VisCad

This page describes how to download, install, and use VisCad on Mac OS X.

Install

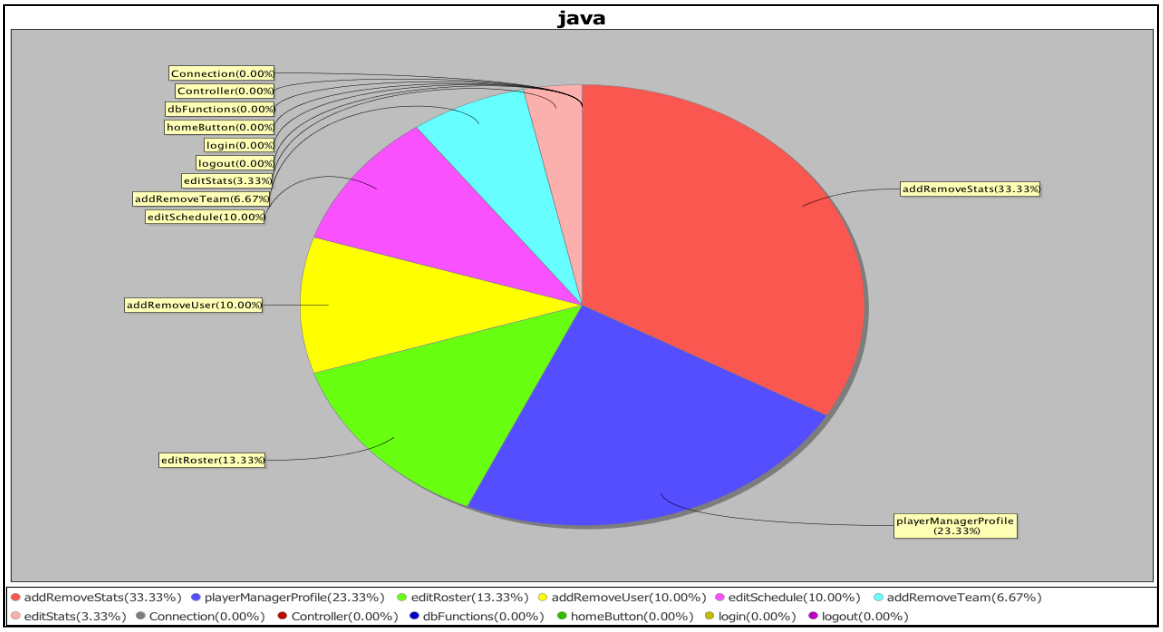
Download from http://homepage.usask.ca/~mua237/viscad/styled-3/index.html. Unzip the the download with your favorite decompression tool.

Using

So using VisCad as a pain. The *_functions-clones-0.3-classes.xml had to have a bunch of custom edits before I could actually import it into VisCad. You might be able to change the config file in NiCad to get the results, but I didn't bother since there aren't that many clones in the system. Once you load the project everything is relatively simple.

Determining Clone Distribution

In the navigation tree, navigate to where our JAVA code resides (*cmpt371group02/src/java*). Below is a visualization of our distribution.



