**BỘ QUỐC PHÒNG**

**HỌC VIỆN KỸ THUẬT QUÂN SỰ**

KHOA CNTT

-----🙞🙜🕮🙞🙜-----

****

**BẢN BÁO CÁO**

***Môn: Thực Tập Nhóm***

***Giáo viên hướng dẫn:* Thầy Cao Tuấn Anh**

Hà Nội, Ngày 28 tháng 02 năm 2019

HỌC VIỆN KỸ THUẬT QUÂN SỰ

KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO MÔN THỰC TẬP NHÓM

ĐỀ TÀI: Game 2048

THÀNH VIÊN NHÓM : Lê Hoàng Cương

Trần Cao Nguyễn

Hoàng Duy Thái

Bùi Nhật Hưng

**LỜI MỞ ĐẦU**

Hiện nay, làm việc nhóm là một kỹ năng không thể thiếu trong một công việc, mỗi cá nhân có thể làm được một công việc, nhưng khi làm việc nhóm công việc có thể hoàn thành sớm hơn và năng suất hơn. Trong ngành CNTT cũng vậy, làm việc nhóm luôn luôn quan trọng.làm việc nhóm thì luôn phải ở cùng nhau làm việc, nhưng hiện nay có nhiều phần mềm có thể giúp bạn làm việc nhóm mà không cần ở gần nhau mà cũng có thể làm việc nhóm được.Có rất nhiều phần mềm hỗ trợ để nhóm em làm đề tài này nhưng phần mềm được nhóm em lựa chọn là phần mềm Git để làm việc nhóm vì nó là hệ thống quản lý source code tập trung.

# 

**Chương I: Tìm hiểu phần mềm quản lý nhóm: GitHub**

1. **Giới thiệu về Github**
2. **Github và Git là gì?**

**-** **GitHub** là một dịch vụ lưu trữ trên web dành cho các dự án có thể sử dụng hệ thống kiểm soát Git revision. Nó là một hệ thống máy chủ dành riêng để chứa các kho này, các repository trên Github được gọi là remote repository. Như vậy, bạn có thể thấy rất rõ ràng sự khác nhau Git là một mô hình hệ thống còn Github là hệ thống máy chủ. Tất nhiên có nhiều dịch vụ giống như Github.

**-Git** là tên gọi của một Hệ thống quản lý phiên bản phân tán (Distributed Version Control System – DVCS) là một trong những hệ thống quản lý phiên bản phân tán phổ biến nhất hiện nay.

1. **Github làm được gì?**

* Đơn giản hơn đó là Git sẽ giúp bạn lưu lại các phiên bản của những lần thay đổi vào mã nguồn. Và có thể dễ dàng khôi phục lại mà không cần copy lại mã nguồn rồi cất vào đâu đó. Một người khác có thể xem các thay đổi của bạn ở từng phiên bản. Họ cũng có thể đối chiếu các thay đổi của bạn rồi gộp phiên bản của bạn vào phiên bản của họ. Cuối cùng là tất cả có thể đưa các thay đổi vào mã nguồn của mình lên một kho chứa mã nguồn
* Cơ chế lưu trữ phiên bản của Git là nó sẽ tạo ra một “ảnh chụp” (snapshot) trên mỗi tập tin và thư mục sau khi commit. Từ đó nó có thể cho phép bạn tái sử dụng lại một ảnh chụp nào đó. Đây cũng chính là lợi thế của Git so với các DVCS khác. Nó không “lưu cứng” dữ liệu mà sẽ lưu với dạng snapshot...Do đó Git là một giải pháp làm việc nhóm từ xa hiệu quả, an toàn.

1. **Lợi ích của việc dùng Git ?**

Có rất nhiều lợi thế để bạn nên sử dụng Git trong việc lập trình ngay từ hôm nay, bất kể là lập trình cái gì đi chăng nữa.

* Git dễ sử dụng, an toàn và nhanh chóng.
* Có thể giúp quy trình làm việc code theo nhóm đơn giản hơn rất nhiều bằng việc kết hợp các phân nhánh (branch).
* Bạn có thể làm việc ở bất cứ đâu vì chỉ cần clone mã nguồn từ kho chứa hoặc clone một phiên bản thay đổi nào đó từ kho chứa, hoặc một nhánh nào đó từ kho chứa.
* Dễ dàng trong việc deployment sản phẩm.
* Và nhiều hơn thế nữa.

