

C++程式設計基礎

lesson 8-2

陳毅

本週概要

- 物件導向程式設計基本概念 – 最初的需求
- 專案管理 : Sourcetree + GitHub
- 其他程式語言簡介
- 課程回顧

需求

- 請為銀行設計一個管理程式，用以管理客戶的帳戶及資料，並處理相關業務。
 - 一間銀行擁有許多客戶和許多帳戶。客戶名稱不會重複。帳戶的帳號也不會重複。
 - 客戶有名稱、出生年月日。
 - 每個客戶可以向銀行開設一個以上的帳戶。
 - 每個帳戶記載著帳號、客戶、歷史交易紀錄等資料。
 - 交易紀錄有2種類型：存入、提出。當客戶向銀行提出交易時，若交易成功，會將交易紀錄記載在帳戶中。
 - 客戶可以向銀行查詢自己帳戶的歷史交易紀錄。

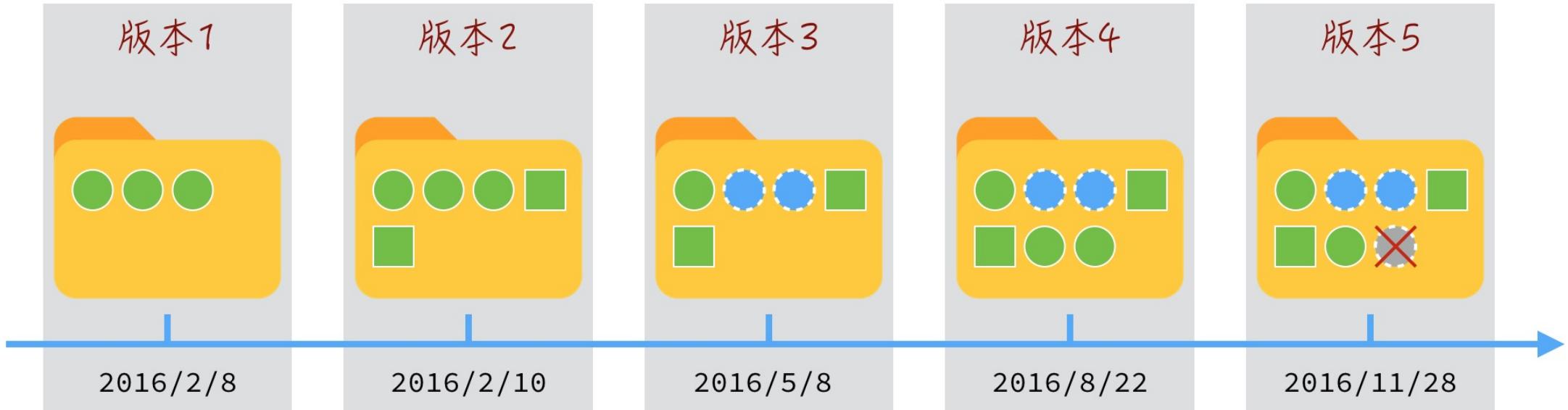
需求

- 請為銀行設計一個管理程式，用以管理客戶的帳戶及資料，並處理相關業務。
 - 一間銀行擁有許多客戶和許多帳戶。客戶名稱不會重複。帳戶的帳號也不會重複。
 - 客戶有名稱、出生年月日。
 - 每個客戶可以向銀行開設一個以上的帳戶。
 - 每個帳戶記載著帳號、客戶、歷史交易紀錄等資料。
 - 交易紀錄有2種類型：存入、提出。當客戶向銀行提出交易時，若交易成功，會將交易紀錄記載在帳戶中。
 - 客戶可以向銀行查詢自己帳戶的歷史交易紀錄。