Clone Class Metrics

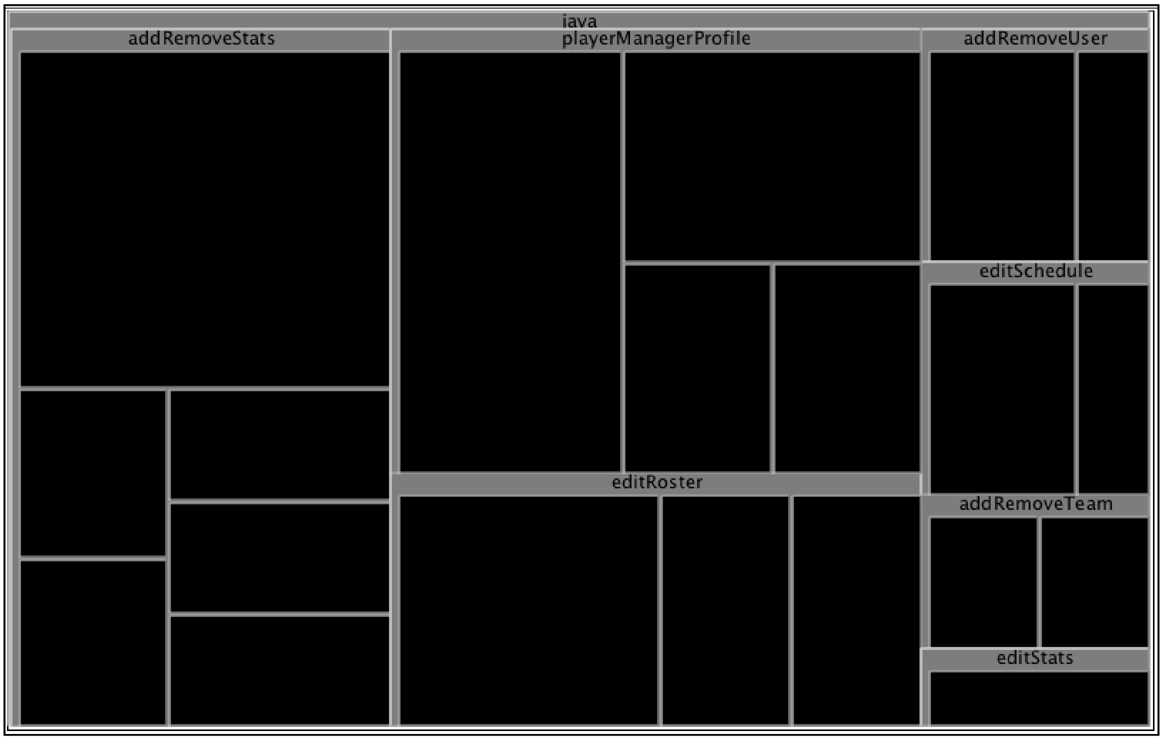
Right click on the *Clone Classes* window and select *Clone Class Metrics*. Here you will find some measurables about your clones. The following is the the report for Team Efforts Clones.

CCID	CloneClassSize	CloneClassLength	Max Length	Min length	Avg. Length	Files Associated	CloneClassRadius
1	2	104	53	51	52	2	0
2	6	214	41	31	35.666668	6	1
3	2	75	38	37	37.5	2	0
4	3	86	31	27	28.666666	3	0
5	2	53	27	26	26.5	2	1
6	2	62	33	29	31	2	1
7	2	30	15	15	15	1	0

8	2	23	12	11	11.5	1	0
9	2	24	13	11	12	1	0
10	7	80	12	11	11.428572	7	1

TreeMap

In the navigation tree, right click on any of files or folders and select *Show TreeMap*. In the *TreeMap* view, the bottom left hand corner contains the clone classes. Clicking on a clone class will highlight that class in the *TreeMap* to the left. Moving the mouse over the *TreeMap* gives the file name, path, clone fragment number, and number of cloned lines. Below is the *TreeMap* for the soccer system.



Clone Code Browser

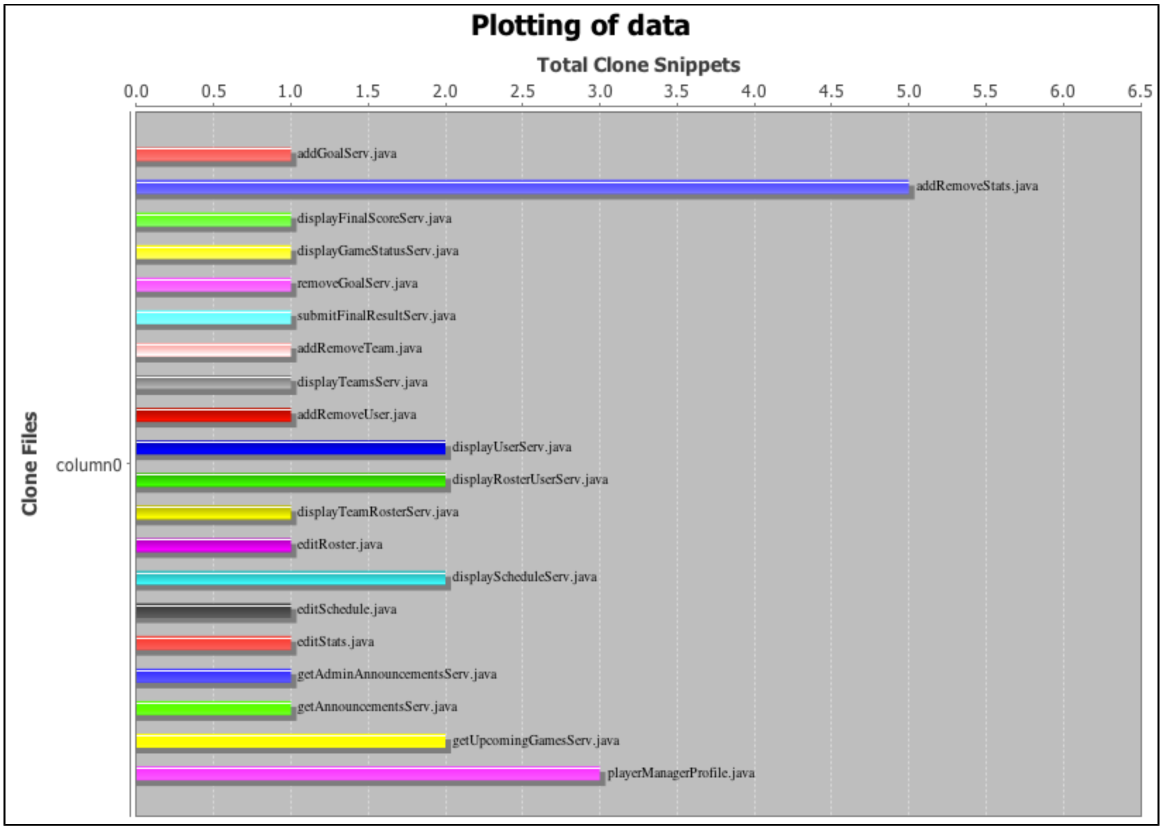
In the navigation tree, right click on any of files or folders and select *Open Clone List.* In the *Code Clone Browser* you will be able to view all code clones, if you switch to the *Clone Pairs* view, you can view the source code for each pair and compare them if you click the *Differencer* button.

