

Graphical User Shell (GUSH)

Alexander Biggs, Spencer Elliott, Ryan Fan, Nicholas Goh,
Peter Lu, Wei Ping (Allen) Hsiao, Kyle Zhou

Problem Space

- Shell is an integral part of programming

```
2. konstantin@Konstantins-MacBook-Air-2: ~/Workspace/travis (zsh)
[~/Workspace/travis]$ travis encrypt F00=bar [master] [1.9.3-p286]
Please add the following to your .travis.yml file:

  secure: "gFD9xGe40J0JKryHPZrzC3SbqPJ0Wd1e4voA6xPuci6HlvPUZKcmg0ttCnC\n0kkHfYnQr3trwKth42
Fd5MxAGFwETUGs/5sa1KAV5RteNc-jncXAFr+1v6utw\n/1GG3l6M/lSr+KZz+Z/pz87z+/bht53Rbo7WSDMSumI4RED
dNqU="

Pro Tip™: You can add it automatically by running with --add.
[~/Workspace/travis]$ cat secret.txt | travis encrypt --add [master] [1.9.3-p286]
[~/Workspace/travis]$ git status * [master] [1.9.3-p286]
# On branch master
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       modified:   .travis.yml
#
no changes added to commit (use "git add" and/or "git commit -a")
[~/Workspace/travis]$ [master] [1.9.3-p286]
```

Problem Space

- Command-line interfaces are unintuitive

```
[root@localhost ~]# ping -q fa.wikipedia.org
PING text.pmtpa.wikimedia.org (208.80.152.2) 56(84) bytes of data.
^C
--- text.pmtpa.wikimedia.org ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 540.528/540.528/540.528/0.000 ms
[root@localhost ~]# pwd
/root
[root@localhost ~]# cd /var
[root@localhost var]# ls -la
total 72
drwxr-xr-x. 18 root root 4096 Jul 30 22:43 .
drwxr-xr-x. 23 root root 4096 Sep 14 20:42 ..
drwxr-xr-x. 2 root root 4096 May 14 00:15 account
drwxr-xr-x. 11 root root 4096 Jul 31 22:26 cache
drwxr-xr-x. 3 root root 4096 May 18 16:03 db
drwxr-xr-x. 3 root root 4096 May 18 16:03 empty
drwxr-xr-x. 2 root root 4096 May 18 16:03 games
drwxrwx--T. 2 root gdm 4096 Jun 2 18:39 gdm
drwxr-xr-x. 38 root root 4096 May 18 16:03 lib
drwxr-xr-x. 2 root root 4096 May 18 16:03 local
lrwxrwxrwx. 1 root root 11 May 14 00:12 lock -> ../run/lock
drwxr-xr-x. 14 root root 4096 Sep 14 20:42 log
lrwxrwxrwx. 1 root root 10 Jul 30 22:43 mail -> spool/mail
drwxr-xr-x. 2 root root 4096 May 18 16:03 nis
drwxr-xr-x. 2 root root 4096 May 18 16:03 opt
drwxr-xr-x. 2 root root 4096 May 18 16:03 preserve
drwxr-xr-x. 2 root root 4096 Jul 1 22:11 report
lrwxrwxrwx. 1 root root 6 May 14 00:12 run -> ../run
drwxr-xr-x. 14 root root 4096 May 18 16:03 spool
drwxrwxrwt. 4 root root 4096 Sep 12 23:50 tmp
drwxr-xr-x. 2 root root 4096 May 18 16:03 yp
[root@localhost var]# yum search wiki
Loaded plugins: langpacks, presto, refresh-packagekit, remove-with-leaves
rpafusion-free-updates | 2.7 kB | 00:00
rpafusion-free-updates/primary_db | 206 kB | 00:04
rpafusion-nonfree-updates | 2.7 kB | 00:00
updates/metalink | 5.9 kB | 00:00
updates | 4.7 kB | 00:00
updates/primary_db 73% [=====] | 62 kB/s | 2.6 MB | 00:15 ETA
```

