

---

# **Creative Software Design**

## **1 - Lab1 - Reference - Vim Advanced**

Yoonsang Lee

Fall 2023

# Shell settings for convenient vim use - 1

- set completion-ignore-case on
  - bash (shell) has an autocompletion function (by pressing <tab> key), and it is convenient to make this autocompletion case insensitive.
  - Just add "set completion-ignore-case on" to your .inputrc file. You can do it like this:

(Shell – home directory)

```
echo 'set completion-ignore-case On' >> ~/.inputrc
```

- Close and reopen the terminal to see it apply.

# Shell settings for convenient vim use - 2

- `stty -ixon`
  - This setting allows you to use the shortcut keys used with Ctrl in vim.
  - Just add "`stty -ixon`" to your `.bashrc`. You can do it like this:

(Shell – home directory)

```
echo 'stty -ixon' >> ~/.bashrc
```

- Close and reopen the terminal to see it apply.

# Install vim-gtk

---

- The vim version installed by default on Ubuntu is a vim-tiny version without many useful features.
- Run the following command to install vim-gtk, which contains most of the functionality.

(Shell)

```
sudo apt-get install vim-gtk
```

# Vim configuration - .vimrc

---

- You can change various settings of Vim through **.vimrc** file in your home directory.
- **.vimrc** is written in Vim's own scripting language called vim script.
  - Comment starts with “
  - For more detail, refer <http://learnvimscriptthehardway.stevelosh.com/>
- Since there is no **.vimrc** file at first, let's start editing the **.vimrc** file with the following command in your home directory:

(Shell)

```
vi .vimrc
```

# Recommended minimal settings for .vimrc

```
.vimrc
syntax on      "use syntax highlighting
filetype plugin indent on  "use auto-indentation

set expandtab   "use spaces instead of a tab
set tabstop=4  "number of spaces for a tab
set shiftwidth=4 "number of spaces for each step of indent (e.g. when using '>' or '<')

set nowrap     "stop line breaking
set clipboard=unnamedplus "use system clipboard (e.g. when using 'yy')
set ignorecase "case-insensitive search
set incsearch  "use incremental search

"disable automatic comment insertion
autocmd FileType * setlocal formatoptions-=c formatoptions-=r formatoptions-=o
```

- After updating .vimrc, enter `:so%` to apply the updated settings. Try to modify and apply the above file.
- The above file contains minimal items, so feel free to modify / add / remove any settings you want.

# Vim Visual Mode

---

## Marking text (visual mode) <https://vim.rtorr.com/>

- v – start visual mode, mark lines, then do a command (like y–yank)
- V – start linewise visual mode
- Ctrl + v – start visual block mode
  
- In the .vimrc file, mark a block and copy (y), paste (p), and so on.

# Vim Windows

---

## Working with multiple files

<https://vim.rtorr.com/>

- Ctrl + ws – split window
  - Ctrl + ww – switch windows
  - Ctrl + wq – quit a window
  - Ctrl + wv – split window vertically
  - Ctrl + wh – move cursor to the left window (vertical split)
  - Ctrl + wl – move cursor to the right window (vertical split)
  - Ctrl + wj – move cursor to the window below (horizontal split)
  - Ctrl + wk – move cursor to the window above (horizontal split)
- Split your .vimrc file into multiple windows and move between them.



# Vim Plug-ins

---

- Many plugins are available to extend Vim's functionality.
  - [https://vim.sourceforge.io/scripts/script\\_search\\_results.php?keywords=&script\\_type=&order\\_by=rating&direction=descending&search=search](https://vim.sourceforge.io/scripts/script_search_results.php?keywords=&script_type=&order_by=rating&direction=descending&search=search)
- You can download and use plug-in files directly, but using "vim plugin managers" is much more convenient to install and manage plug-ins.

# Install vim-plug

- One of the most popular vim plugin manager
- <https://github.com/junegunn/vim-plug>
- To install it, you need to install git first and then configure it initially.
- Replace the gray part below with your information.

(Shell)

```
sudo apt-get install git
git config --global user.name "Your Name"
git config --global user.email "you@example.com"
```

- Install vim-plug following the instructions on the page above.

