**作業系統期末專案報告  
第14組**

**103703013 資科三 黃育萱**

**103703015 資科三 蔡雨芝**

**103702012 心理三 陳冠聞**

**目錄**

1. **報告實作主題、內容簡介**
2. **Thread程式碼與功能介紹**
3. **Multi-Thread之間的同步與合作**
4. **根據Poisson Distribution產生客人**
5. **GUI呈現與附加功能**
6. **Github與版本管理、分工**
7. **組員心得**
8. **參考資料**

**一、報告實作主題、內容簡介**

**主題：The Sleeping Barber (SB) Problem**

**語言：**C ++ 11

**關鍵：**Multi-thread, synchronization, without race conditions.

**內容：**

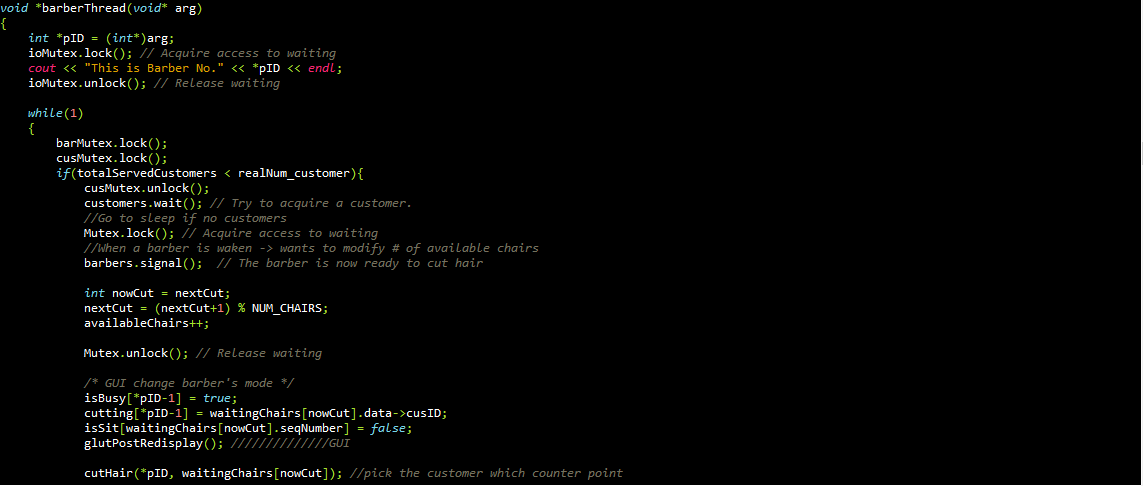
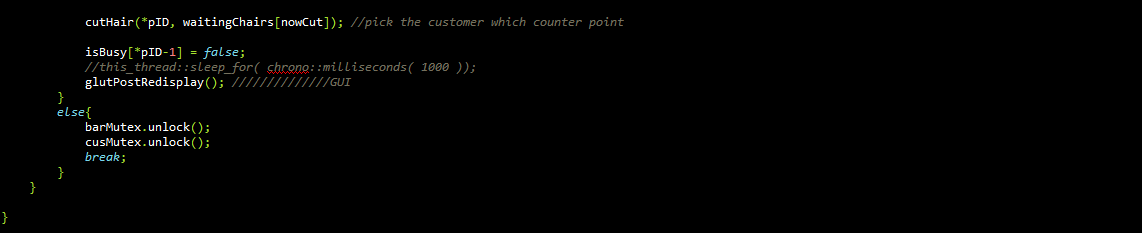
The barber shop has m barbers with m barber chairs, and n chairs (m < n) for waiting customers, if any, to sit in. If there are no customers present, a barber sits down in a barber chair and falls asleep. When a customer arrives, he has to wake up a sleeping barber.

If additional customers arrive while all barbers are cutting customers’ hair, they either sit down (if there are empty chairs) or leave the shop (if all chairs are full). The thread synchronization problem is to program the barbers and the customers without getting into race conditions.