專案管理：Sourcetree + GitHub

Git是什麼？

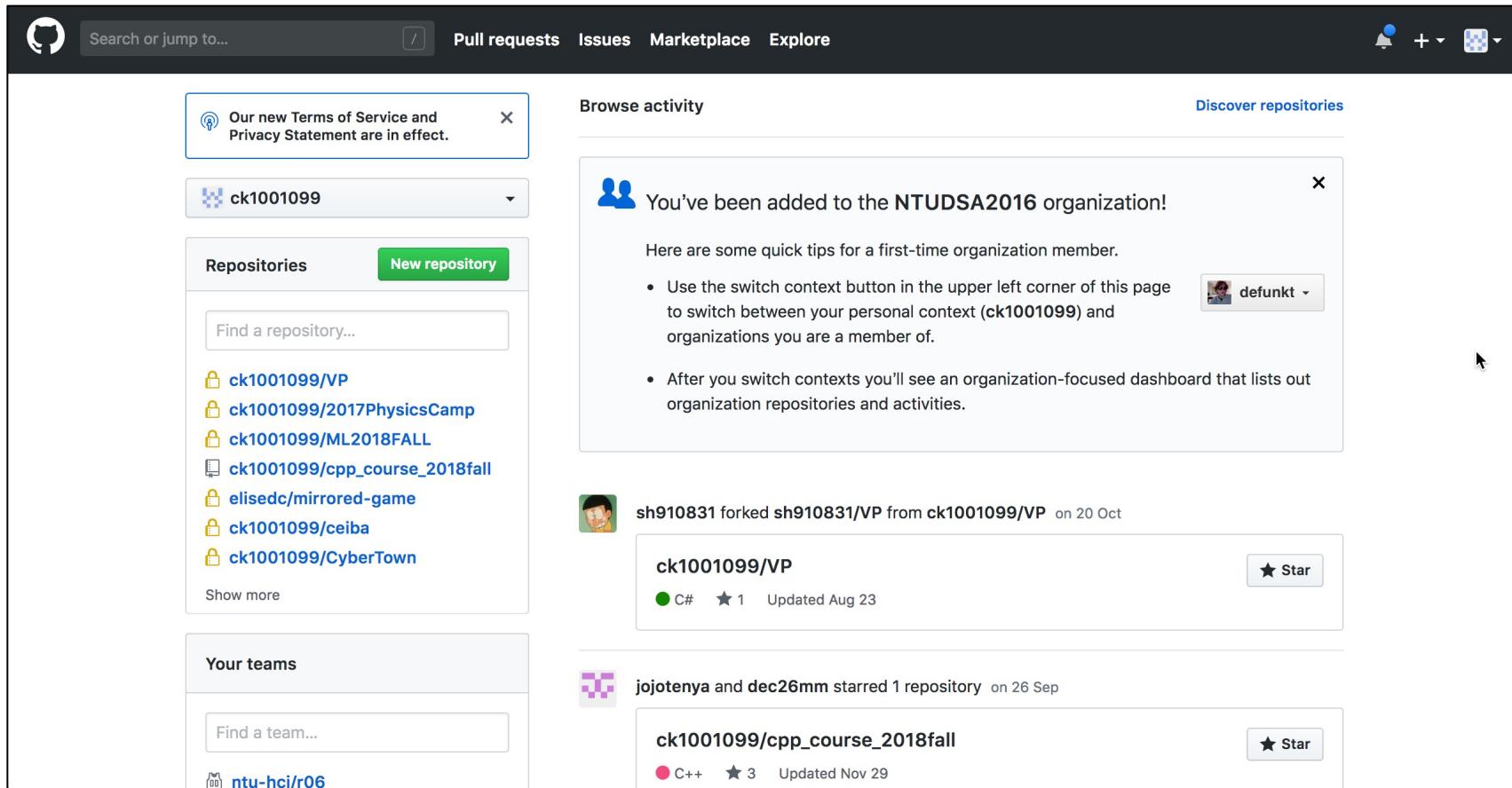
- Git是一種分散式版本的**版本控制系統(Version Control System)**。