1. **So sánh Git với các tool khác**

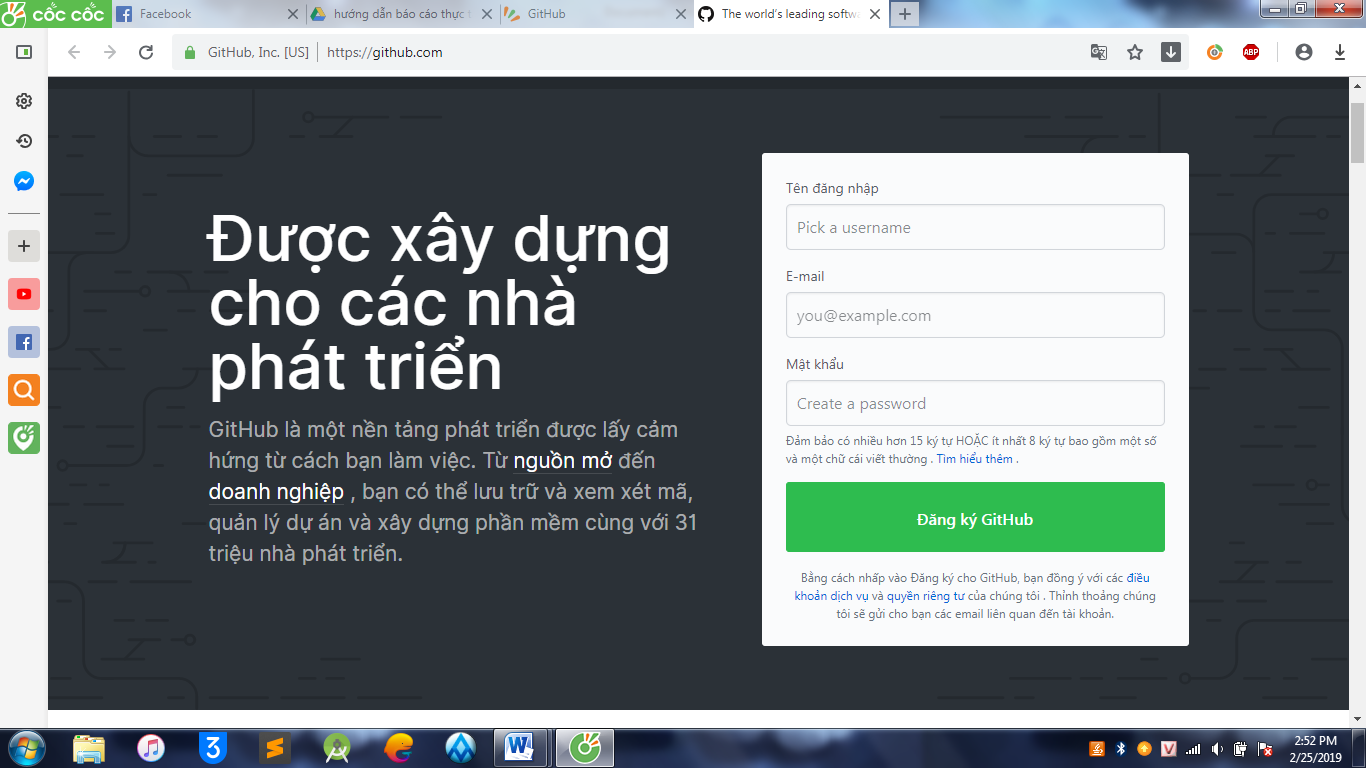
|  |  |  |
| --- | --- | --- |
| **Subversion** | **CVS** | **Git** |
| Là hệ thống quản lý source code trung | Quản lý source code tập trung | Là một phương pháp quản lý theo hướng phân tán |
| Các commit có thể roll-back lại trạng thái trước đó | Không thể roll-back | Các commit có thể roll-back lại trạng thái trước đó |
| Các tập tin được đổi tên hoặc loại bỏ vẫn mang đầy đủ history và meta-data của nó trước đó | Tập tin được đổi tên hoặc di chuyểnsẽ bị mất history trước đó | Commit hoặc update trực tiếp từ máy của họ mà không phải thông qua reponsitory gốc trên server và mang theo cả history |
| Có khả năng đẩy bất cứ những thay đổi mới đến repository cha | Chỉ có thể đẩy những thay đổi mới đến repository con của nó | Tất cả những thao tác làm việc với Git đều ở trên máy của local , local repository , Có thể những thay đổi push lên server nếu muốn |
| Sử dụng SSH, ngoài ra còn bổ sung WebDAV DeltaV, giao thức này được dựa trên HTTP và HTTPS cung cấp cho người dùng một tùy chọn để kết nối với các SVN qua web. | Sử dụng giao thức SSH để truyền tải | Sử dụng SSH để truyền tải. (SSH là 1 giao thức mạng ) |
| Làm việc trong mạng LAN hay Internet | Làm việc trong mạng LAN hay Internet | Làm việc trong mạng LAN hay Internet |
| Nếu server down thì các thành viên khác không thể update thay đổi từ server - > khả năng làm việc offline không cao. | Nếu server down thì các thành viên khác không thể update thay đổi từ server - > khả năng làm việc offline không cao | Có khả năng làm việc offline(server down),vì mỗi bản copy của thành viên đều là full copy từ repository gốc trên máy thành viên (update trực tiếp không cần server). |
| Thao tác trên GUI hay command | Thao tách trên GUI | Thao tách trên GUI hay command |

1. **Cài đặt và sử dụng Git**
2. **Nguyên tắc hoạt động**

Để làm việc với **GitHub**bạn cần:

1. Đăng ký một tài khoản **GitHub**và tạo một **Repository** (**GitHub Repository**).
2. Cài đặt **GitHub Desktop**, một công cụ trực quan quản lý **Local Repository** (Kho chứa dữ liệu địa phương).
3. Cấu hình để có thể đồng bộ hóa dữ liệu bằng **GitHub Desktop** lên **Repository server**.
4. **Đăng ký tài khoản GitHub**

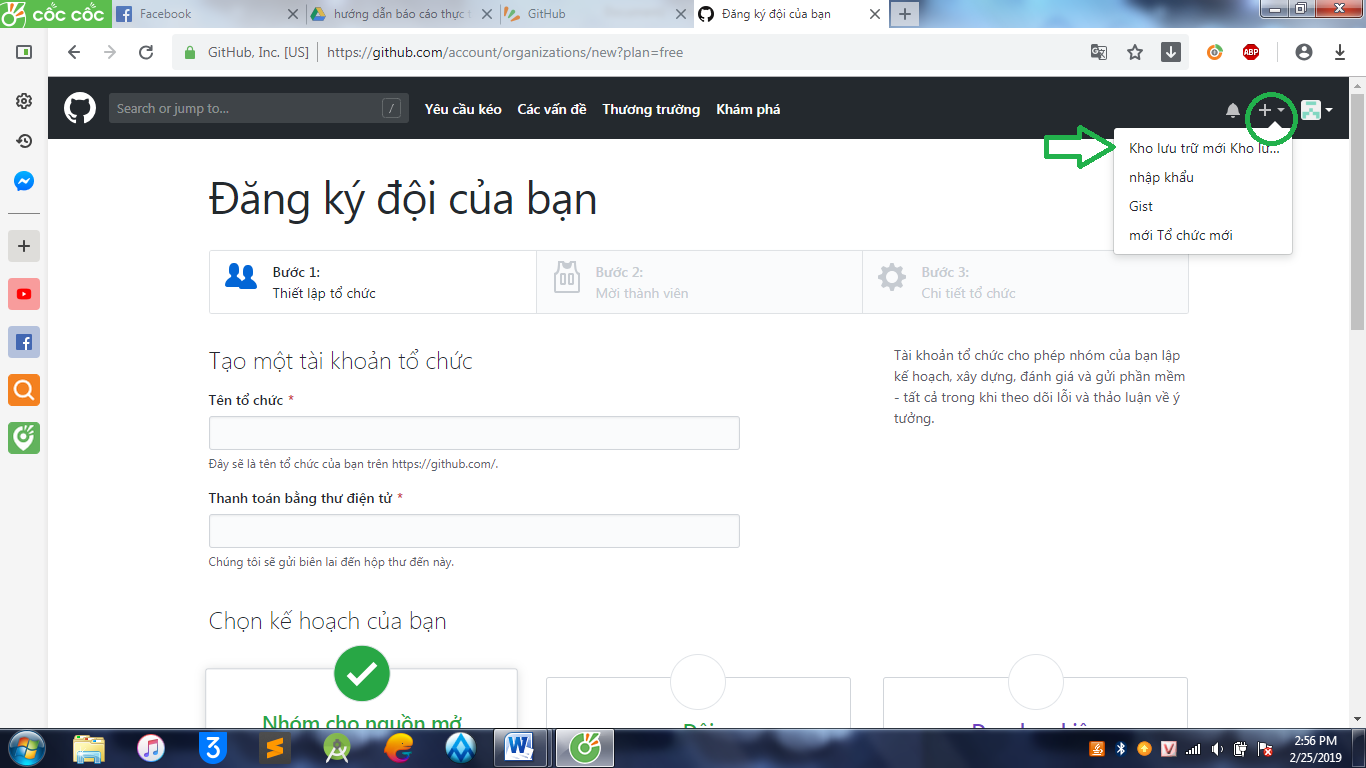
* Trước hết bạn cần phải đăng ký miễn phí một tài khoản **GitHub**. Bạn có thể vào trang chủ của **GitHub**tại: [https://github.com](https://github.com/)