1. **Thread程式碼與功能介紹**

**1. barberThread：**

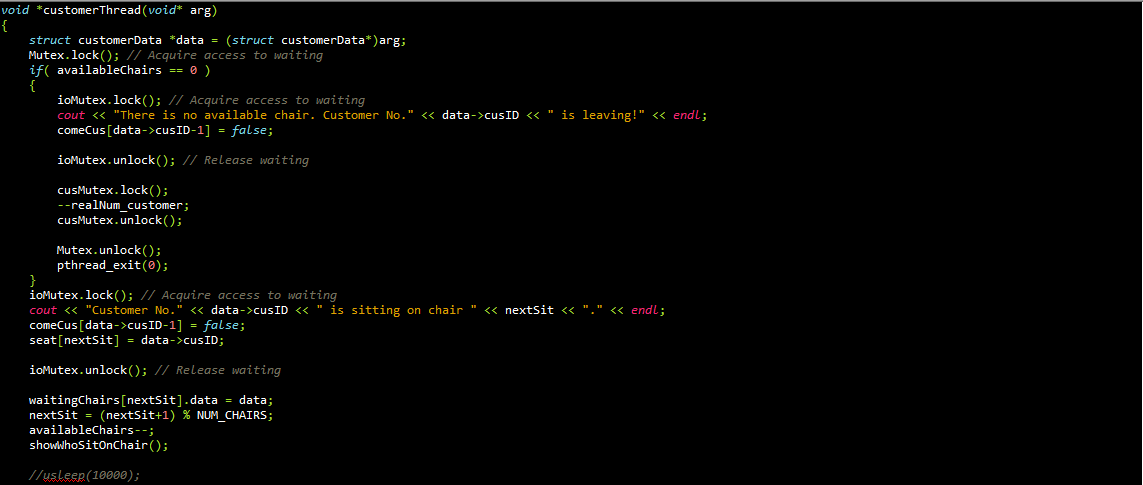
每個barber各自擁有獨立的thread，用來控管barber的狀態（現在是sleeping or cutting hair的狀態）。

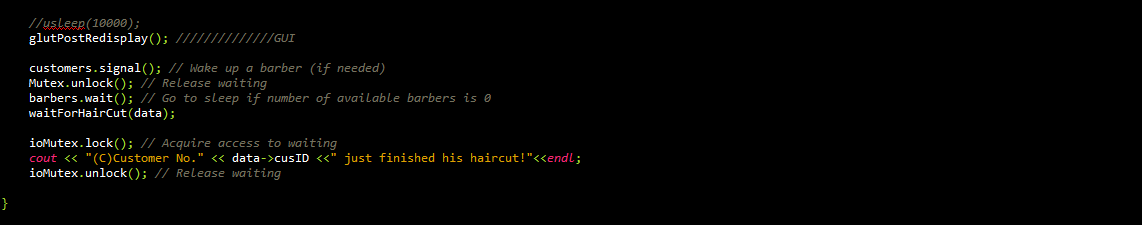


1. **Thread程式碼與功能介紹 (cont.)**

**2. customerThread：**

每個customer各自擁有獨立的thread，用來模擬客人的行為。當客人進入理髮廳：會判斷是否有空的座位，若有的話就坐下等待、沒有的話就離開。





**三、Multi-Thread之間的同步與合作**

**# 透過Semaphore & Mutex避免Race condition：**

**Semaphore barbers(NUM\_BARBERS);**

現在available的barber數量( 有NUM\_BARBERS個barber是有空的 )

**Semaphore customer(0);**

現在的客人數量( 一開始有0個客人 )