Similar Problems

- Similar issues found in programming with a new language

```
function aa_pc()
{
    [local M=$( free -o1m | sed "/Mem/Id, s/Mem:[^0-9]*\\([0-9]*\\)[^0-9]*\\([0-9]*\\)[^0-9]*\\([0-9]*\\).*/ \\
    ${GREP_COLOR}\\2MB${R}\\3MB/"; ); echo -e "$M `pwd`"
}

aa_grep_color(){ export GREP_COLOR=$(tput setaf $(aa_random_under 0)); }
aa_random_under(){ echo -en $(( ( $RANDOM % ${1:-$RANDOM} + 1 )); }
uptime1(){ sed '/.*, \\([0-9]*\\) users, load average: \\(.*\\)/Id; s//[ \\2, \\1 users ]/;q';

$SED -n '/!IQ2' <<< $- && \
{
    case ${TERM:-dummy} in xterm|screen|*color*)echo; ;; *) return; esac;
}

# hack for being able to source all the functions in this file from a script
MASTER_RESET=$(echo -e "rs1\nrs2\ninitc\nis2\ncnorm\nrmso\nsgr0" | tput -S`

# only for color-aware (some guessing) terminals, kick everyone else OUT
case ${TERM:-dummy} in xterm|screen|*color*)echo; ;; *) return; esac;

[[ -r /etc/DIR_COLORS ]] && eval `dircolors -b /etc/DIR_COLORS`

# changes the window title in terminals like putty or konsole
echo -ne "\033]0; ${USER}@${HOSTNAME} +${SHLVL} @${SSH_TTY}/${devV} - `uptime1` \007"

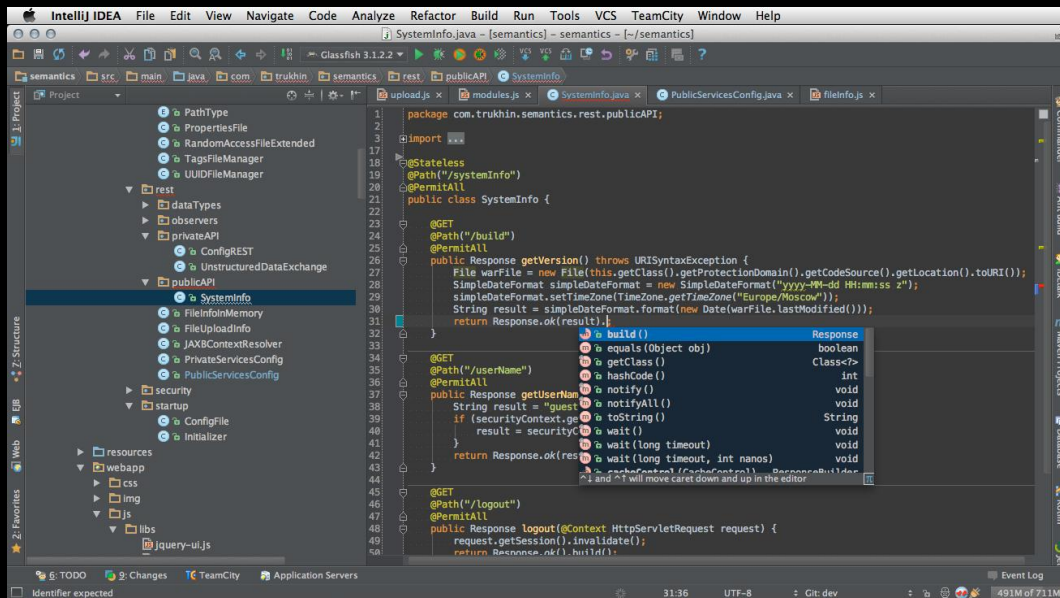
# show title, setup the colors, vars, funcs, settings
asetup_colors && ascript title
export PROMPT_COMMAND='history -a;aa_pc'
export PS1='\[\033[1;30m\][${$PPID} - \j:\[\033[1;30m\]]\[\033[0;36m\] \t\
\d \[\033[1;30m\][\[\033[1;34m\]\u@\h\[\033[1;30m\]:\[\033[0;37m\]${SSH_TTY} \
\[\033[0;32m\]+${SHLVL}\[\033[1;30m\]] \[\033[1;37m\]\w\[\033[0;37m\] \n$ '

/etc/profile.d/zzz_askapache-bash-profile.sh[+][RO] [unix] [SH] 99% [a=009] [h=09] [2411,6]

xterm-256color - 0-$ ROOT SCREEN 1$ unb 2$ bash 3$ im 4$* askapach
0 ROOT SCREEN unb 2 bash 3 im 4* askapach 21:55
```

