VG101 Lab Manual

Lab 3

Instructor: Dr. Yifei ZHU
TA: Hangrui CAO
TA: Qinhang WU
TA: Muchen XU

Table of Contents

Workflow

Content	Approx. Time
Git Tutorial	25 mins
Warmup: Basic Syntax Exercise	30 mins
Break	10 mins
Lab Design : Three-body Animation	35 mins
Break	10 mins
Practical Exercise	70 mins

Git

Git is a distributed version-control system for tracking changes in source code during software development.

It is designed for coordinating work among programmers, but it can be used to track changes in any set of files. Its goals include speed, data integrity, and support for distributed, non-linear workflows.

Basic Topics

- Working Directory
  - the folder where the repository locates
- Index (Stage)

    - a staging area between your working directory and your repository.
    - build up a set of changes that you want to commit together.
    - what is committed is those currently in the index, not what is in your working directory.
- Repository
    - the whole project
    - all history is recorded
- Remote (Repository)
    - remote repository (like GitHub)
    - should be synchronized with your local repository

Usage

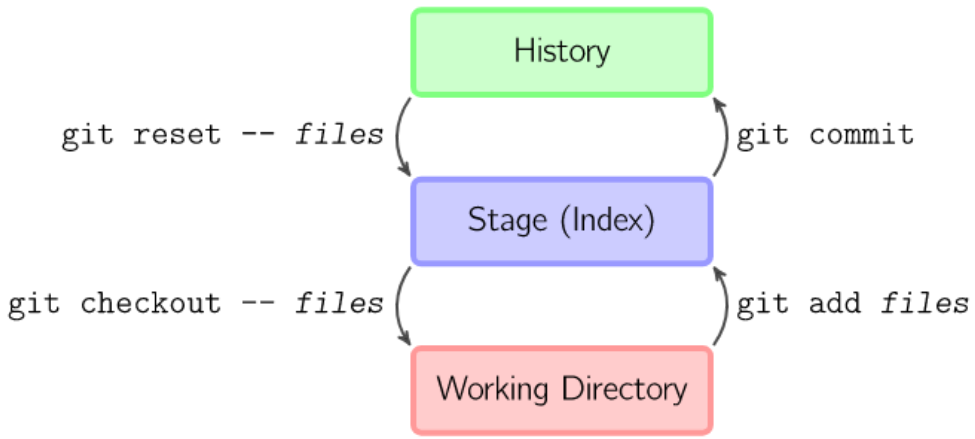
Environment and Git Installation

See Environment Setup and Git Installation. And you will also need a SSH Key if you want to push your local repository to GitHub.

Recommended solution (in CLI):

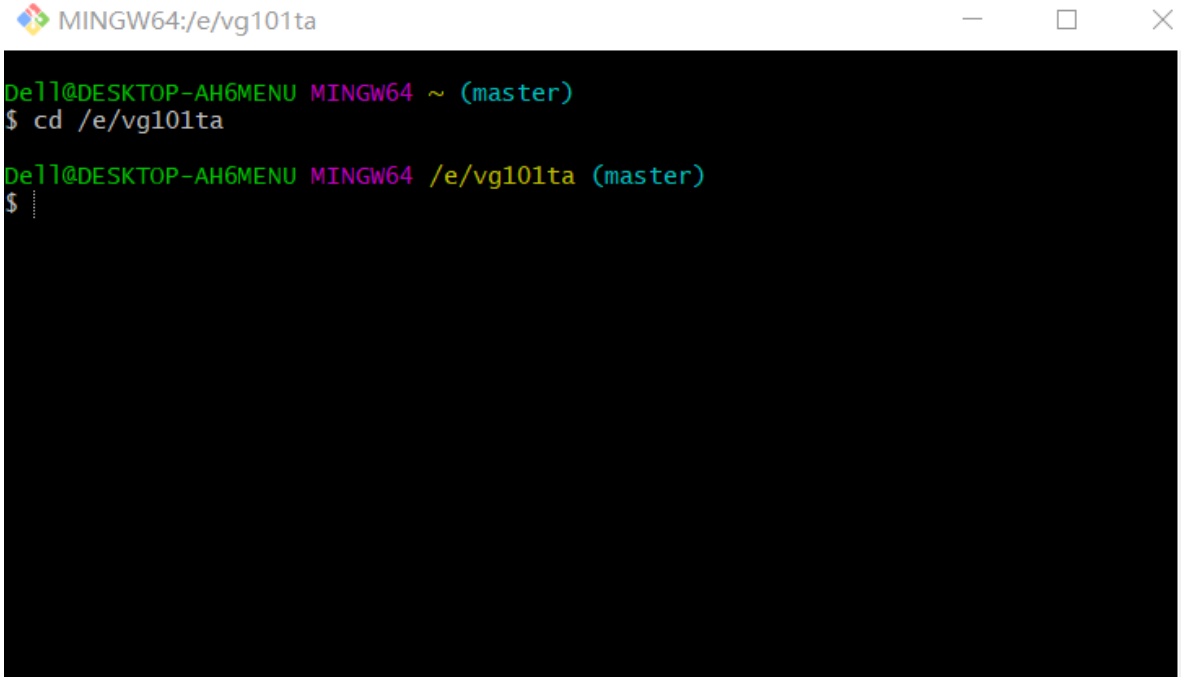
- Windows10
    - WSL + install git from apt
- MacOS
    - install git from brew
- Otherwise
    - gitbash