# Using vim-plug

- Let's install two most useful plugins.
  - Add the following lines to .vimrc and save it, and type :so% and :PlugInstall

**.vimrc**

```
call plug#begin()  
Plug 'scrooloose/nerdtree'  
Plug 'scrooloose/nerdcommenter'  
call plug#end()
```

- To install other plugins, you can add them like Plug '<Github account name> / <Github project name>'.
  - Most Vim plugins are maintained on Github.
  - e.g.) NERDTree <https://github.com/scrooloose/nerdtree>

# NERDTree

---

- Plugin that allows directory browsing on Vim.
- How to run: Enter `:NERDTree` in Vim and press Enter
- Go to the NERDTree window and press `?` to test its functions.

# NERD Commenter

- Plugin that provides comment / uncomment function for various programming languages.
- Let's add a new shortcut by adding the following to `.vimrc`:

## `.vimrc`

```
let mapleader=","          " change <leader> key
let NERDCreateDefaultMappings = 0      "disable default mapping
let NERDCommentWholeLinesInVMode = 1 "always comment whole line
map <Leader>c <plug>NERDCommenterComment
map <Leader>x <plug>NERDCommenterUncomment
```

- After reloading `.vimrc` by typing `:so%`,
- `,c` : comment
- `,x` : uncomment

# Vim colorscheme

```
brown
#define UNICODE
#include <windows.h>

int main(int argc, char **argv)
{
    int speed1 = 0, speed2 = 0, speed = 0;
    printf("Set Mouse Speed by Maverick\n");

    SystemParametersInfo(SPI_GETMOUSESPEED, 0,
        printf("Current speed: %2d\n", speed);

    if (argc == 1) return 0;
    if (argc >= 2) sscanf(argv[1], "%d", &spe
    if (argc >= 3) sscanf(argv[2], "%d", &spe

    if (argc == 2)
    { // set speed to fixed value
        speed = speed1;
    }
    else if (argc == 3)
    { // alternate between two speed, oth

camo
#define UNICODE
#include <windows.h>

int main(int argc, char **argv)
{
    int speed1 = 0, speed2 = 0, speed = 0;
    printf("Set Mouse Speed by Maverick\n");

    SystemParametersInfo(SPI_GETMOUSESPEED, 0,
        printf("Current speed: %2d\n", speed);

    if (argc == 1) return 0;
    if (argc >= 2) sscanf(argv[1], "%d", &spe
    if (argc >= 3) sscanf(argv[2], "%d", &spe

    if (argc == 2)
    { // set speed to fixed value
        speed = speed1;
    }
    else if (argc == 3)
    { // alternate between two speed, oth

campfire
#define UNICODE
#include <windows.h>

int main(int argc, char **argv)
{
    int speed1 = 0, speed2 = 0, speed = 0;
    printf("Set Mouse Speed by Maverick\n");

    SystemParametersInfo(SPI_GETMOUSESPEED, 0,
        printf("Current speed: %2d\n", speed);

    if (argc == 1) return 0;
    if (argc >= 2) sscanf(argv[1], "%d", &spe
    if (argc >= 3) sscanf(argv[2], "%d", &spe

    if (argc == 2)
    { // set speed to fixed value
        speed = speed1;
    }
    else if (argc == 3)
    { // alternate between two speed, oth
```

```
candy
#define UNICODE
#include <windows.h>

int main(int argc, char **argv)
{
    int speed1 = 0, speed2 = 0, speed = 0;
    printf("Set Mouse Speed by Maverick\n");

    SystemParametersInfo(SPI_GETMOUSESPEED, 0,
        printf("Current speed: %2d\n", speed);

    if (argc == 1) return 0;
    if (argc >= 2) sscanf(argv[1], "%d", &spe
    if (argc >= 3) sscanf(argv[2], "%d", &spe

    if (argc == 2)
    { // set speed to fixed value
        speed = speed1;
    }
    else if (argc == 3)
    { // alternate between two speed, oth

candycode
#define UNICODE
#include <windows.h>

int main(int argc, char **argv)
{
    int speed1 = 0, speed2 = 0, speed = 0;
    printf("Set Mouse Speed by Maverick\n");

    SystemParametersInfo(SPI_GETMOUSESPEED, 0,
        printf("Current speed: %2d\n", speed);

    if (argc == 1) return 0;
    if (argc >= 2) sscanf(argv[1], "%d", &spe
    if (argc >= 3) sscanf(argv[2], "%d", &spe

    if (argc == 2)
    { // set speed to fixed value
        speed = speed1;
    }
    else if (argc == 3)
    { // alternate between two speed, oth

caramel
#define UNICODE
#include <windows.h>

int main(int argc, char **argv)
{
    int speed1 = 0, speed2 = 0, speed = 0;
    printf("Set Mouse Speed by Maverick\n");

    SystemParametersInfo(SPI_GETMOUSESPEED, 0,
        printf("Current speed: %2d\n", speed);

    if (argc == 1) return 0;
    if (argc >= 2) sscanf(argv[1], "%d", &spe
    if (argc >= 3) sscanf(argv[2], "%d", &spe

    if (argc == 2)
    { // set speed to fixed value
        speed = speed1;
    }
    else if (argc == 3)
    { // alternate between two speed, oth
```

```
carvedwood
#define UNICODE
#include <windows.h>

int main(int argc, char **argv)
{
    int speed1 = 0, speed2 = 0, speed = 0;
    printf("Set Mouse Speed by Maverick\n");

    SystemParametersInfo(SPI_GETMOUSESPEED, 0,
        printf("Current speed: %2d\n", speed);

    if (argc == 1) return 0;
    if (argc >= 2) sscanf(argv[1], "%d", &spe
    if (argc >= 3) sscanf(argv[2], "%d", &spe

    if (argc == 2)
    { // set speed to fixed value
        speed = speed1;
    }
    else if (argc == 3)
    { // alternate between two speed, oth

carvedwoodcool
#define UNICODE
#include <windows.h>

int main(int argc, char **argv)
{
    int speed1 = 0, speed2 = 0, speed = 0;
    printf("Set Mouse Speed by Maverick\n");

    SystemParametersInfo(SPI_GETMOUSESPEED, 0,
        printf("Current speed: %2d\n", speed);

    if (argc == 1) return 0;
    if (argc >= 2) sscanf(argv[1], "%d", &spe
    if (argc >= 3) sscanf(argv[2], "%d", &spe

    if (argc == 2)
    { // set speed to fixed value
        speed = speed1;
    }
    else if (argc == 3)
    { // alternate between two speed, oth
```

```
50 end
51
52 #
53 def
54 p
55 end
56 end
57
58 modul
59 att
60
61 def
62 s
63 @
64 end
65

70 },
71
72 class
73 Name
74
75 public
76 Tree
77 :
78 {}
79
80 virt
81 fo
82
83
84
85
86
87
88
89
90
91
92
93
94 super
95 STDOUT.sync =
96 STDOUT.puts H
97 end
98
99 def show(kind,
100 @count += 1
101 perc = (100 *
102 bar_size = (3
103
104 bar = "#{ '='
105
106 if cmd.size >
107 cmd = "..#{
108 else
109 cmd = cmd.l
110
```