**mutex Mutex;**

barbers要搶客人時，為了避免race condition，設立的Mutex。

**mutex cusMutex;**

當有發生客滿（客人離開），或是客人剪完頭髮的狀況，  
為了避免有兩個事件同時修改customer的相關數據，而設立的Mutex。

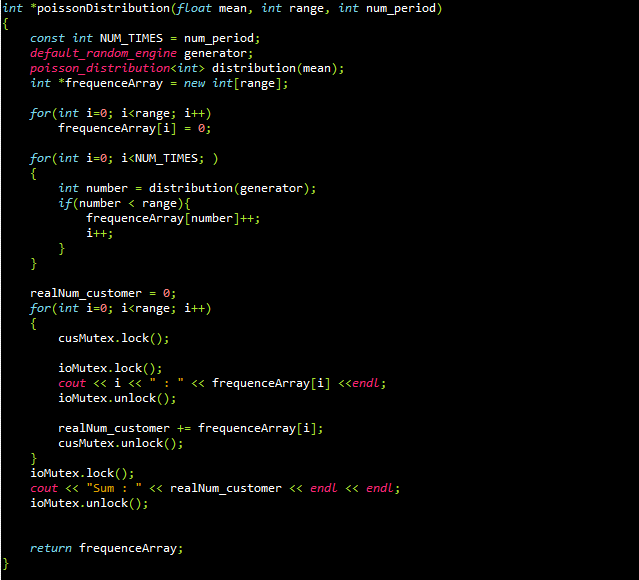
**mutex barMutex;**

避免有兩個barbers同時在等待最後一個客人，造成無限等待（因為只剩一個客人需要被服務，但兩個barbers同時發現還有客人，便開始等待）的狀況，而設立的Mutex。

**四、根據Poisson Distribution產生客人**

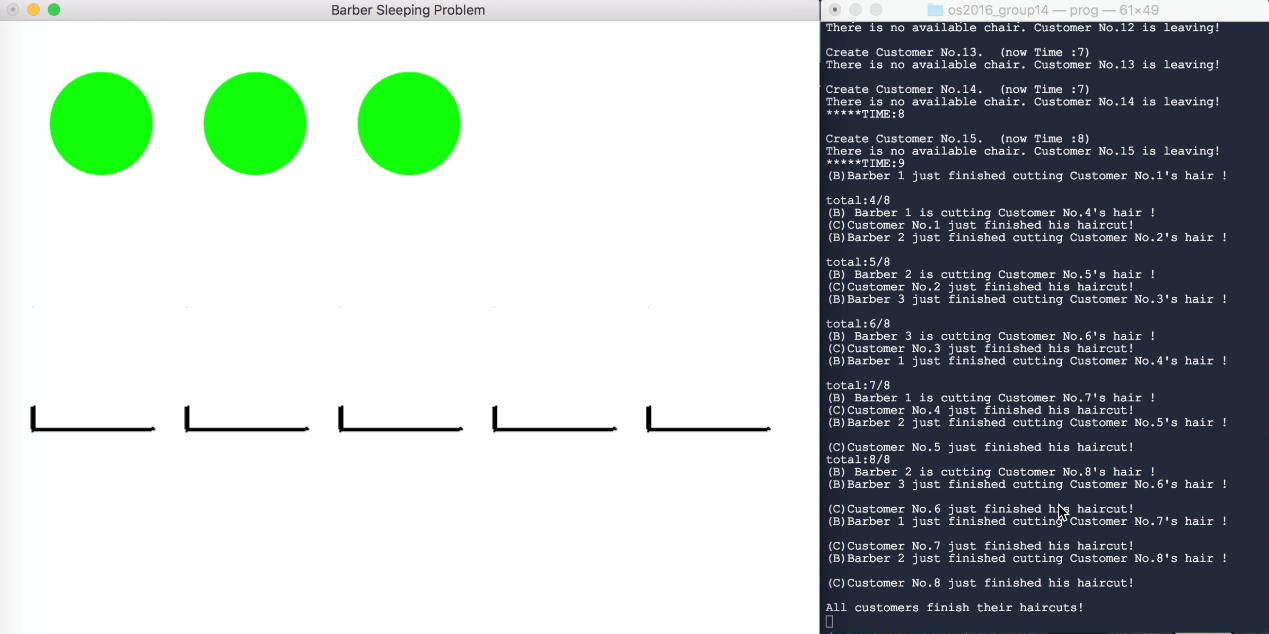
用Poisson Distribution的方式，分配在該秒鐘內會產生幾個客人。