Version Control



Init

After installation, there will be a terminal and you can use for all following commands:



The simple way of init: Use the git command to set up a repository.

```
git init myrepo
```

Add

If you want to add files to version control, using the add commands:

```
git add filename
```

Examples:

```
git add mycode.c
git add README
```

You can use `git status -s` to see whether the files have been added or not.

Commit

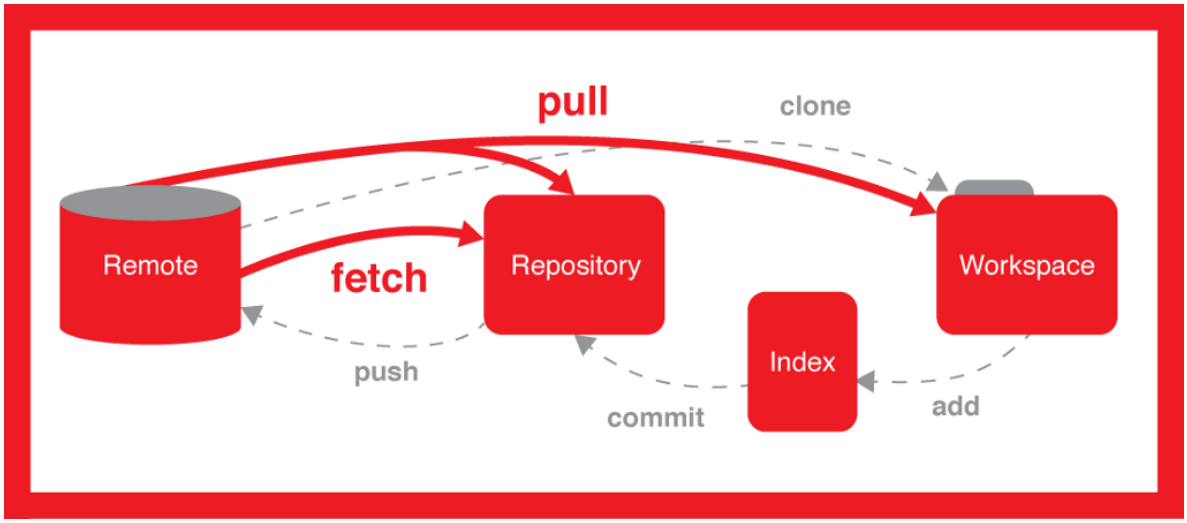
Commit is used to save the changes to local repositories. The format is:

```
git commit -m 'message'
```

Reset

Reset is used when you want to reset the changes you made.

Remote Repository



Using commands of remote, you can check the branches existed:

```
git remote -v
```

Clone

Clone is used when you want to copy a git repository to local. The basic command format:

```
git clone [url]
```

If you need to cooperate with others or copy one project from github to see the code, you can use clone.

For more information, you can look up git commands on the official website: https://git-scm.com/docs/git-commit

SSH-KEY

Before you want to push something on github (ssh url), you need a ssh key. The procedures to set up a ssh key is:

```
ssh-keygen -t rsa -C "caohangrui@sjtu.edu.cn"
```

-t refers to the types of code and the automatic type is rsa. -C refers to the annotation, like the email or something you like. The email is the email that you use to sign up for Github.

Then, you can add SSH-key to the ssh-agent:

```
# Make sure the agent is running(if you use git shell, it is running)
$ eval $(ssh-agent -s)
#add the key(when the name of the key files is id_rsa)
$ ssh-add ~/.ssh/id_rsa
```

At last, add the key to github. Copy the id_rsa.pub files(or the name of the files you set up). You can use following commands:

```
$ clip < ~/.ssh/id_rsa.pub
```

Click the right upper corner of your account, and find "settings->SSH and GPG keys" and paste the information you copy.

Push

Push is used to upload local repository content to a remote repository. The basic command format:

```
git push <remote> <branch>
```

Practice:

Clone one repository you set in last week and add some files in it. Test use of push and other commands.

Lab Design

In this section, we will use a comprehensive problems to illustrate MATLAB plotting, by realizing a three-body simulation in MATLAB.

Three-body Animation

In classical mechanics, the **three-body problem** is the problem of taking the initial positions and velocities of three point masses and solving for their subsequent motion according to Newton's laws of motion and Newton's law of universal gravitation. Indeed, no general closed-form solution exists, as the resulting dynamical system is chaotic for most initial conditions, and numerical methods are generally required.

