Homework 4

Software Engineering

Anna Jinneman and William Roberts

Question 1

a.)

1. We downloaded a system called Angry IP Scanner.
2. Angry IP Scanner is a network scanner that can be useful for network administrators.
3. Angry IP Scanner has 14,303 lines of code. We calculated this by using an open source LOC counter named LocMetrics.

b.)

4. Output of tool:

```
Found 13 files that possibly contain design patterns.

C:\Users\risau\Desktop\College\angryIpScanSourceCode\ipscan-3.6.1\src\net\azib\ipscan\config\CommandLineProcessor.java
Possible patterns: Command

C:\Users\risau\Desktop\College\angryIpScanSourceCode\ipscan-3.6.1\src\net\azib\ipscan\config\CommandProcessor.java
Possible patterns: Command

C:\Users\risau\Desktop\College\angryIpScanSourceCode\ipscan-3.6.1\src\net\azib\ipscan\config\LoggerFactory.java
Possible patterns: Factory

C:\Users\risau\Desktop\College\angryIpScanSourceCode\ipscan-3.6.1\src\net\azib\ipscan\core\PortIterator.java
Possible patterns: Iterator

C:\Users\risau\Desktop\College\angryIpScanSourceCode\ipscan-3.6.1\src\net\azib\ipscan\core\ScannerDispatcherThreadFactory.java
Possible patterns: Factory

C:\Users\risau\Desktop\College\angryIpScanSourceCode\ipscan-3.6.1\src\net\azib\ipscan\core\state\ScanningState.java
Possible patterns: State

C:\Users\risau\Desktop\College\angryIpScanSourceCode\ipscan-3.6.1\src\net\azib\ipscan\core\state\StateMachine.java
Possible patterns: State

C:\Users\risau\Desktop\College\angryIpScanSourceCode\ipscan-3.6.1\src\net\azib\ipscan\core\state\StateTransitionListener.java
Possible patterns: State
```

```
C:\Users\risau\Desktop\College\angryIpScanSourceCode\ipscan-3.6.1\src\net\azib\ipscan\fetchers\HTTPProxyFetcher.java
Possible patterns: Proxy

C:\Users\risau\Desktop\College\angryIpScanSourceCode\ipscan-3.6.1\src\net\azib\ipscan\gui\SWTAwareStateMachine.java
Possible patterns: State

C:\Users\risau\Desktop\College\angryIpScanSourceCode\ipscan-3.6.1\src\net\azib\ipscan\gui\actions\CommandsMenuActions.java
Possible patterns: Command

C:\Users\risau\Desktop\College\angryIpScanSourceCode\ipscan-3.6.1\src\net\azib\ipscan\gui\menu\CommandsMenu.java
Possible patterns: Command

C:\Users\risau\Desktop\College\angryIpScanSourceCode\ipscan-3.6.1\src\net\azib\ipscan\util\SequenceIterator.java
Possible patterns: Iterator
```

5. This tool looks for instances of design patterns by first cycling through all the directories within a given starter file and finding source files, from which it enters each one and compares it's java source code to predefined definitions of design patterns. Each pattern definition contains its own compareTo() function and implements IComparable to ensure easy comparing. For each block of code that fits the bill (positively returned compareTo()) it adds that instance to a list of PatternFinderResults, which it can write to a file. Quite simple and easy to use, but I'm sure it took quite a while to create.

6. The process used by this tool can definitely be considered inaccurate at times. There is no learning of the algorithm just strict comparing and that can become messy and inaccurate. Each example of a specific design pattern the program found could be just an unused segment of code or an accidental inclusion (as if the programmer didn't mean to use that design method.) It really comes down to how close the segments of code are to the examples he compares them too, and what those examples might look like.

   If I were to sit down and take my time in creating a program to detect design patterns I would create a Machine Learning model that would learn itself what design patterns are which. Feed it (in learning) thousands of examples of code that contain the different patterns and let it construct what the general Strategy Method, Singleton Method, etc, look like. With a good enough model this method would be far superior to a simple CompareTo.

Question 2

a.) We uploaded our files to GitHub by simply opening our new Repository on Github's website and dragging our finished homework files into the upload section. We then commented on what was changing on this commit and committed the changes to master.

b.) We used the commands Git add, Commit and Push

Stage:

```
C:\Users\annaj\OneDrive\Documents\GitHub\ESOF322>git add dummy.txt
```

Commit:

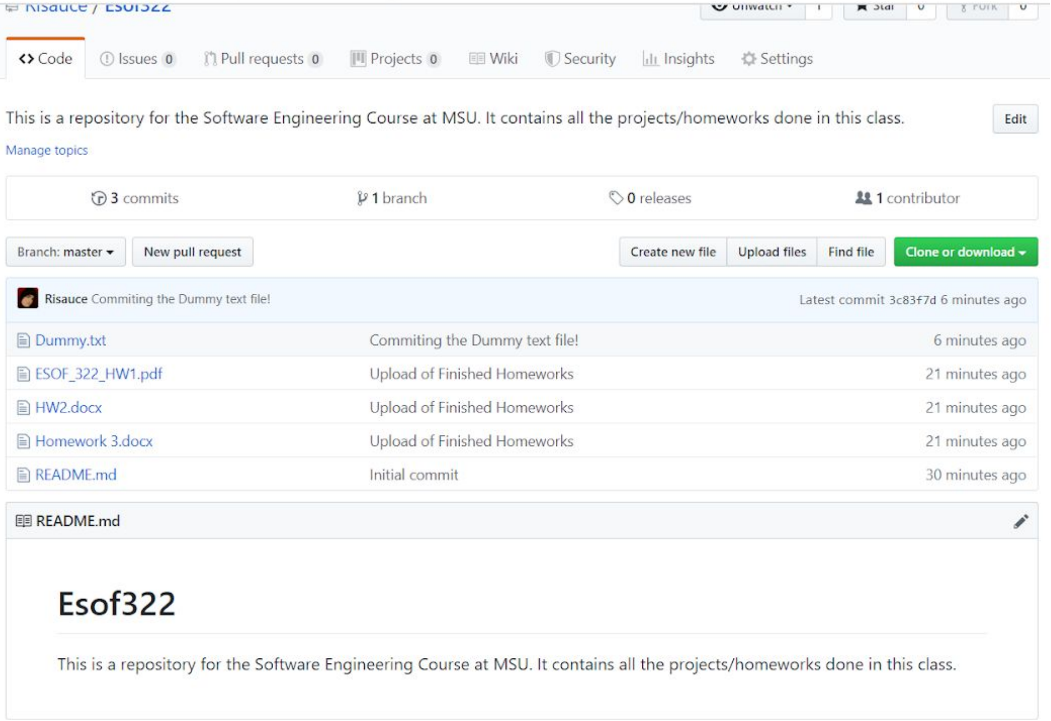
```
C:\Users\annaj\OneDrive\Documents\GitHub\ESOF322>git commit -m "Commiting dummy file"
[master 823c18b] Commiting dummy file
 1 file changed, 1 insertion(+)
 create mode 100644 dummy.txt
```

Push:

```
C:\Users\annaj\OneDrive\Documents\GitHub\ESOF322>git push origin master
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 342 bytes | 342.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/annajinneman/ESOF322.git
36f9656..823c18b  master -> master
```

Screenshot:

William Roberts



This is a repository for the Software Engineering Course at MSU. It contains all the projects/homeworks done in this class.

3 commits | 1 branch | 0 releases | 1 contributor

Risauce Commiting the Dummy text file! Latest commit 3c83f7d 6 minutes ago

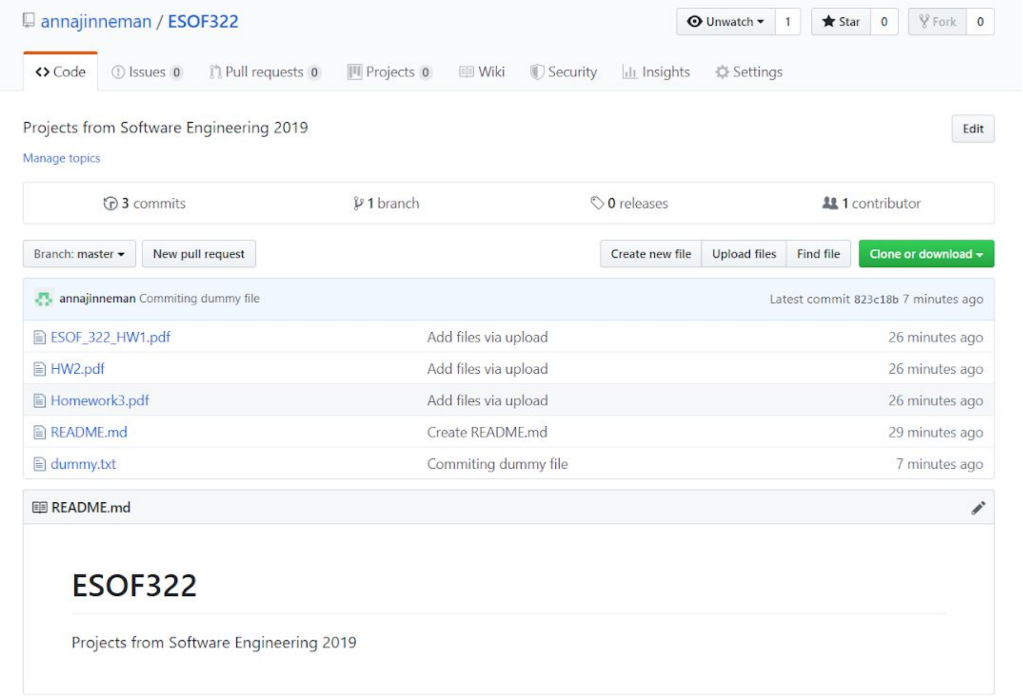
Dummy.txt	Commiting the Dummy text file!	6 minutes ago
ESOF_322_HW1.pdf	Upload of Finished Homeworks	21 minutes ago
HW2.docx	Upload of Finished Homeworks	21 minutes ago
Homework 3.docx	Upload of Finished Homeworks	21 minutes ago
README.md	Initial commit	30 minutes ago

README.md

Esof322

This is a repository for the Software Engineering Course at MSU. It contains all the projects/homeworks done in this class.

Anna Jinneman



Projects from Software Engineering 2019

3 commits | 1 branch | 0 releases | 1 contributor

annajinneman Commiting dummy file Latest commit 823c18b 7 minutes ago

ESOF_322_HW1.pdf	Add files via upload	26 minutes ago
HW2.pdf	Add files via upload	26 minutes ago
Homework3.pdf	Add files via upload	26 minutes ago
README.md	Create README.md	29 minutes ago
dummy.txt	Commiting dummy file	7 minutes ago

README.md

ESOF322

Projects from Software Engineering 2019