Browse Clones By Files

In the navigation tree, right click on any of files or folders and select *Open Clone File List.* In the *Browse Clone By Files* window you will be able to view all files within the source code. You may choose which file to view and then you may select which source code to view that matches that clone.

Metrics

In the navigation tree, right click on any of files or folders and select *Metrics,* you will then be presented with two options *Metrics For Directories* and *Metrics For Files.*
This could be used to target specific files or directories to tackle first when trying to eliminate clones. Below is a plot of the total clone snippets of Team Efforts Files.

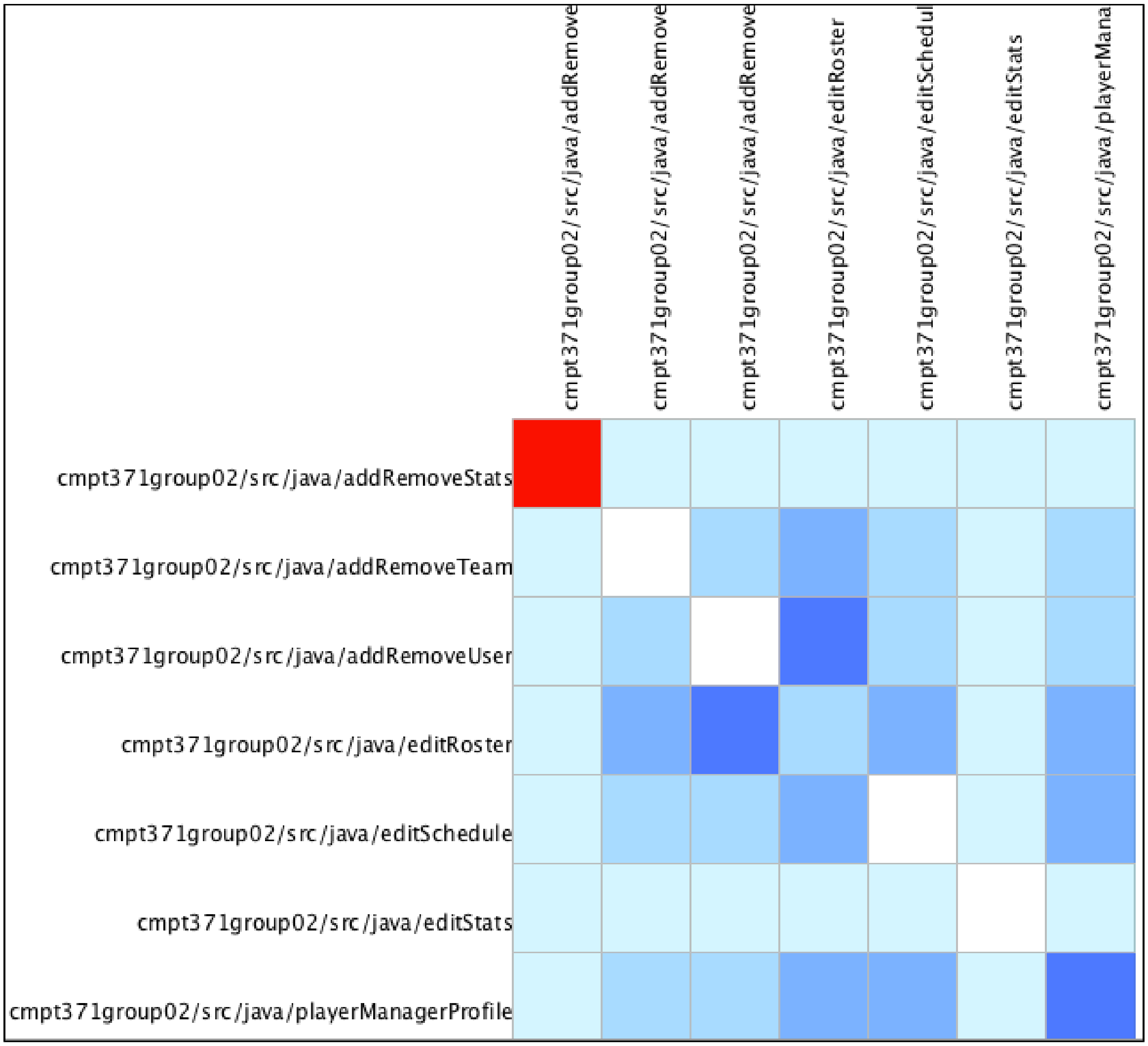