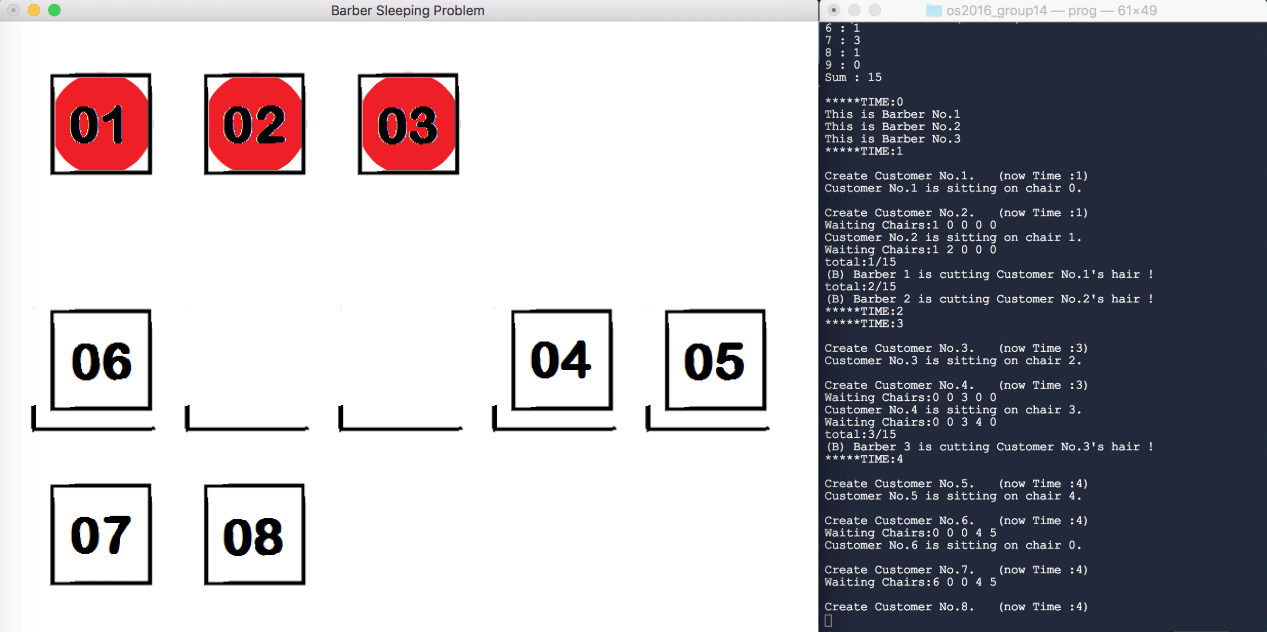
**※註：Demo時經過老師提醒才知道理解錯原題目的意思**



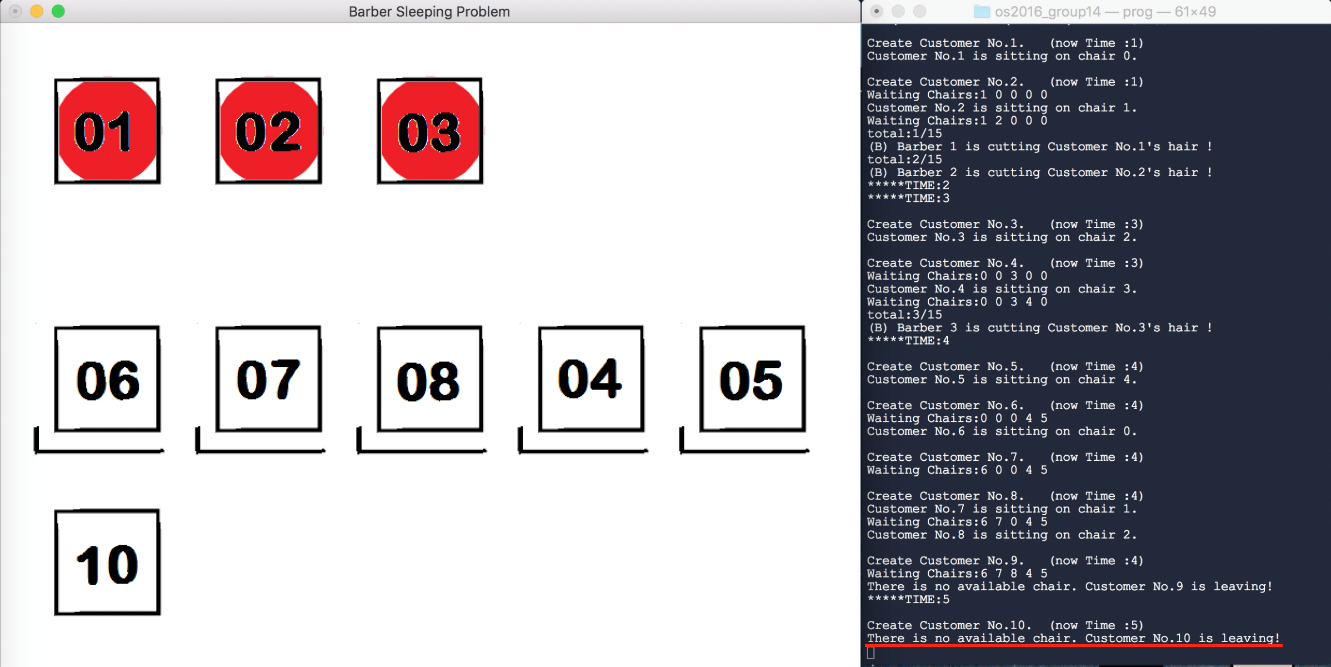