- 常見的線上版本控制平台：GitHub、BitBucket等。

GitHub 介紹

- GitHub: <https://github.com/>



Sourcetree 介紹

- Sourcetree: <https://www.sourcetreeapp.com/>

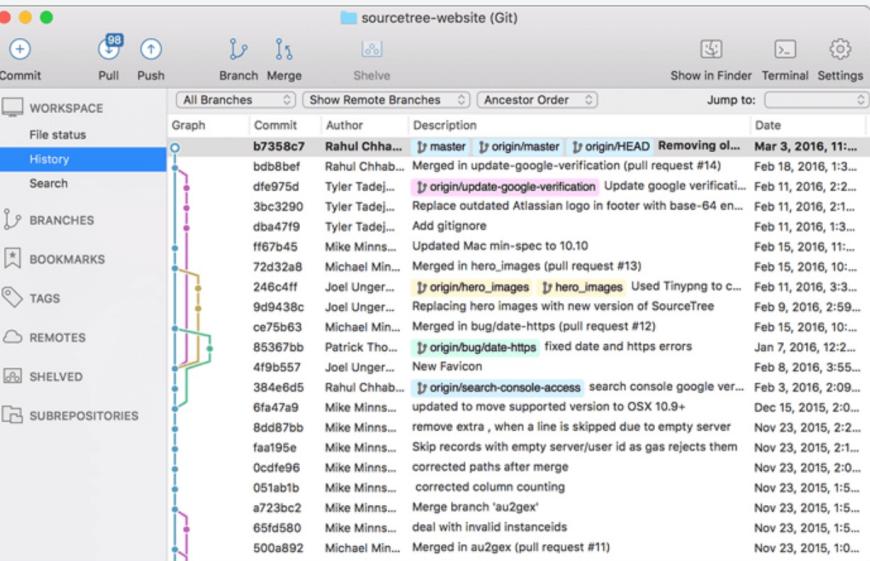
 Sourcetree

[Download free](#)

Simplicity and power in a beautiful Git GUI

[Download for Mac OS X](#)