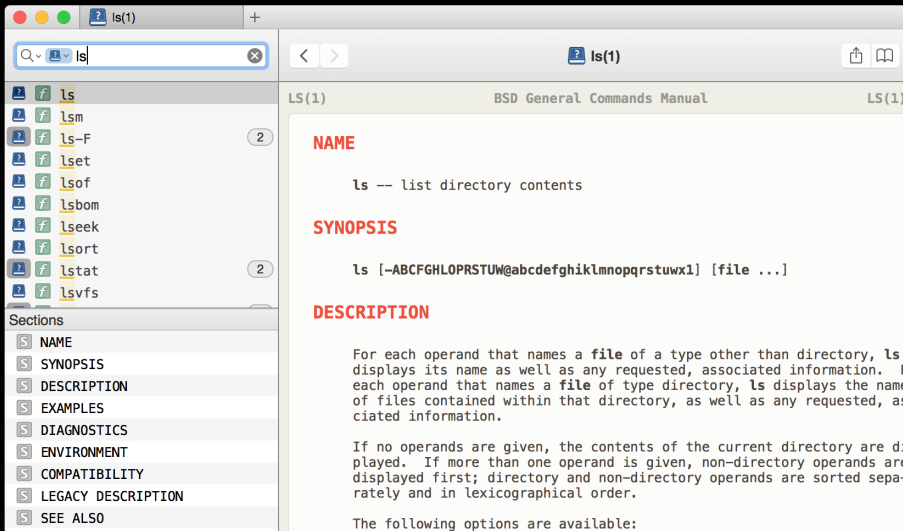
Solution

- Programming introduces the IDE



Proposed Solution

- Shell presented as an IDE
 - Graphical Interactive Shell
 - Accessible documentation



Design Principles

1. Should not restrict the user
2. Techniques learned should carry over to other shells 1:1
3. Designed with first-timers in mind
4. Useful to users of any experience level
5. Use visuals to explain whenever possible

Welcome to Graphical User Shell (GUSH).
Type `/?` to list basic commands or `ctrl+space` to perform
a search through documentation.

g3gohnic | wolf:~/CSC301/Presentation/

> /?

ls: Lists all files in the current directory
cd <dir>: Change current directory (navigate file system)
rm <file>: Delete a file (does not delete directories)
rm -r <dir>: Delete a directory and everything in it
mkdir <name>: Makes a new directory with specified name
nano <file>: Opens file in a text editor

g3gohnic | wolf:~/CSC301/Presentation/

> ls -

-ls	All files w/ size
-S	Sorted by size
-a	All file names

g3gohnic | wolf:~/CSC301/Presentation/

> cd ~/Files/testdir1/autocomplete_test.md

ls | list directory
contents

ls -a
List all files, even hidden

ls -A1
List all file names
(no extra info)

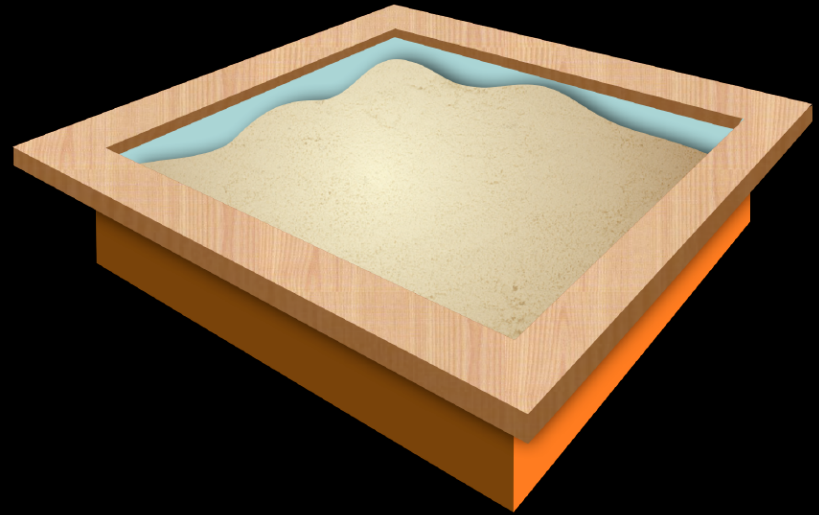
ls -ls
List all files with their
rights, groups, owner