**五、GUI呈現與附加功能**

**# 初始/結束畫面（附圖為結束）**

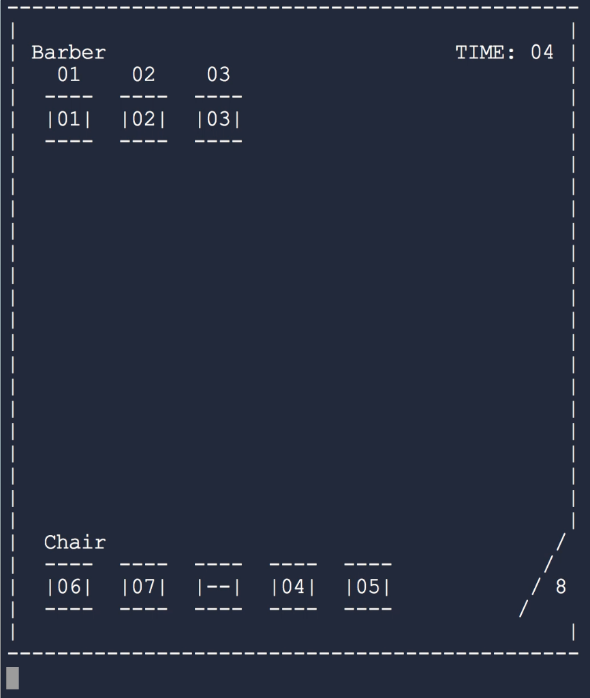
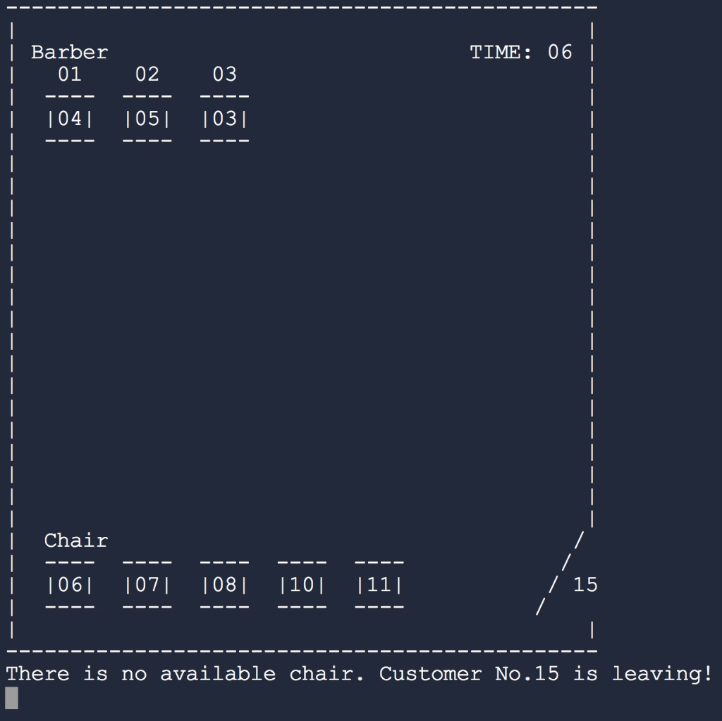