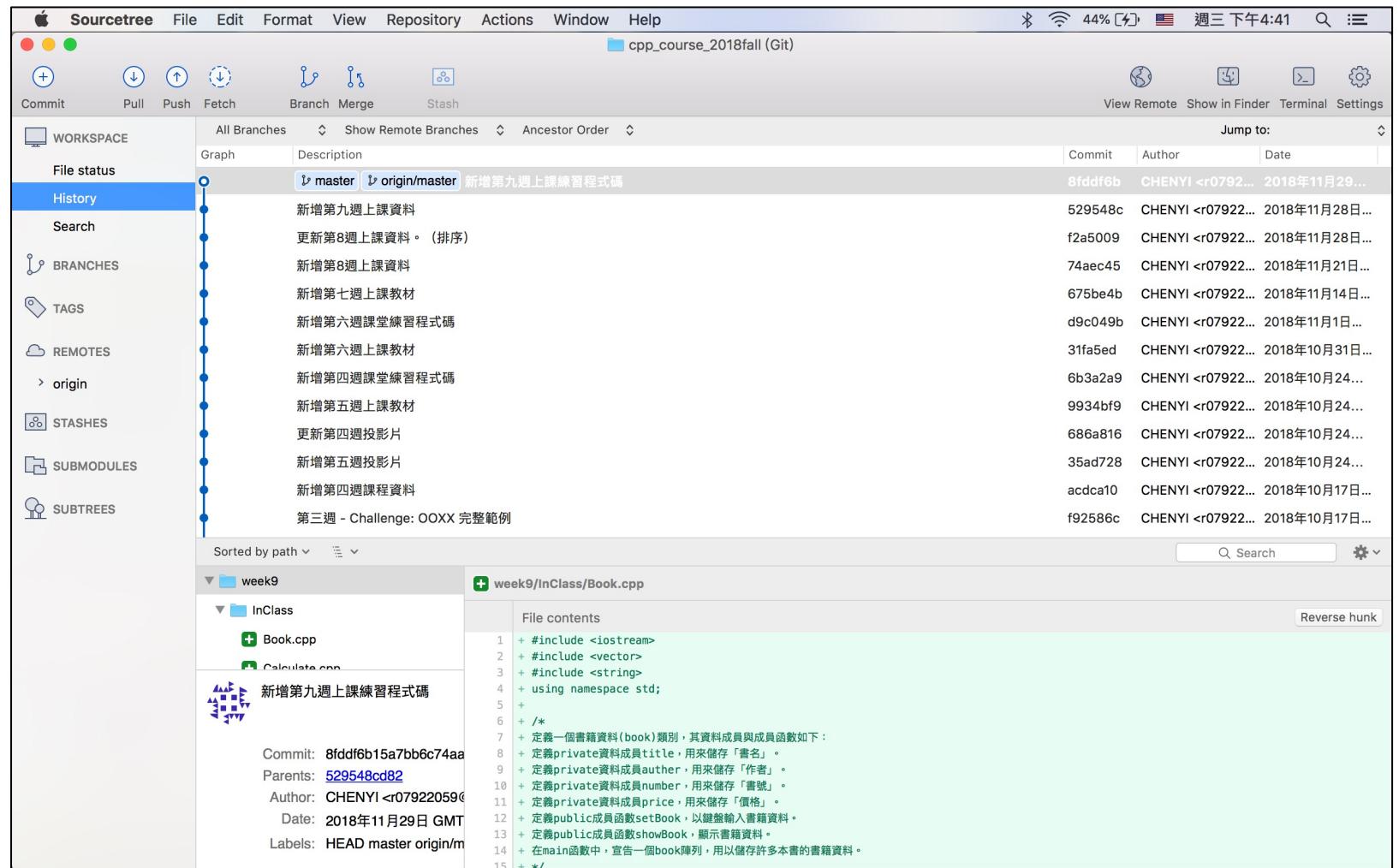
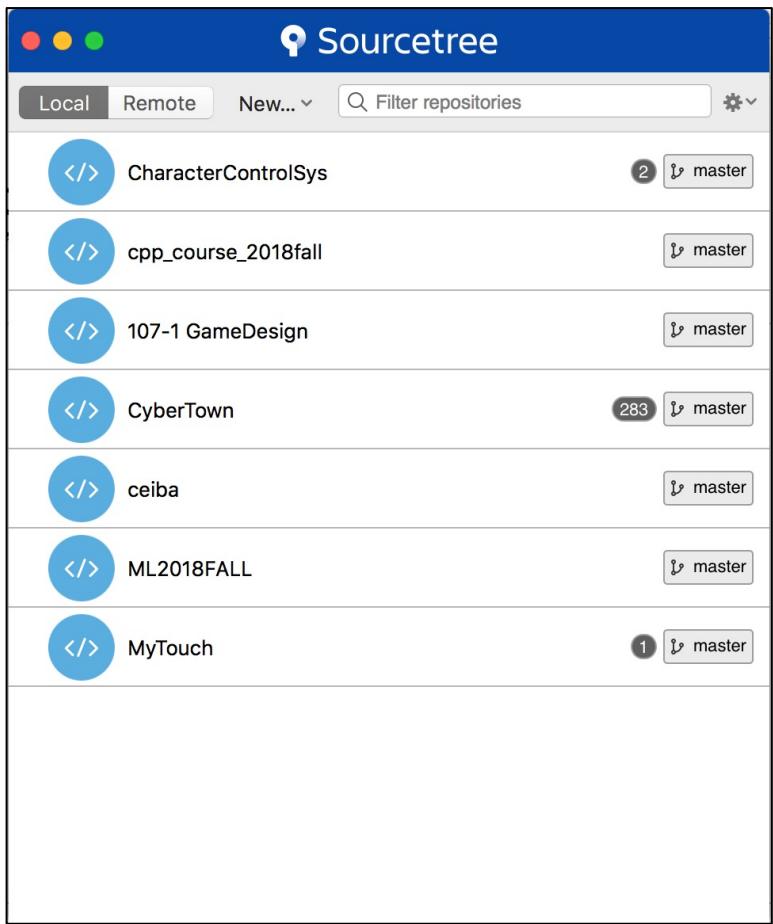
Also available for Windows