Analyze Overlapping

In the navigation tree, right click on any of files or folders and select *Analyze Overlapping.* There were no overlapping clones in Team Efforts project.

ScatterPlotView

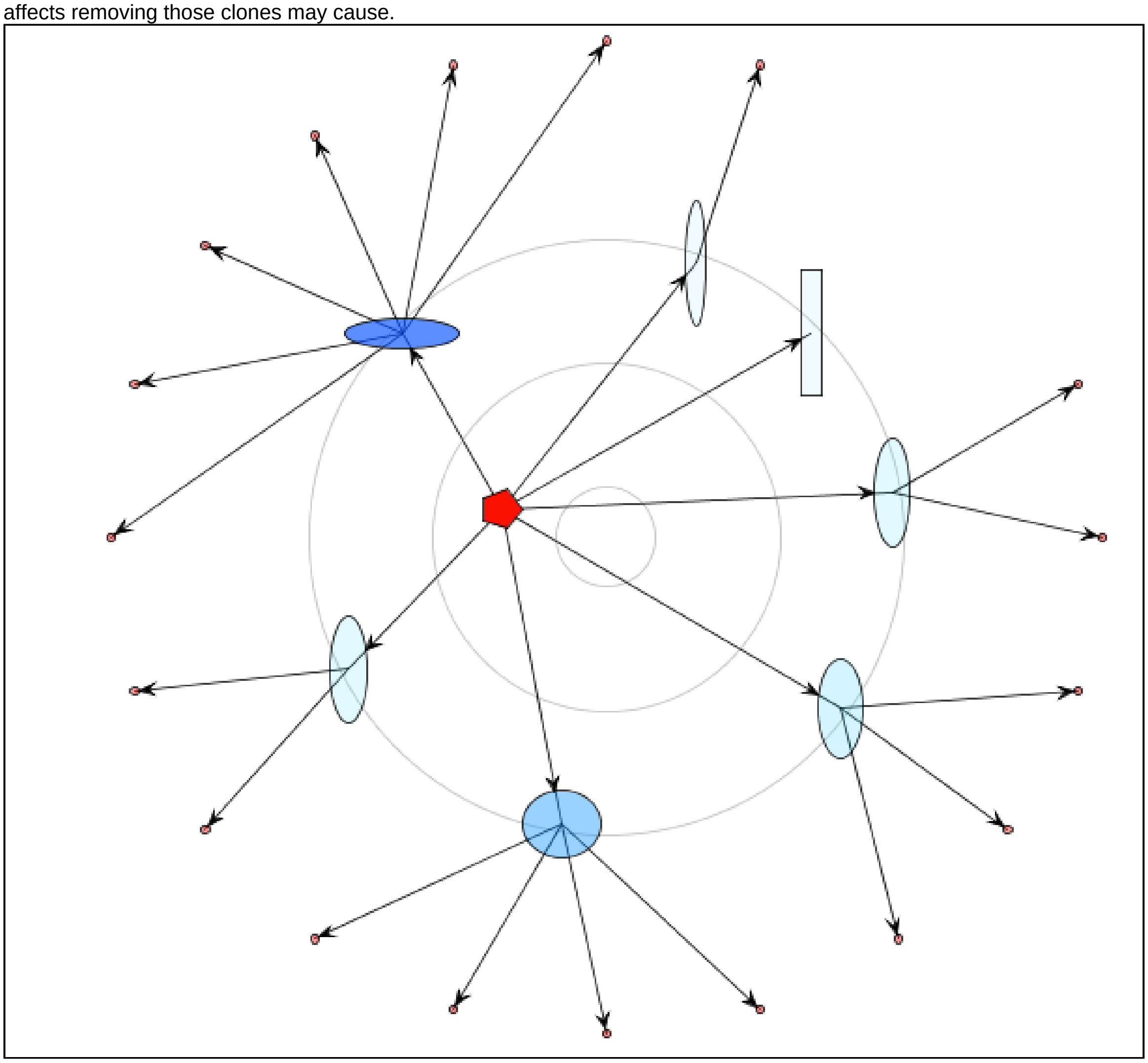
In the navigation tree, right click on any of files or folders and select *ScatterPlotView.* The ScatterPlotView gives a heat map of where code clones occur, either by files or directories. The diagonal line can be ignored as it compares a file (or directory) to itself. For the Soccer System the *editRoster* and *addRemoveUser* have the highest number of code clones. The heat map below can be read as the darker the color the more code clones the directory (file) pair contains.