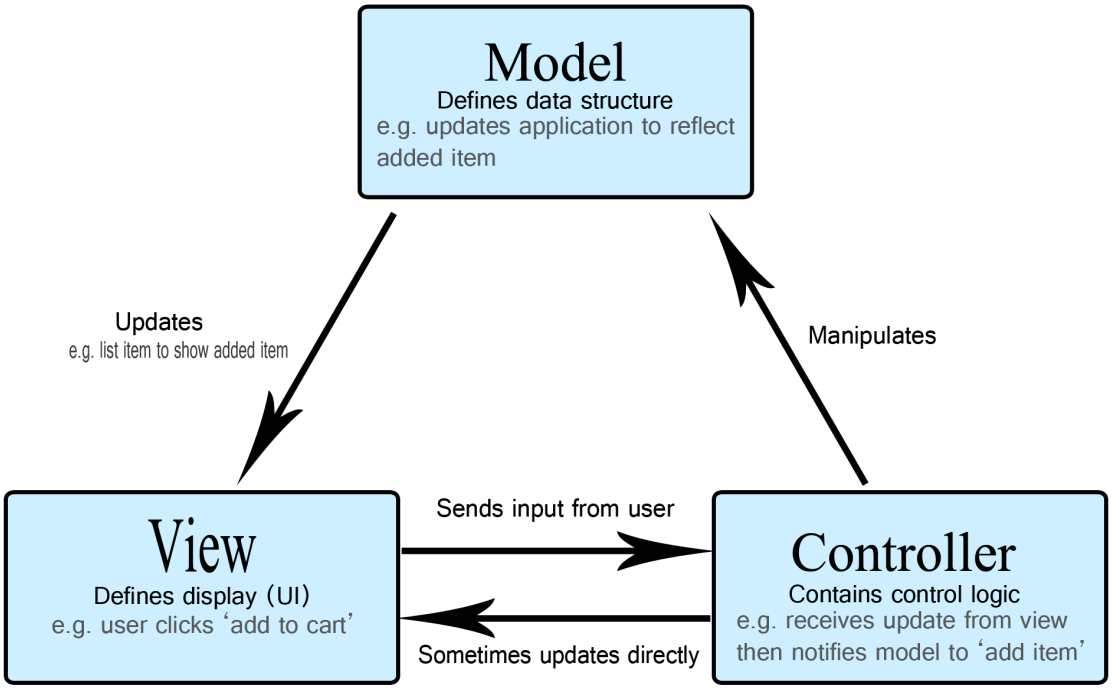
In this section, we will simulate the three-body problem in **2D** while using **MVC** to structure the program.

Tags: `#plot` `#function` `#MVC structure`

MVC: Model - View - Controller

Model-View-Controller, denoted by MVC, is a software design pattern for user interface development. It divides the related program logic into three interconnected blocks so as to enable simultaneous development and encourage code reuse. Besides, future development or modification is easier to applied.

The **model** is responsible for managing the data of the application. It receives user input from the controller. The **view** means presentation of the model in a particular format. The **controller** responds to the user input and performs interactions on the data model objects. It receives the input, optionally validates it and then passes the input to the model. Their relationship can be illustrated as follow:



Three-body Animation in MVC

You may use the code skeleton provided below to complete the Three-body Animation:

View

```
% myView.m

function myView(x)
    clf;
    axis equal
    hold on;
    % TODO: draw three circles with different color
    hold off;
end

function drawCircle(coor,radius,color)
    % TODO: draw a circle with coordinate radius and color
end
```

Perform a unit test after programming. You may try `view([0 0;1 1;2 2])` to check whether your program runs as expected.

Model

The equation below might be useful:

$$F = G \cdot \frac{m_1 \cdot m_2 \cdot (r_1 - r_2)}{|(r_1 - r_2)|^3}$$

```
% myModel.m

```

```
function [x,v]=myModel(x,v,t)
    x=x+v*t;
    v=v+acc(x)*t;
end

function acceleration=acc(x)
    % TODO: calculate the acceleration due to gravity
end

function force=F(x1,x2)
    % TODO: calculate the gravity force from coordinate x2 to x1
end
```

Perform a unit test after programming.

Controller

```
% myController.m

t=0.01;         % time interval
x=rand(3,2)*100;% coordinate
v=rand(3,2)*40; % velocity
while 1
    myView(x);
    [x,v]=myModel(x,v,t);
    pause(0.01)
end
```

Main program.

Grading Rubric

Criteria	Weight	Available Time	Due Time	Entry
Attendance	30%	**4:00pm**, May.29	**11:59pm**, May.29	Canvas Assignment
In-lab quiz	70%	**9:00pm**, May.29	**11:59pm**, May.29	Canvas Quiz
Git bonus	20%	**9:00pm**, May.29	**11:59pm**, May.29	Canvas Quiz

- For the attendance score, you need to submit your code for **exercises** in the **worksheet** on Canvas. We won't judge the correctness of your code. You'll earn full credits as long as you've tried these exercises.
- The attendance, in-lab quiz, and the Git bonus will be ceiled together if greater than 100%.