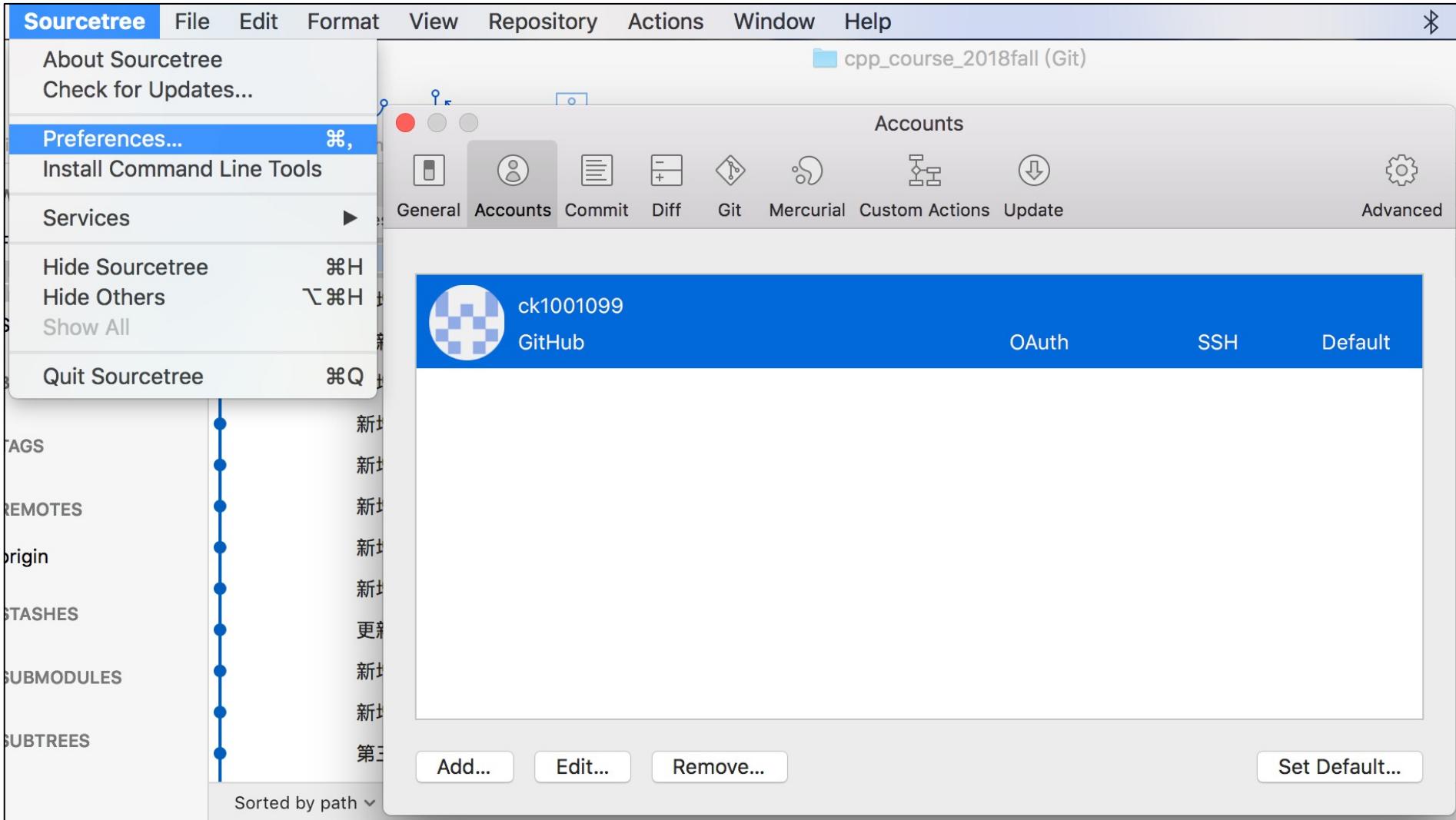
A free Git client for Windows and Mac

Sourcetree simplifies how you interact with your Git repositories so you can focus on coding. Visualize and