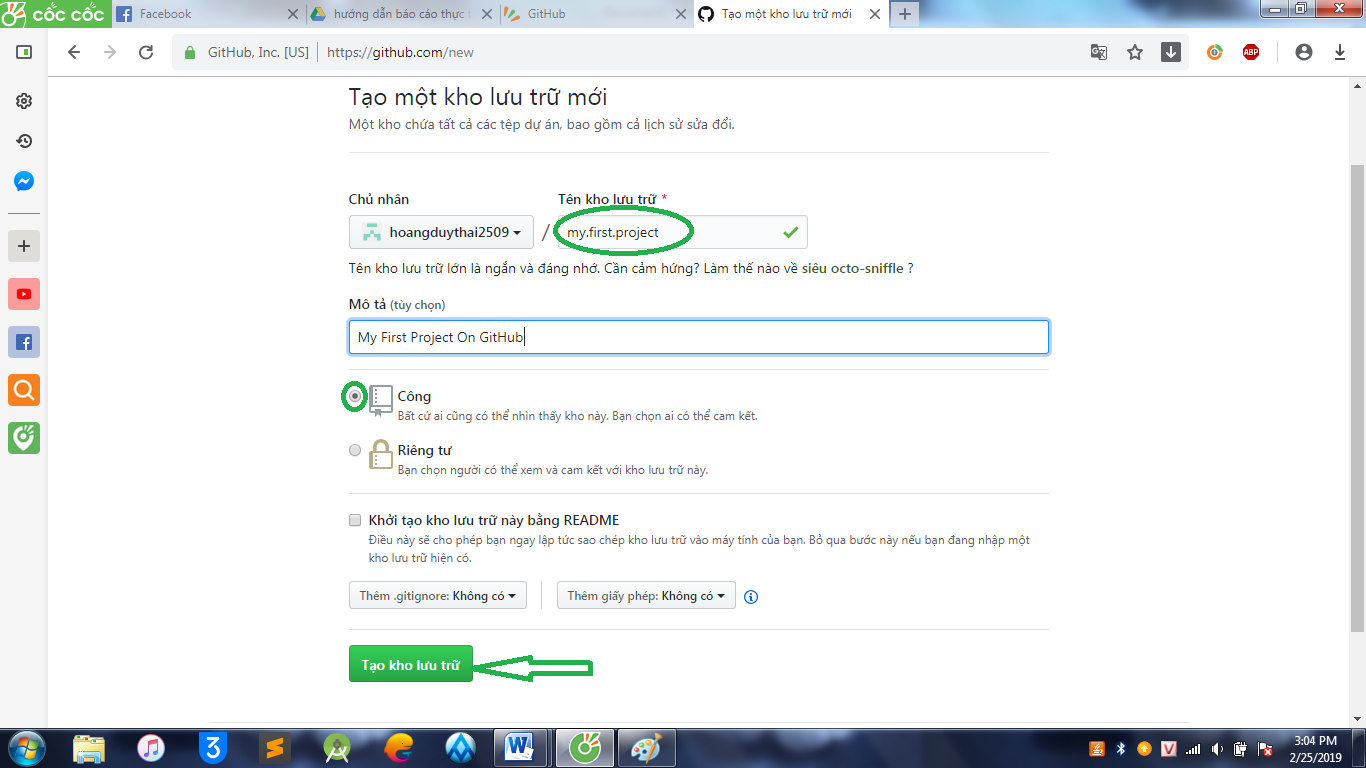


Việc đăng ký một tài khoản là đơn giản, bạn chỉ cần nhập **username/password** và địa chỉ **email**. Sau khi đăng ký xong bạn cần vào Email kích hoạt tài khoản.

