Code Development Environment

Experiences and Outlook

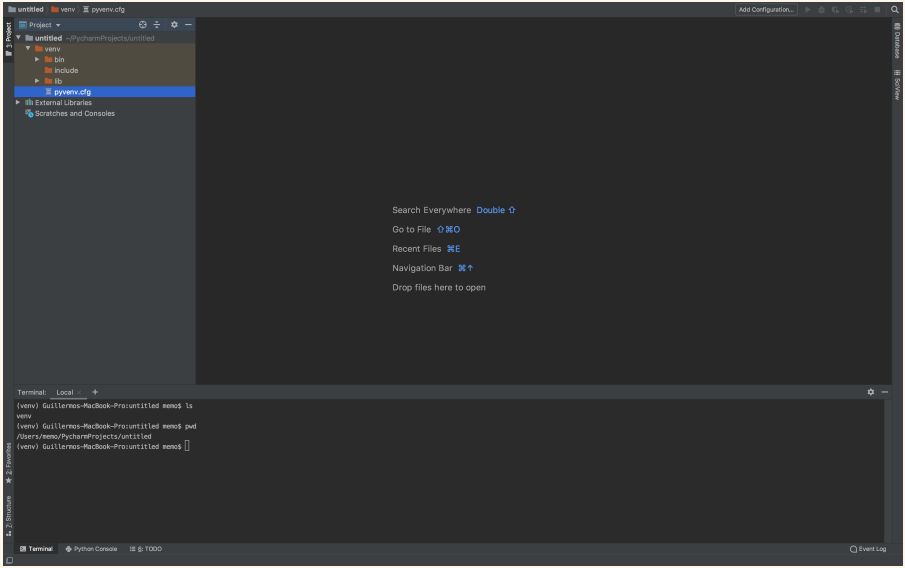
Guillermo Ibarra

Nuclear Engineering Research Seminar, February 4th, 2020

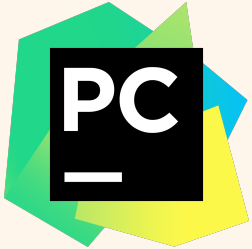
First Experiences

First Experiences

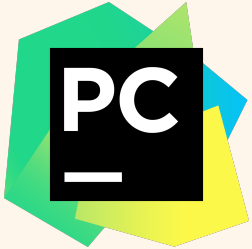
PyCharm IDE


untitled
venv
pyvenv.cfg
Add Configuration...
Project
untitled ~/PycharmProjects/untitled
venv
bin
include
lib
pyvenv.cfg
External Libraries
Scratches and Consoles
Search Everywhere Double ⇧
Go to File ⇧⌘O
Recent Files ⌘E
Navigation Bar ⌘↑
Drop files here to open
Terminal: Local
(venv) Guillermos-MacBook-Pro:untitled memo$ ls
venv
(venv) Guillermos-MacBook-Pro:untitled memo$ pwd
/Users/memo/PycharmProjects/untitled
(venv) Guillermos-MacBook-Pro:untitled memo$
Terminal
Python Console
6: TODO
Event Log
Database
SciView
1: Project
2: Favorites
Z: Structure

First Complete-ish Developmental Environment



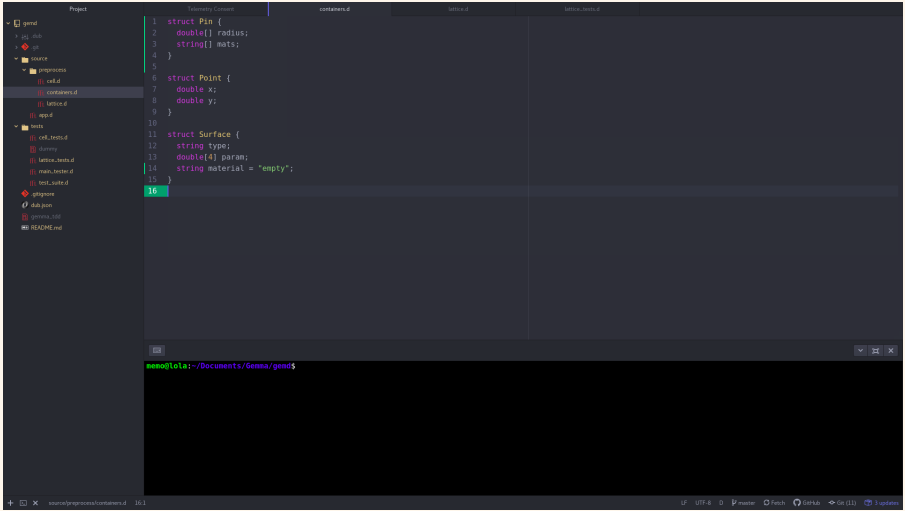
First Complete-ish Developmental Environment