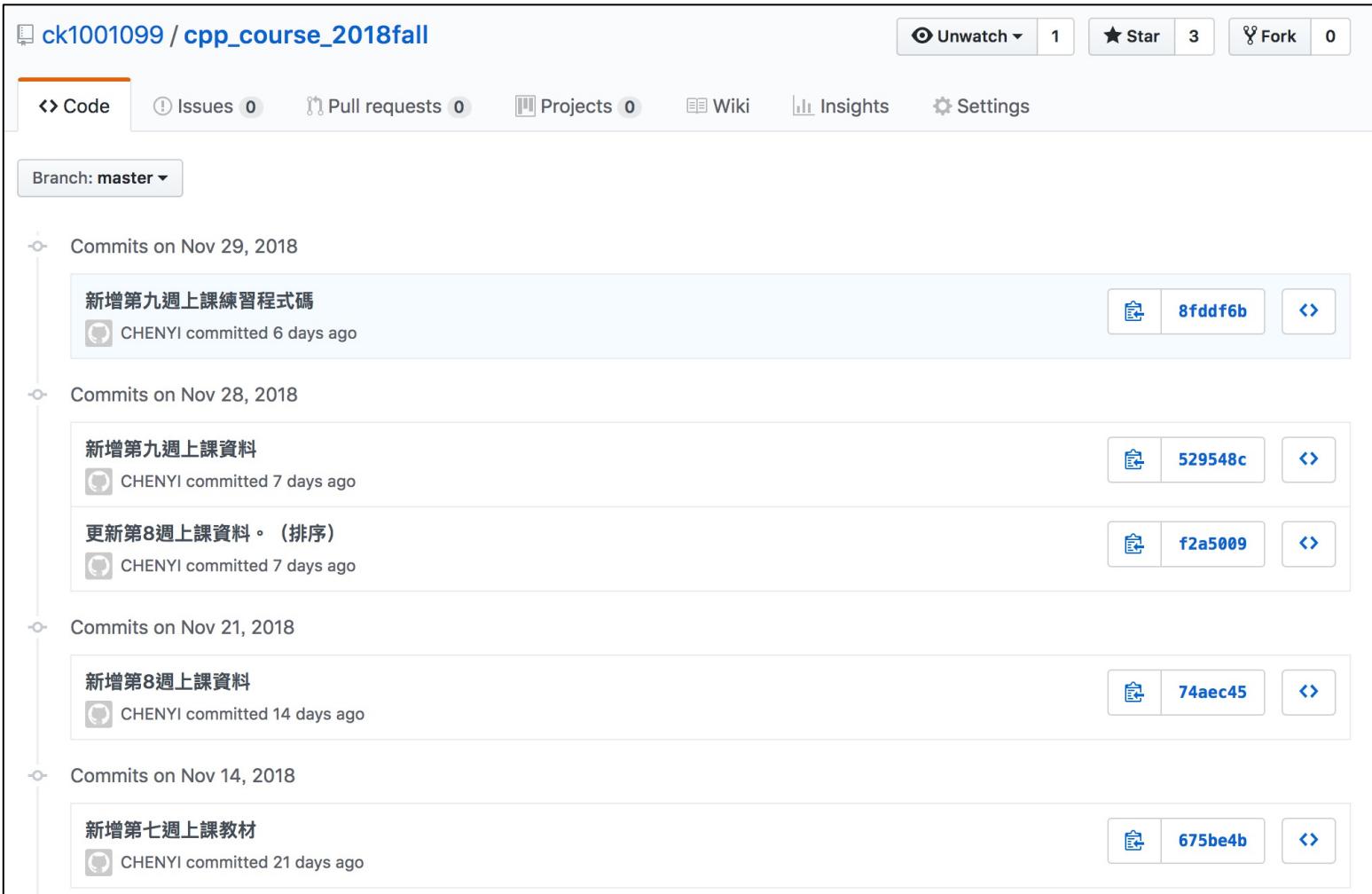
Sourcetree + GitHub



Sourcetree + GitHub



Sourcetree + GitHub



補充資料

- 30 天精通 Git 版本控管
<https://github.com/doggy8088/Learn-Git-in-30-days>
- GitHub 版本控制
<https://hackmd.io/s/Bk2AaU6o>
- Git 的基本介紹
https://backlog.com/git-tutorial/tw/intro/intro1_1.html
- 一步一步教你用 GitHub + SourceTree 做版本控制
<https://hackmd.io/s/Bymbf3y2>