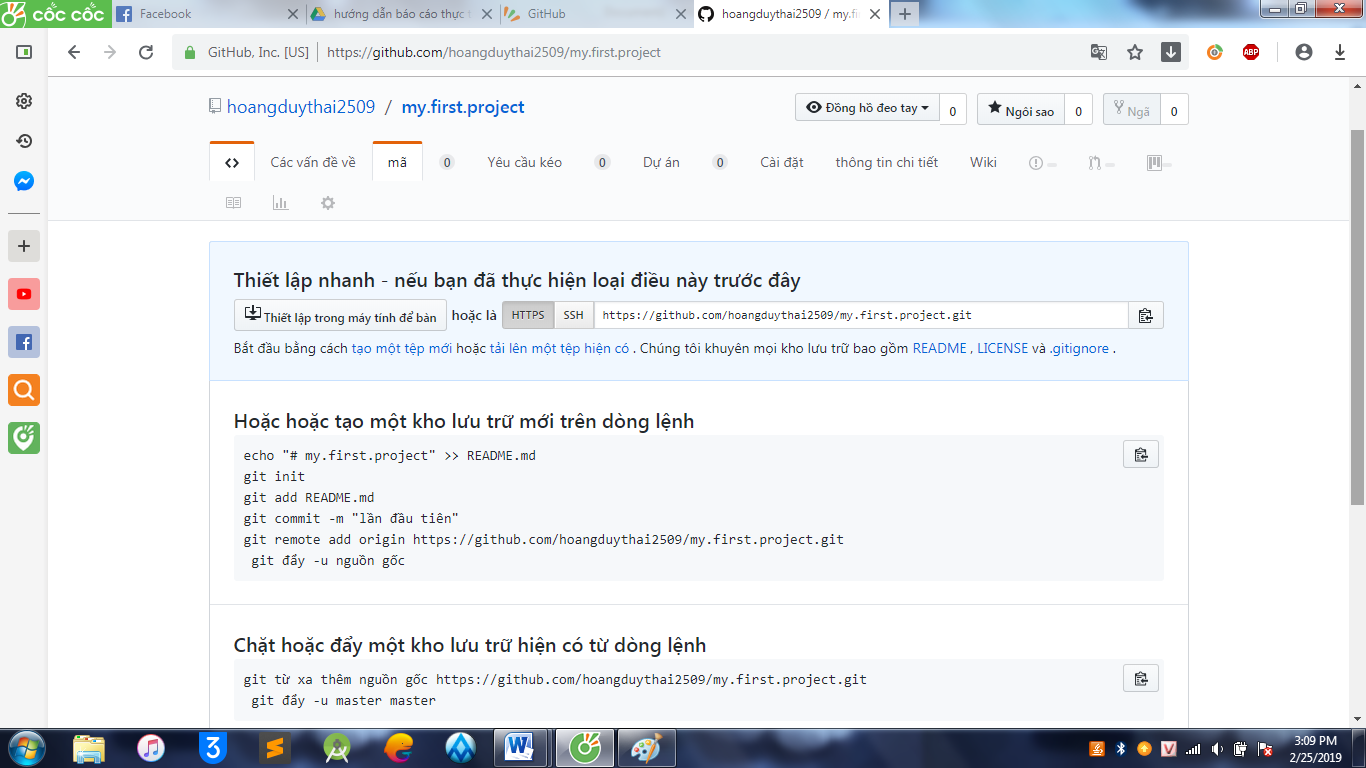
1. **Tạo GitHub Repository**

Sau khi đăng ký xong tài khoản **GitHub**, và đăng nhập vào. Bạn có thể tạo một **GitHub Repository**.



****

**Repository**đã được tạo ra , sau đó bạn có thể tạo nhóm và mời các thành viên.



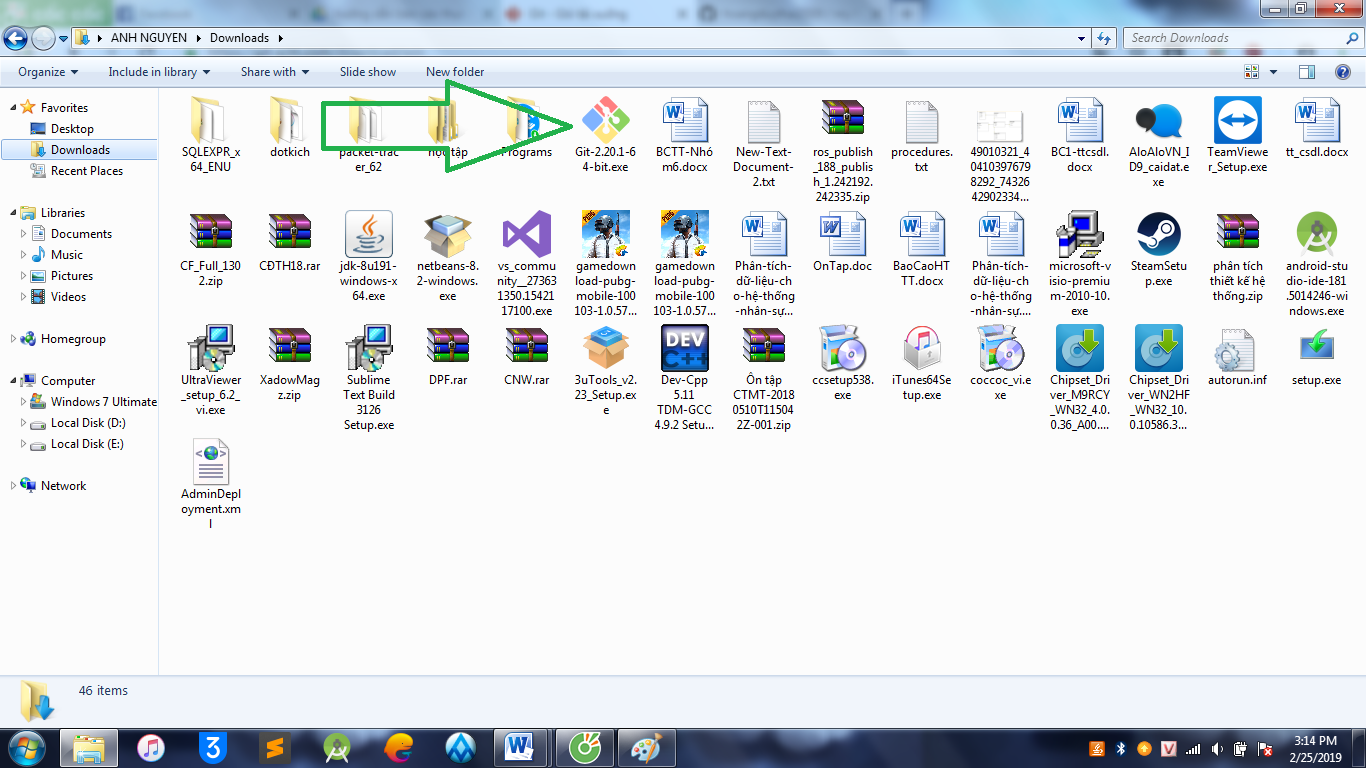
1. **Download & cài đặt Git**

-Để download **Git**  bạn vào địa chỉ:

<https://git-scm.com/downloads>

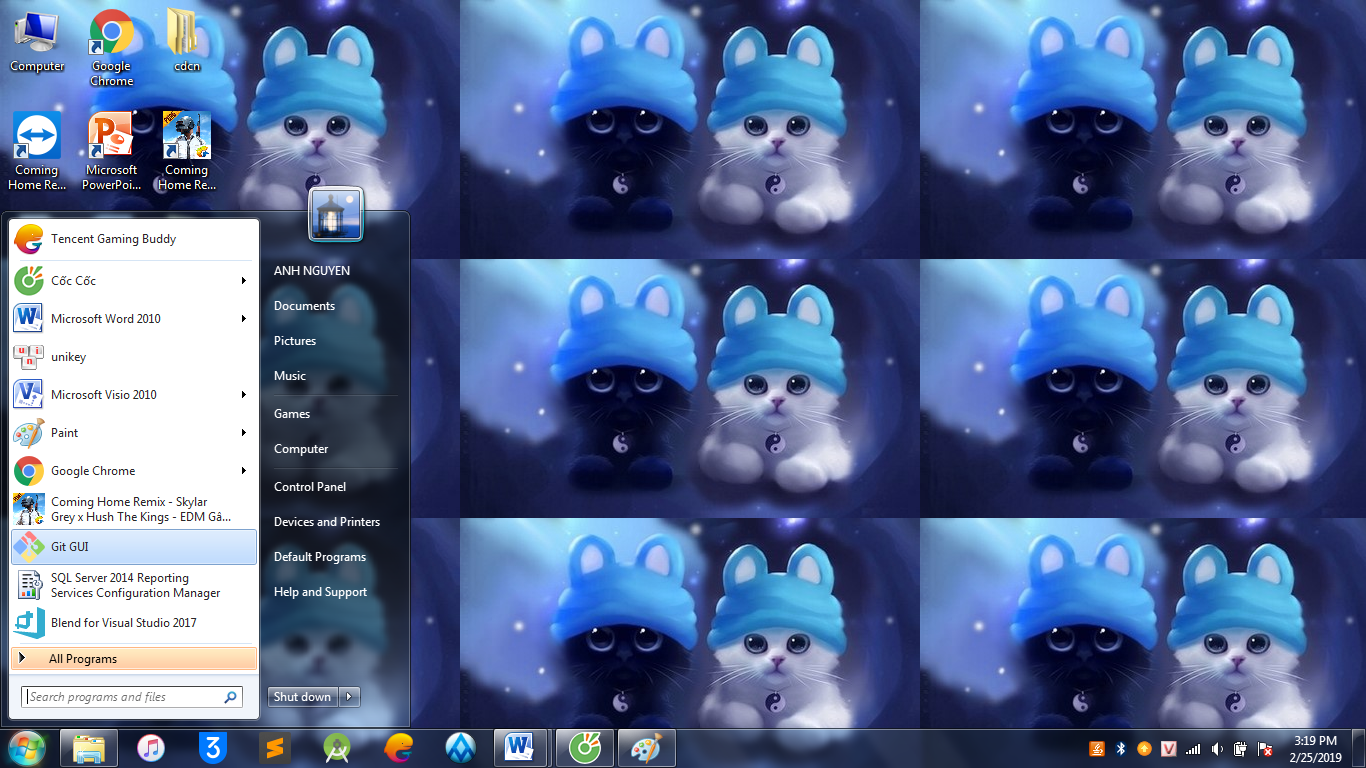


*-*Kết quả download:

****

Sau khi download xong, bạn cài đặt **Git** vào máy tính:

**Git**đã được cài đặt thành công.



1. **Sử dụng Git và GitHub**

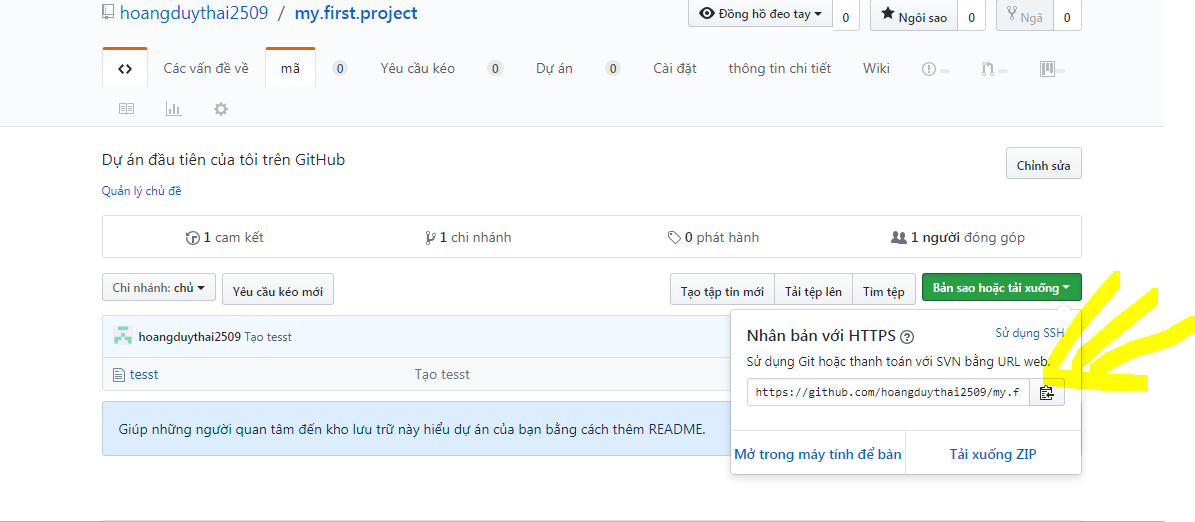
-Việc đầu tiên bạn nên làm khi cấu hình Git là chỉ định tên tài khoản và địa chỉ e-mail. Điều này rất quan trọng vì mỗi Git sẽ sử dụng chúng cho mỗi lần commit, những thông tin này được gắn bất di bất dịch vào các commit:

*$ git config --global user.name "hoangduythai2509”*

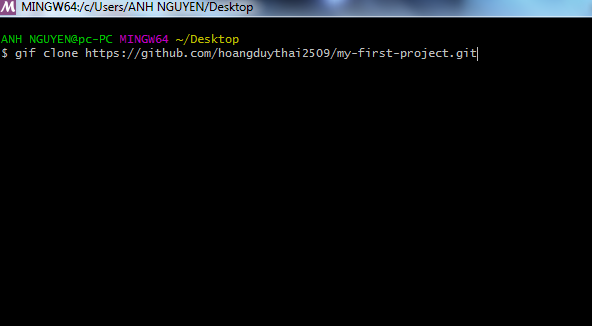
*$ git config --global user.email thaiduyhoang25@example.com*

-Git sẽ sử dụng các thông tin đó cho tất cả những gì bạn làm trên hệ thống. Nếu bạn muốn sử dụng tên và địa chỉ e-mail khác cho một dự án riêng biệt nào đó, bạn có thể chạy lại lệnh trên không sử dụng **--global** trên dự án đó.

