cpp
create mode 100644 mainwindow.cpp
create mode 100644 mainwindow.h
create mode 100644 mainwindow.ui
create mode 100644 ncyu_study_racing_car_src.zip
create mode 100644 racing_car_1.pro
create mode 100644 src/car.png
create mode 100644 src/game_failed.png
create mode 100644 src/game_name.png
create mode 100644 src/game_pause.png
create mode 100644 src/game_timeout.png
create mode 100644 src/icon.ico
create mode 100644 src/item1.png
create mode 100644 src/item2.png
create mode 100644 src/item3.png
create mode 100644 src/map.png
create mode 100644 src/start_btn.png

C:\data\Code\Qt\racing_car_1 (main)
λ git branch -M main

C:\data\Code\Qt\racing_car_1 (main)
λ git remote add origin git@github.com:linwebs/ncyu_study_racing_car.git

C:\data\Code\Qt\racing_car_1 (main)
λ git push -u origin main
Enumerating objects: 23, done.
Counting objects: 100% (23/23), done.
Delta compression using up to 8 threads
Compressing objects: 100% (22/22), done.
Writing objects: 100% (23/23), 293.37 KiB | 1.53 MiB/s, done.
Total 23 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:linwebs/ncyu_study_racing_car.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.

C:\data\Code\Qt\racing_car_1 (main -> origin)
λ

```

45. 恭喜您已完成介面的設計，接下來請閱讀下一篇文章【[跨平台圖形化程式開發 - \(3\)賽車遊戲開發-程式碼實作](#)】撰寫程式碼。

建立時間：2021/4/26 AM 4:06

修改時間：2021/5/26 PM 10:04

作者: Linwebs

標籤