其他程式語言簡介

常用程式語言

語言名稱	用途
C	數據處理、研究用
python	機器學習、爬蟲、數據處理
C#	遊戲開發（Unity主要使用語言）
swift	iOS開發
java	Android開發
R	統計專用
php	網頁後端
javascript	網頁前端

python (ML)

Show data

In [4]:

```
input_shape = (256,256,3)

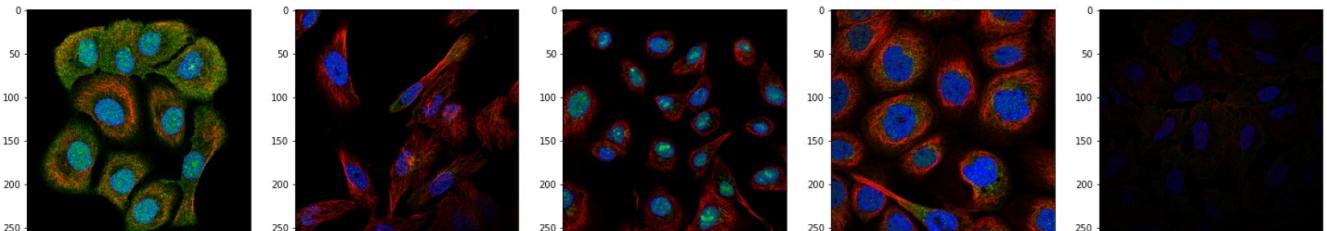
# create train datagen
train_datagen = data_generator.create_train(
    train_dataset_info, 5, input_shape, augment=True)
```

In [5]:

```
images, labels = next(train_datagen)

fig, ax = plt.subplots(1,5,figsize=(25, 5))
for i in range(5):
    ax[i].imshow(images[i])
print('min: {}, max: {}'.format(images.min(), images.max()))
```

min: 0.0, max: 1.0



Featured Prediction Competition

Human Protein Atlas Image Classification

Classify subcellular protein patterns in human cells

Human Protein Atlas · 1,440 teams · a month to go (a month to go until merger deadline)

Overview Data Kernels Discussion Leaderboard Rules Team My Submissions Submit Predictions

Overview

Description

In this competition, Kagglers will develop models capable of classifying mixed patterns of proteins in microscope images. [The Human Protein Atlas](#) will use these models to build a tool integrated with their smart-microscopy system to identify a protein's location(s) from a high-throughput image.

Evaluation

Prizes

Timeline

Special Prize Instructions

Proteins are “the doers” in the human cell, executing many functions that together enable life. Historically, classification of proteins has been limited to single patterns in one or a few cell types, but in order to fully understand the complexity of the human cell, models must classify mixed patterns across a range of different human cells.

Images visualizing proteins in cells are commonly used for biomedical research, and these cells could hold the key for the next breakthrough in medicine. However, thanks to advances in high-throughput



\$37,000 Prize Money

C# (Unity)

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

public class GameController : MonoBehaviour {

    public static GameController _GameController;