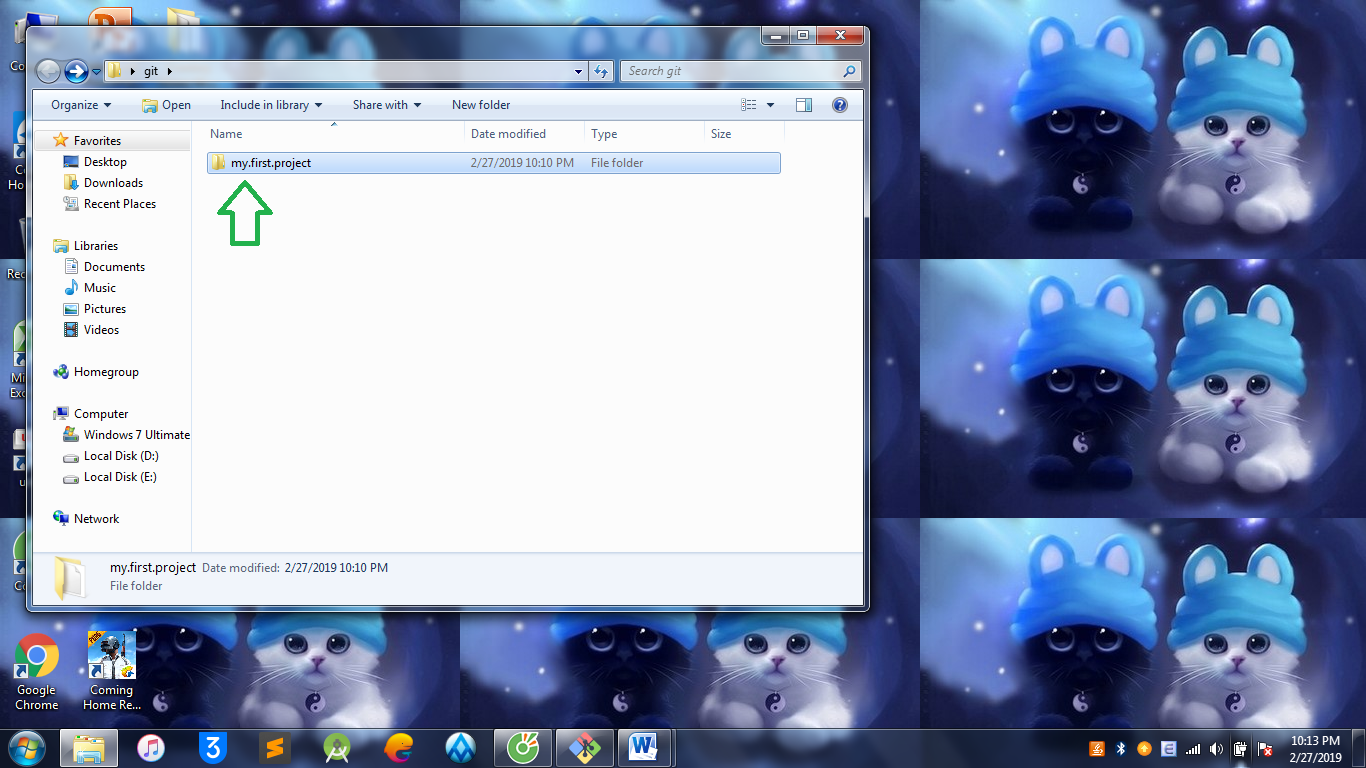
-Lựa chọn một **Repository đã tạo trên GitHub để clone**(Tạo bản sao chép) thành một bản ở máy tính địa phương của bạn.

****

-Tạo bản sao về máy của bạn:

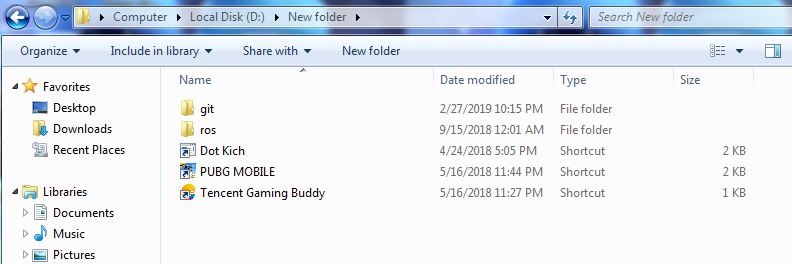
****

-Kết quả :

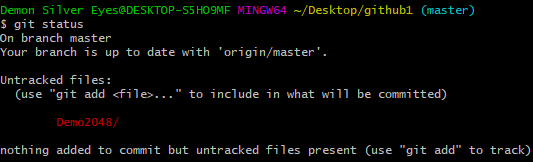
****

1. Các thao tác và lệnh của **Git**

-Thử copy vào một vài file dữ liệu của bạn :

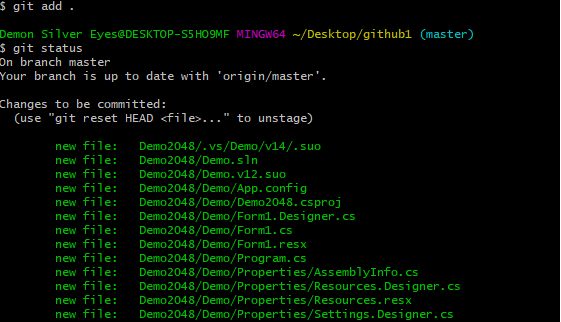
****

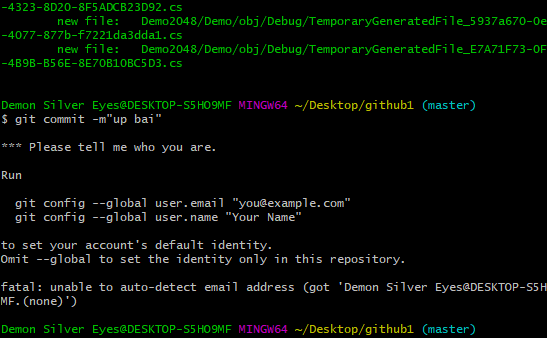
**-Sử dụng lệnh $git status để kiểm tra xem có thay đổi gì trên máy chủ**