ls -lh
List all files and display
the file size in a human
readable format

ls -r

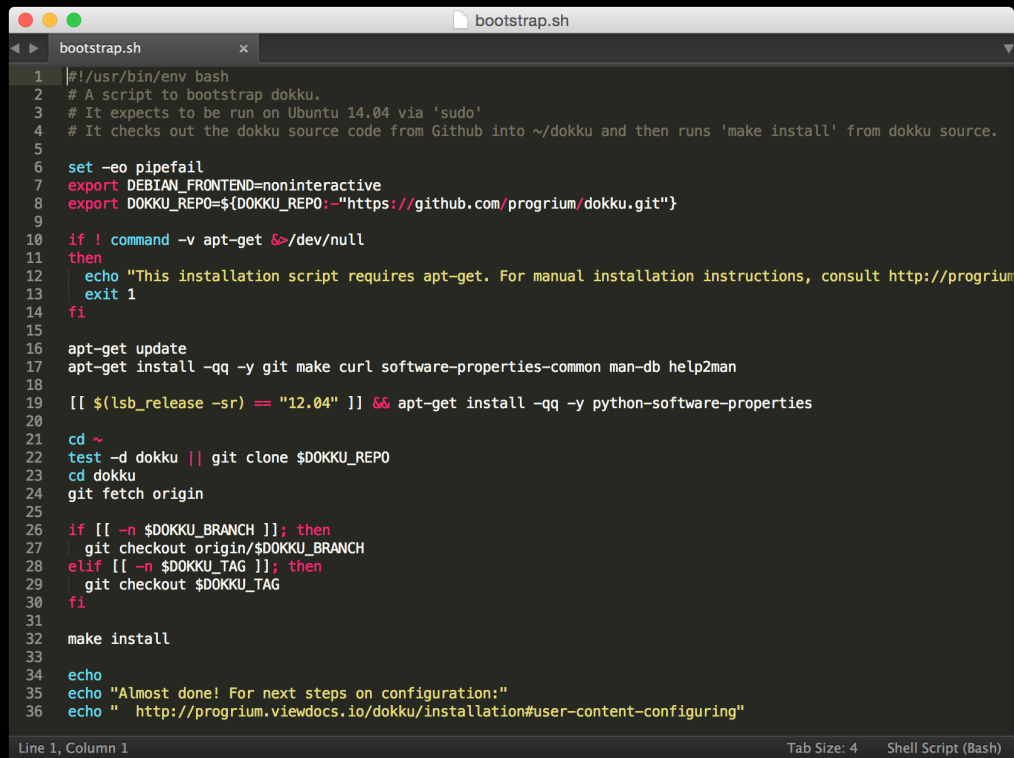
Extensibility

- Sandbox mode gives safe environment
- Facilitates user-made tutorials
- Testing ground for users
 - Features a pre-built file system



Extensibility

- In-app text editor
- Allows for quick creation of shell scripts
- ex. Sublime Text



```
bootstrap.sh
1 #!/usr/bin/env bash
2 # A script to bootstrap dokku.
3 # It expects to be run on Ubuntu 14.04 via 'sudo'
4 # It checks out the dokku source code from Github into ~/dokku and then runs 'make install' from dokku source.
5
6 set -eo pipefail
7 export DEBIAN_FRONTEND=noninteractive
8 export DOKKU_REPO=${DOKKU_REPO:-"https://github.com/progrium/dokku.git"}
9
10 if ! command -v apt-get &>/dev/null
11 then
12     echo "This installation script requires apt-get. For manual installation instructions, consult http://progrium
13     exit 1
14 fi
15
16 apt-get update
17 apt-get install -qq -y git make curl software-properties-common man-db help2man
18
19 [[ $(lsb_release -sr) == "12.04" ]] && apt-get install -qq -y python-software-properties
20
21 cd ~
22 test -d dokku || git clone $DOKKU_REPO
23 cd dokku
24 git fetch origin
25
26 if [[ -n $DOKKU_BRANCH ]]; then
27     git checkout origin/$DOKKU_BRANCH
28 elif [[ -n $DOKKU_TAG ]]; then
29     git checkout $DOKKU_TAG
30 fi
31
32 make install
33
34 echo
35 echo "Almost done! For next steps on configuration:"
36 echo "  http://progrium.viewdocs.io/dokku/installation#user-content-configuring"
```