Hierarchical Dependency Graph

In the navigation tree, right click on any of files or folders and select *Hierarchical Dependency Graph.* You will then be asked some viewing options. The red pentagon (as seen below) is the selected directory (in this case the *java* folder). The squares represent folders that can be expanded, circles represent folders that can be collapsed. The color of these objects represent the number of clones detected in those folders.

The small circles represent the files with code clones. The Hierarchical Dependency Graph gives a great view of where clones exist and what affects removing those clones may cause.



Report

Detection of code clones is the first step in eliminating clones. However, detecting them is not enough. Clones should be eliminated one by one, piece by piece. To determine which clones should be eliminated first, Team Effort used VisCad to visualize the report created by NiCad. VisCad is a visualization tool that allows users to analyze large volumes of raw cloning data in an interactive fashion. Users can analyze and identify distinctive code clones through a set of visualization techniques, metrics and data filtering operations. The use of VisCad is relatively simple, however, the report produced by NiCad3 is not in the proper format for VisCad to read, so some intelligent editing had to be done to reformat the report. Once VisCad was able to interpret the report, Team Effort was able to analyze the current state of the soccer system.

Clone Distribution

As seen in the above figure, addRemoveStats package contains the most code clones at 33%, and combined with playerManagerProfile package the two packages make up over 50% of the detected clones in the software. Eliminating the code clones in these two packages would eliminate the majority of clones in the system.

Clone Class Metrics

It is often important to analyze the metrics of clones. For instance, as seen above, there are two clone classes that occur 6 or more times. If those clones are eliminated almost half of all clones will be eliminated. It may also be prudent for Team Effort to remove the clones with the most lines of code contained in them. VisCad will also produce a visual graph of most of the clone class metrics in addition to the table output.

Tree Map

The Tree Map re-enforces which clones to eliminate first, and which clones have the potential to be the most harmful. AddRemoveStats has a very large clone that could prove to have high risk in future software maintenance.

Clone Code Browser & Browse Clones By Files

VisCad not only allows users to view clones through charts, tables, and graphs, but it also allows through the Clone Code Browser, and Browse Clones By files, to directly view the source code. In the browsers, the user can select which clones (files) to view and the source code is presented along with the highlighted clone. If the clone is in more than one file, the clone source can be viewed side by side.

Analyze Overlapping

VisCad does provide visualization for cone overlapping. Our soccer system, however, didn't have any overlapping.

Scatter Plot View

The Scatter Plot View allows Team Effort to determine which files share the most clones with one another. For the Soccer System the *editRoster* and *addRemoveUser* have the highest number of code clones. The heat map below can be read as the darker the color the more code clones the directory (file) pair contains.

Hierarchical Dependency Graph

The Hierarchical Dependency Graph is a visualization that allows for determination of the location (in the file system) of clones. Since the soccer system is not a large system the graph did not provide a lot of insight, but in a large system, the graph could provide an important visualization.