****

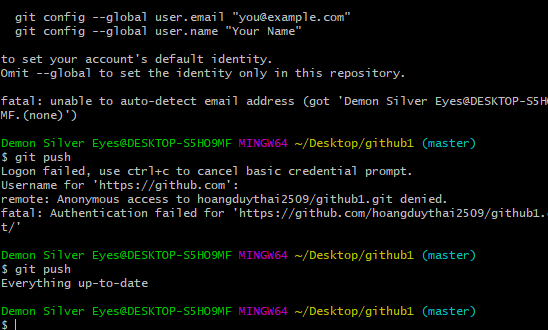
-Sử dụng lệnh $git add để add dữ liệu

-Nhập thông tin ghi chú (Comment) và dùng lệnh **$git commit** **để ghi lại sự thay đổi**.

****

****

**-Cuối cùng sử dụng lệnh $git push để đẩy dữ liệu lên:**

****

*(dữ liệu sau khi đẩy thành công)*

=>Các file dữ liệu bạn và thành viên có thể nhìn thấy trên **Server**

-Khi các thành viên khác thao tác thì cần dùng lệnh **$git pull** để cập nhật **repository** cục bộ của bạn và commit mới nhất.

# **Chương II: Phềm mềm hoàn chỉnh khi sử dụng Git**

1. **Chức năng chính của hệ thống game:**

Hệ thống gồm các chức năng đơn giản và cần thiết của một phần mềm game:

1.Chiều dài khoảng cách các ô trong game 2048

2.Tạo các ô ngẫu nhiên

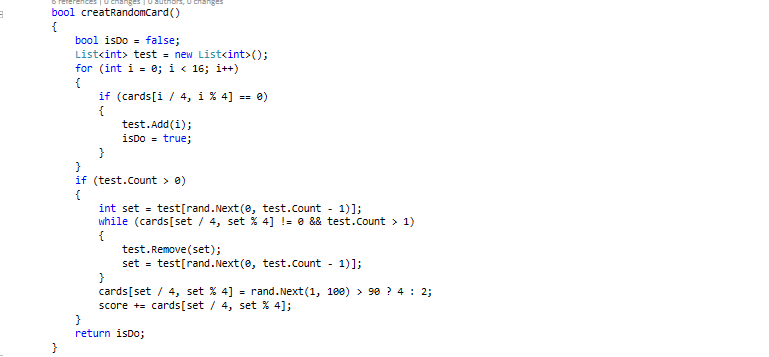
3.Tạo sự kiện con số trong game 2048

4.Thao tác các nút điều khiển trong game 2048

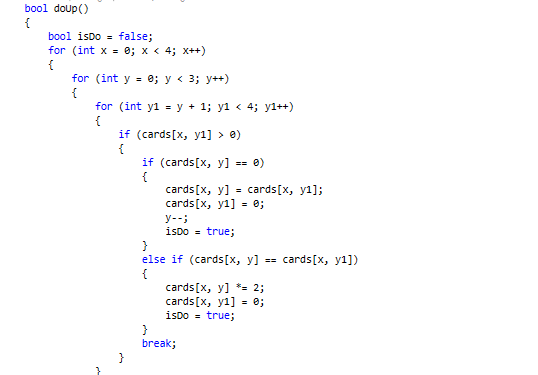
**1.Chiều dài khoảng cách các ô trong game 2048**