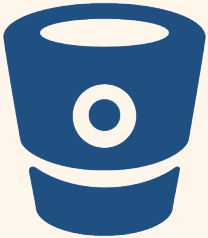
Upgrading the Workflow

Upgrading the Workflow

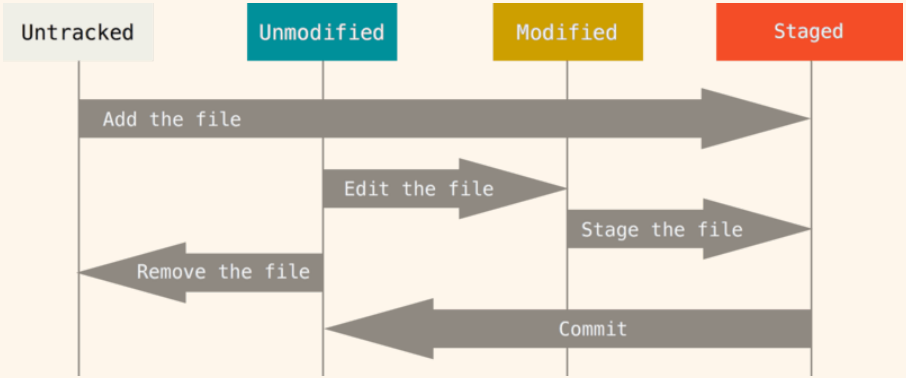
Atom IDE


Project
gemd
dub
git
source
preprocess
cell.d
containers.d
lattice.d
app.d
tests
cell_tests.d
gamma
lattice_tests.d
main_tester.d
test_suite.d
.gitignore
dub.json
gemma_log
README.md
Keyword Content
containers.d
lattice.d
lattice_tests.d
struct Pin {
double[] radius;
string[] mats;
}
struct Point {
double x;
double y;
}
struct Surface {
string type;
double[4] param;
string material = "empty";
}
memo@lola:~/Documents/Gemma/gemd$
source/preprocess/containers.d
16:1
UTF-8
master
Fetch
GitHub
Git (11)
3 updates

Cloud Repositories



Git Basics


Untracked
Unmodified
Modified
Staged
Add the file
Edit the file
Stage the file
Remove the file
Commit

Git Basics - Checking file status

```
$ echo 'My Project' > README
$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    README

nothing added to commit but untracked files present (use "git add" to track)
```

Git Basics - Tracking New Files

```
$ git add README
$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)

    new file:   README
```

Git Basics - Staging Modified Files

```
$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   README

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:   CONTRIBUTING.md
```

Git Basics - Staging Modified Files

```
$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   README

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:   CONTRIBUTING.md
```

```
$ git add CONTRIBUTING.md
$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   README
    modified:   CONTRIBUTING.md
```

Git Basics - Staging Modified Files...

```
$ vim CONTRIBUTING.md
$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   README
    modified:   CONTRIBUTING.md

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:   CONTRIBUTING.md
```

Git Basics - Staging Modified Files...

```
$ vim CONTRIBUTING.md
$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   README
    modified:   CONTRIBUTING.md

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:   CONTRIBUTING.md
```

```
$ git add CONTRIBUTING.md
$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   README
    modified:   CONTRIBUTING.md
```

Git Basics - Short Status

```
$ git status -s
 M README
MM Rakefile
A  lib/git.rb
M  lib/simplegit.rb
?? LICENSE.txt
```

Git Basics - Ignoring Files

```
$ cat .gitignore
*.[oa]
*~
```

Git Basics - Ignoring Files

```
$ cat .gitignore
*.[oa]
*~
```

```
$ cat .gitignore
# ignore all .a files
*.a

# but do track lib.a, even though you're ignoring .a files above
!lib.a

# only ignore the TODO file in the current directory, not subdir/TODO
/TODO

# ignore all files in any directory named build
build/

# ignore doc/notes.txt, but not doc/server/arch.txt
doc/*.txt

# ignore all .pdf files in the doc/ directory and any of its subdirectories
doc/**/*.pdf
```