**# 客人進入理髮店等待**

**五、GUI呈現與附加功能(cont.)**

**# 若客人進入理髮店時無位置則離開**

**# 附加功能：文字UI顯示**

**六、Github與版本管理、分工**

**# Version 1：<pthread.h>(適用Windows/ Ubuntu(Mac OS有編譯問題))**

[SB01~SB10.cpp]  
虛擬碼撰寫：陳冠聞  
c++程式碼撰寫、優化：陳冠聞、蔡雨芝

**# Version 2：Windows -> Mac OS(解決Mac OS編譯問題)**

[SB11.cpp] 程式碼轉換：黃育萱

**# Version 3：程式碼優化、轉換為適用於Windows版本**

[SB12、SB13win.cpp] 程式碼優化：陳冠聞、蔡雨芝

**# Version 4：GUI (using Qt)**

[SB13.cpp] GUI程式碼：黃育萱

**# Version 5：<thread> & <mutex>(適用於Windows/Mac OS/Ubuntu)**

[SB15.cpp] 程式碼轉換：蔡雨芝

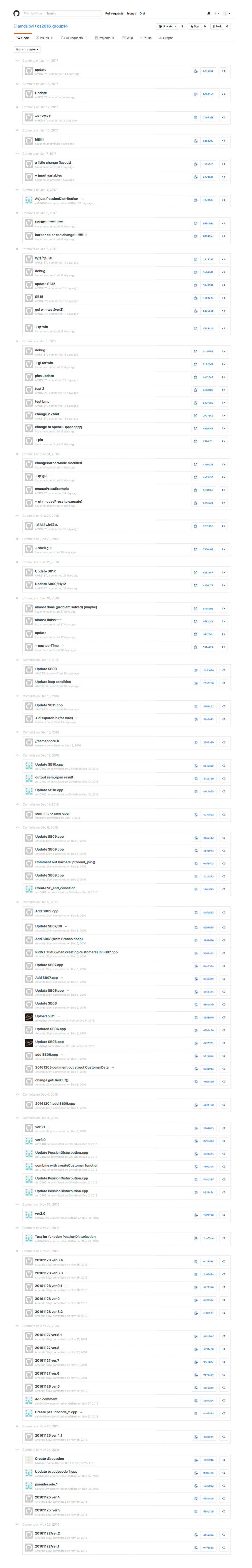
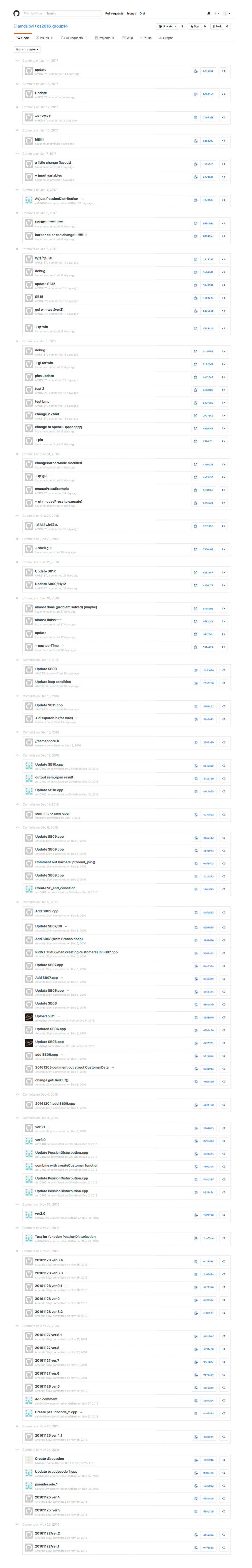
**# Version 6 (Final)：GUI (using OpenGL)**