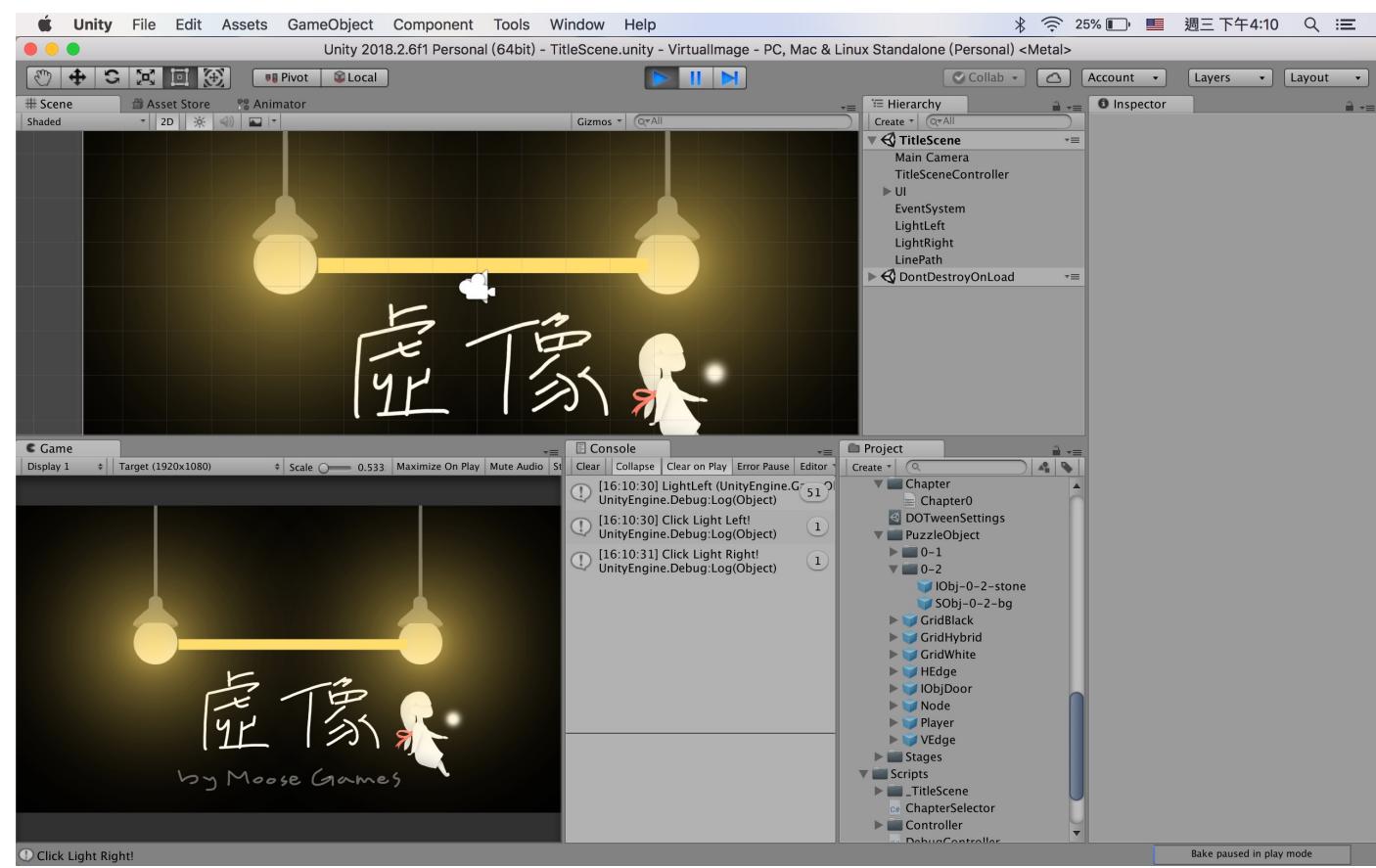
    public GameObject loadingScreen;
    public float loadingTimeMin = 1.0f;

    public string chapterName;

    void Awake(){
        _GameController = this;
        DontDestroyOnLoad(this.gameObject);
        DontDestroyOnLoad(loadingScreen);
    }

    // Use this for initialization
    void Start () {
        Screen.SetResolution(1920, 1080, FullScreenMode.Windowed);
    }

    // Update is called once per frame
    void Update () {
    }
}
```



關於學習程式語言

- Google上幾乎可以找到任何你想學的語言的教學。
- 我們使用過的一些網站也可以幫助你學習
 - Zerojudge
 - HackerRank
 - LeetCode
 - CODESIGNAL
 - FreeCodeCamp
- 學程式語言不外乎就是：**多看，多想，多練習**。
- 實戰也可以大幅提升你對特定語言的熟悉度。

課程回顧

學習目標

- 能夠獨自撰寫簡易的C++程式
- 了解基礎的物件導向程式設計方法
- 閱讀程式碼的能力

時間表

- 8/4] • 資料型態、變數、基本輸入輸出
- 8/6] • 流程控制（條件控制、迴圈控制）
- 8/9] • 函式與陣列
- 8/11] • 指標
- 8/13]
- 8/16]
- 8/18] • 資料結構
- 8/20] • 物件導向程式設計基礎 (2)

- Reading: 課本 Ch11.5.3, Ch12

課程結束！