Git Basics - Removing Files

```
$ rm PROJECTS.md
$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        deleted:    PROJECTS.md

no changes added to commit (use "git add" and/or "git commit -a")
```

Git Basics - Removing Files

```
$ rm PROJECTS.md
$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        deleted:    PROJECTS.md

no changes added to commit (use "git add" and/or "git commit -a")
```

```
$ git rm PROJECTS.md
rm 'PROJECTS.md'
$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    deleted:    PROJECTS.md
```

Git Basics - Committing

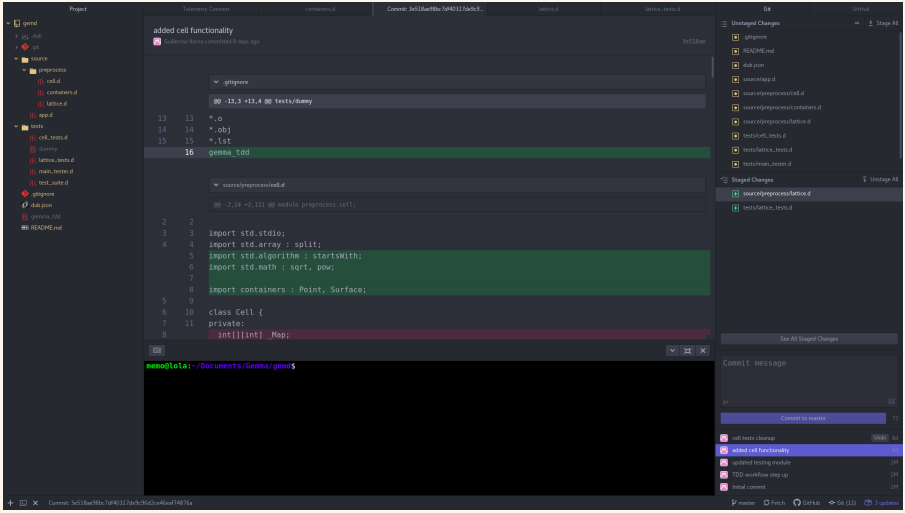
```
$ git commit
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
# On branch master
# Your branch is up-to-date with 'origin/master'.
#
# Changes to be committed:
#       new file:   README
#       modified:   CONTRIBUTING.md
#
~
~
~
".git/COMMIT_EDITMSG" 9L, 283C
```

Git Basics - Committing

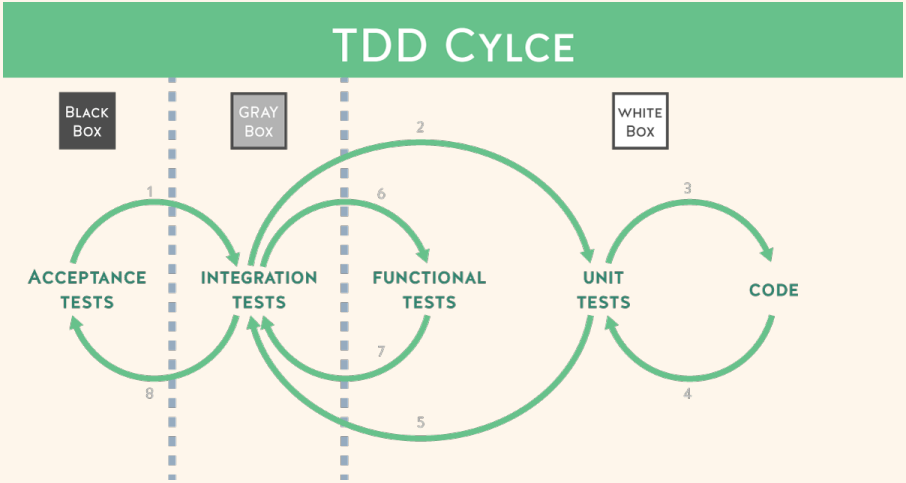
```
$ git commit
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
# On branch master
# Your branch is up-to-date with 'origin/master'.
#
# Changes to be committed:
#       new file:   README
#       modified:   CONTRIBUTING.md
#
~
~
~
".git/COMMIT_EDITMSG" 9L, 283C
```

```
$ git commit -m "Story 182: Fix benchmarks for speed"
[master 463dc4f] Story 182: Fix benchmarks for speed
 2 files changed, 2 insertions(+)
 create mode 100644 README
```