# Install Vim colorscheme

---

- A colorscheme can be installed like a plugin
- Let's install two most popular colorschemes.
  - Add the following bold lines to `.vimrc` and save it, and type `:so%` and `:PlugInstall`

## `.vimrc`

```
call plug#begin()  
Plug 'scrooloose/nerdtree'  
Plug 'scrooloose/nerdcommenter'  
Plug 'vim-scripts/xoria256.vim'  
Plug 'vim-scripts/peaksea'  
call plug#end()
```

# Apply Vim colorscheme

---

- Add the following lines to `.vimrc` to apply an installed colorscheme.
  - Add the following lines to `.vimrc` and save it, and type `:so%`
  - Apply another colorscheme by changing commented / uncommented lines.

```
.vimrc
```

```
colorscheme xoria256  
"colorscheme peaksea
```



# More Colorschemes?

---

- <http://vimcolors.com/>
- [https://vim.sourceforge.io/scripts/script\\_search\\_results.php?keywords=&script\\_type=color+scheme&order\\_by=rating&direction=descending&search=search](https://vim.sourceforge.io/scripts/script_search_results.php?keywords=&script_type=color+scheme&order_by=rating&direction=descending&search=search)
- Googling ‘vim colorscheme’
- Find the address of a colorscheme's Github project and add it between `plug#begin()` and `plug#end()`.