[SB16.cpp] GUI程式碼：黃育萱

**# 報告書：**

主要撰寫：蔡雨芝

Github截圖、GUI呈現與附加功能：黃育萱



**七、組員心得**

**# 103703013 資科三 黃育萱：**

我負責的工作主要是將有文字輸出的資料圖形化，在組員剛寫完初步文字介面程式的版本，遇到的第一個問題就是作業系統的差別而不能執行（主因為Mac OS系統不能接受無名的Semaphore而有初始問題），所以找了很多方法嘗試，大概經過跟組員完成主程式程式碼的時間差不多。在終於能在電腦裡執行之後，當初先利用QT Creator原本預想在短時間內就能完成，卻因為不能先顯示gui介面再開始執行程式，也試著用openGL實作，因為都不能成功，先做過了一個以符號表示的介面，後來又試著將pthread改成thread及mutex，最後經過跟同學討論中發現實作gui需要以函式呼叫才能正常同步執行（經測試後的QT也可以在multi-thread的環境執行gui）。我覺得從討論到實作其實常常會遇到各種不同的問題，有很多是在跨平台的執行結果不同，就要猜測及檢視程式碼，經過反覆的測試才能完成這個專案。

**# 103703015 資科三 蔡雨芝：**

一開始在和組員討論、撰寫虛擬碼時，還算是順利。但後來加入隨機的客人之後，就出現了不少問題，有些小細節很容易被忽略，因此就要靠著多次嘗試才會發現bug。在Demo的時候也發現原來我們誤解了部分題目的意思（例如Poisson、Daemon），有些細節也沒有做得很好（Thread數量的限制（沒辦法無限生成）、GUI更新畫面速度太快等等）。在跨OS的時候也出現了許多編譯器的相關問題，還好後來找到<dispatch>函式庫才解決了相關問題。實作時雖然曾出現了許多跨OS的棘手問題，但也因此對於各種OS有更深入的了解。

**# 103702012 心理三 陳冠聞：**

這次的期末project剛開始覺得滿簡單的，只要做好thread的同步就好。但後來發現在一些題目的要求下，有些實作確實也有些難度，甚至我們誤解了一些題目的意思，有些功能也不能做到完美，如不能產生太多客人等等。本身身為非本系的學生，這學期透過這次期末project學到滿多東西的，例如github的操作、用途，撰寫不同部分code彼此之間的協調，測試結果錯誤如何找bug等等。

**八、參考資料**

**1. C++ Reference** [**http://www.cplusplus.com/reference/**](http://www.cplusplus.com/reference/)

**2. pthread程式撰寫** [**http://blog.xuite.net/nikoung/wretch/154071878**](http://blog.xuite.net/nikoung/wretch/154071878)

**3. multi-thread同步** [**http://blog.yam.com/swwuyam/article/11459124**](http://blog.yam.com/swwuyam/article/11459124)

**4. C++multi-thread程式開發** [**https://kheresy.wordpress.com/2012/07/06/multi-thread-programming-in-c-thread-p1/**](https://kheresy.wordpress.com/2012/07/06/multi-thread-programming-in-c-thread-p1/)

**5. 使用 Signal 與 Slot（使用按鈕關閉視窗）**

[**http://openhome.cc/Gossip/Qt4Gossip/SimpleSignalSlot-1.html**](http://openhome.cc/Gossip/Qt4Gossip/SimpleSignalSlot-1.html)

**6. Semaphore on OS X**[**http://stackoverflow.com/questions/1413785/sem-init-on-os-x**](http://stackoverflow.com/questions/1413785/sem-init-on-os-x)