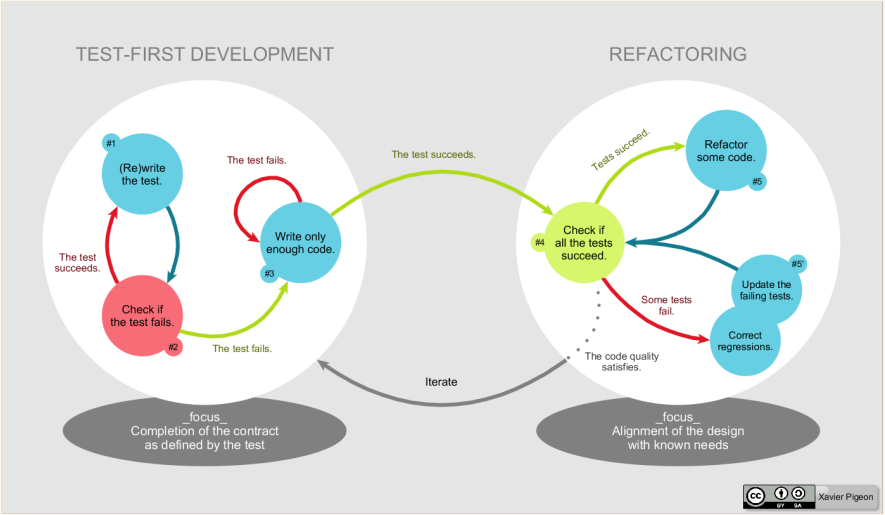
Git Addons with Atom



Tests? Tests!


TDD Cylce
Black Box
Gray Box
White Box
1
2
3
4
5
6
7
8
Acceptance Tests
Integration Tests
Functional Tests
Unit Tests
Code

Tests in Greater Detail



Testing with D

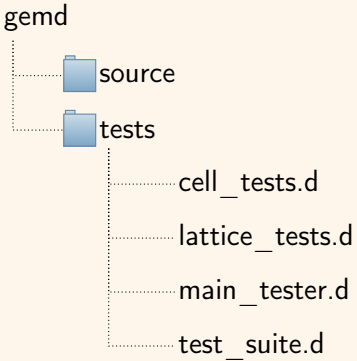
Using DUB package manager, dub.json file:

```
{
  "name": "gemma_tdd",
  "description": "2D neutron transport solver, TDD D implementation",

  "authors": [ "Guillermo", "Samuel Vargas", "Hirepan Palomares"],
  "copyright": "Copyright  2020, Guillermo, Samuel Vargas, Hirepan Palomares",
  "license": "proprietary",

  "configurations": [
    {
      "name": "application",
      "targetType": "executable",
    },
    {
      "name": "unittest",
      "targetType": "executable",
      "targetPath": "tests",
      "targetName": "dummy",
      "mainSourceFile": "tests/main_tester.d",
      "excludedSourceFiles": ["source/app.d"],
      "sourcePaths": ["tests/"],
      "importPaths": ["tests/"]
    }
  ]
}
```

Testing Structure



Compiling with DUB

```
$ dub build
Performing "debug" build using /usr/bin/dmd for x86_64.
gemma_tdd ~master: building configuration "application"...
Linking...
```

Compiling with DUB

```
$ dub build
Performing "debug" build using /usr/bin/dmd for x86_64.
gemma_tdd ~master: building configuration "application"...
Linking...
```

```
$ dub test
Running custom 'unittest' configuration.
Performing "unittest" build using /usr/bin/dmd for x86_64.
gemma_tdd ~master: building configuration "unittest"...
Linking...
Running ./tests/dummy
------------------------------------------------------------
                          TESTS
------------------------------------------------------------
Running surfaceSpecificationTest()                    : PASS
Running cellSpecificationTest()                       : PASS
Running cellDimensionTest()                           : PASS
Running pinConsistancyTest()                          : FAIL
------------------------------------------------------------
End of tests. Total: 4 Failed: 1
```

Code Documentation

Code Documentation

HPC Potential

Thanks!
Questions?

Extras

- Git Manual https://git-scm.com/book/en/v2
- Git branch strategies
  https://nvie.com/posts/a-successful-git-branching-model/
- Test Driven Development http://www.agiledata.org/essays/tdd.html