Line 1, Column 1

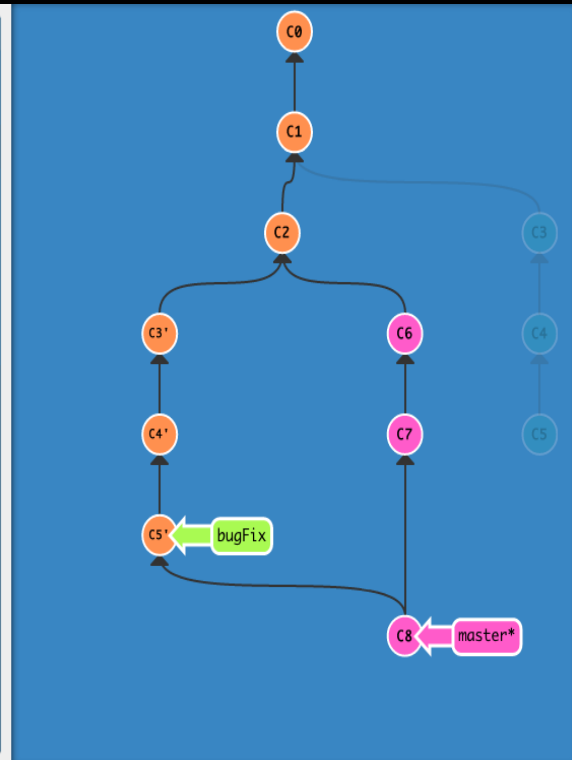
Tab Size: 4

Shell Script (Bash)

Extensibility

- In-app tutorials
 - Non-technical
 - Graphical
- Fully sandboxed
- ex. Git, Codecademy

```
Learn Git Branching
$ gc
$ git checkout HEAD~1
$ git commit
Warning!! Detached HEAD state
$ git checkout -b bugFix
$ gc
$ gc
$ git rebase master
$ git checkout master
$ gc
$ gc
$ git merge bugFix
```



Competitors / Inspiration

- Explain Shell (explainshell.com)
- Fish Shell (fishshell.com)
- tldr-pages (<https://github.com/tldr-pages>)

```
$ tldr tar
```

Archiving utility
Optional compression with gzip / bzip

- create an archive from files

```
tar cf target.tar file1 file2 file3
```

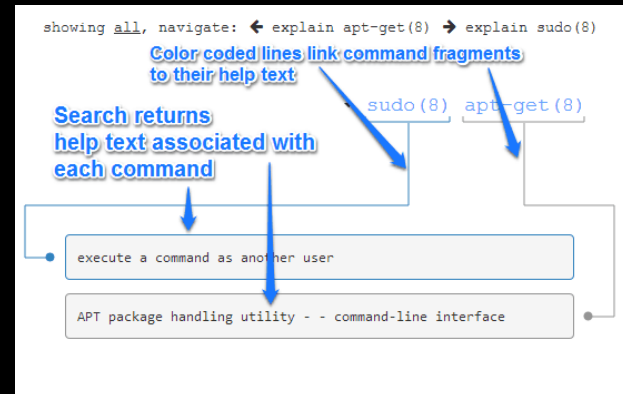
- create a gzipped archive

```
tar cfz target.tar.gz file1 file2 file3
```

- extract an archive in a target folder

```
tar xf source.tar -C folder
```

```
fish /Users/demo — fish — 80x24
demo > cat explanation.txt
Remember that long command to ssh into a host, or to check out that Subversion
project? You type it every day, but you've never gotten around to making that
function to do it for you. With the new fish, just type it once, and it will
remember it and suggest it for you again when it sees that you've started to
type it.
demo > ssh -l demo SomeLongHost@SomeLongDomainIAIwaysMisspell.com
```



Questions?