****

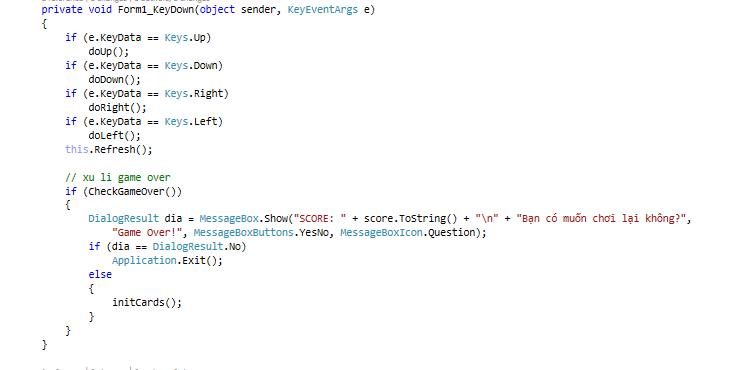
**2.Tạo các ô ngẫu nhiên**

****

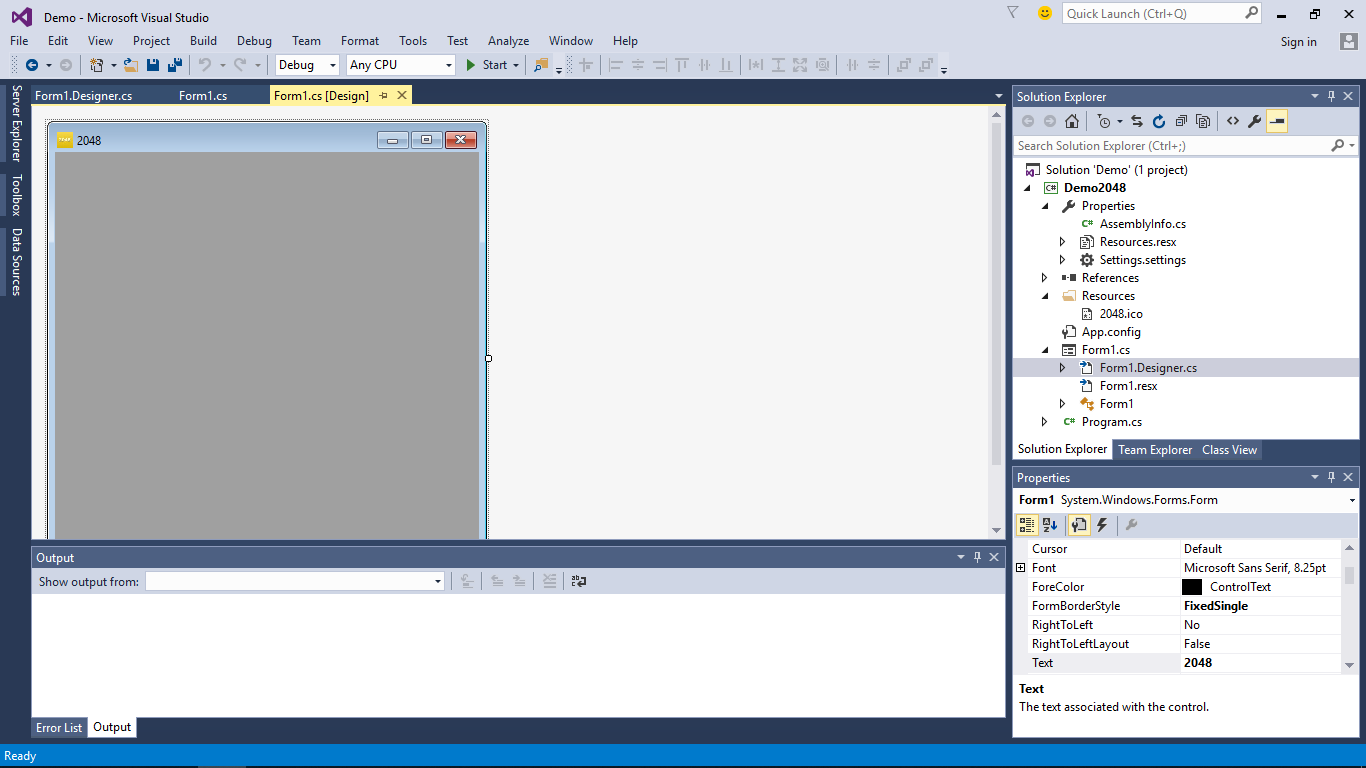
**3.Tạo sự kiện con số trong game 2048**

****

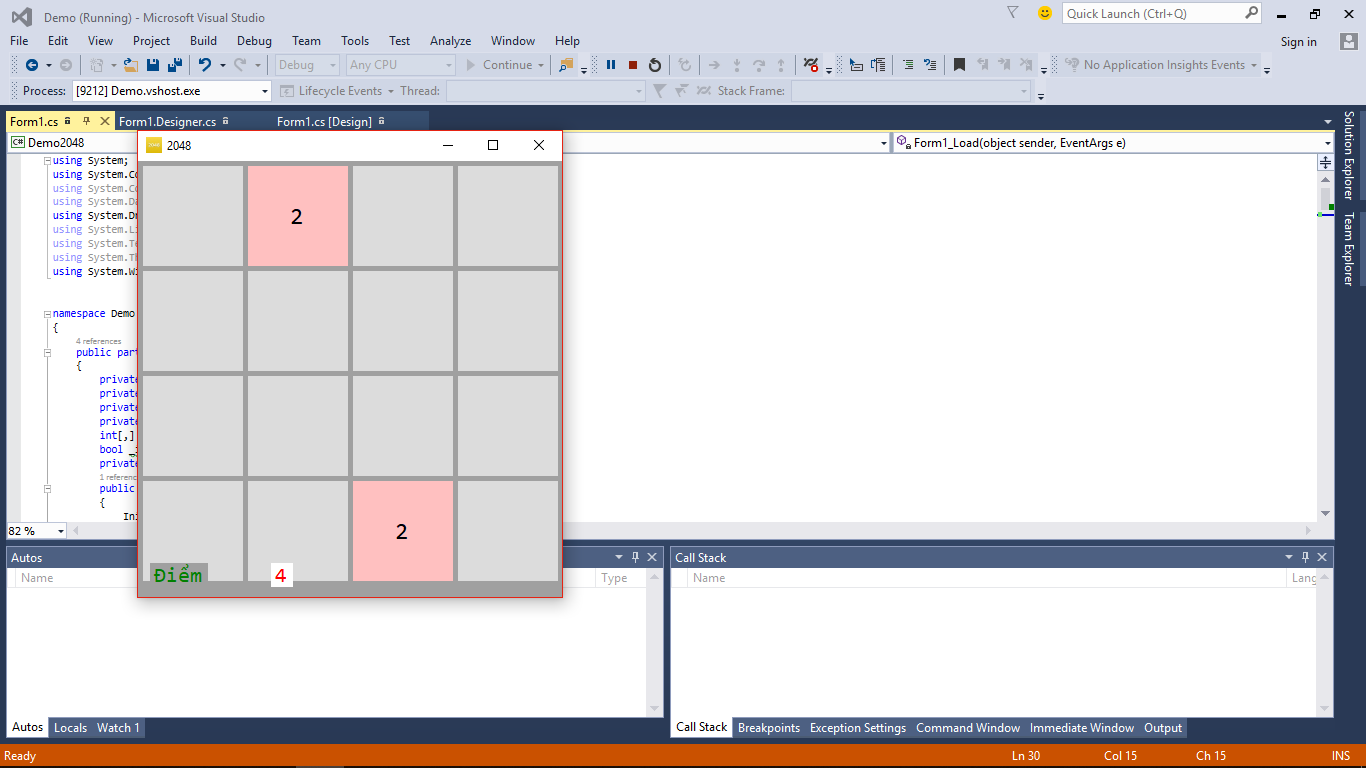
**4.Thao tác các nút điều khiển trong game 2048**

****

**Design game 2048:**

****

**Kết quả:**

****

# **Chương III: Phân công nhóm**

**DANH MỤC CÔNG VIỆC CẦN THỰC HIỆN**

* + - 1. **Lê Hoàng Cương: Lên ý tưởng,code,Sửa code toàn nhóm.**
      2. **Trần Cao Nguyễn: Sử lý các thao tác điều khiển và sử lý game over**
      3. **Hoàng Duy Thái: Viết báo cáo và github**
      4. **Bùi Nhật Hưng : Fix vÀ hoàn thành thêm**