



Projekt: JBombberman
Metriken

Pascal Kistler
Silvan Adrian
Fabian Binna

1 Änderungshistorie

Datum	Version	Änderung	Autor
09.03.15	1.00	Erstellung des Dokuments	Gruppe
27.05.15	1.01	Metriken zu Projekt einfügen	Silvan Adrian
27.05.15	1.02	Vorbereitung Abgabe	Silvan Adrian

Inhaltsverzeichnis

1	Änderungshistorie	2
2	Einführung	4
2.1	Zweck	4
2.2	Gültigkeitsbereich	4
2.3	Definitionen und Abkürzungen	4
2.4	Übersicht	4
2.5	STAN	5
2.6	Beschreibung	6
3	Findbugs	7
3.1	Beschreibung	7
4	Jacoco	8
4.1	Beschreibung	8
5	Eclipse Metrics	9
5.1	Beschreibung	9

2 Einführung

2.1 Zweck

Dieses Dokument beinhaltet die Metriken zum Projekt.

2.2 Gültigkeitsbereich

Dieses Dokument ist ab der Projektabgabe gültig.

2.3 Definitionen und Abkürzungen

Siehe Glossar

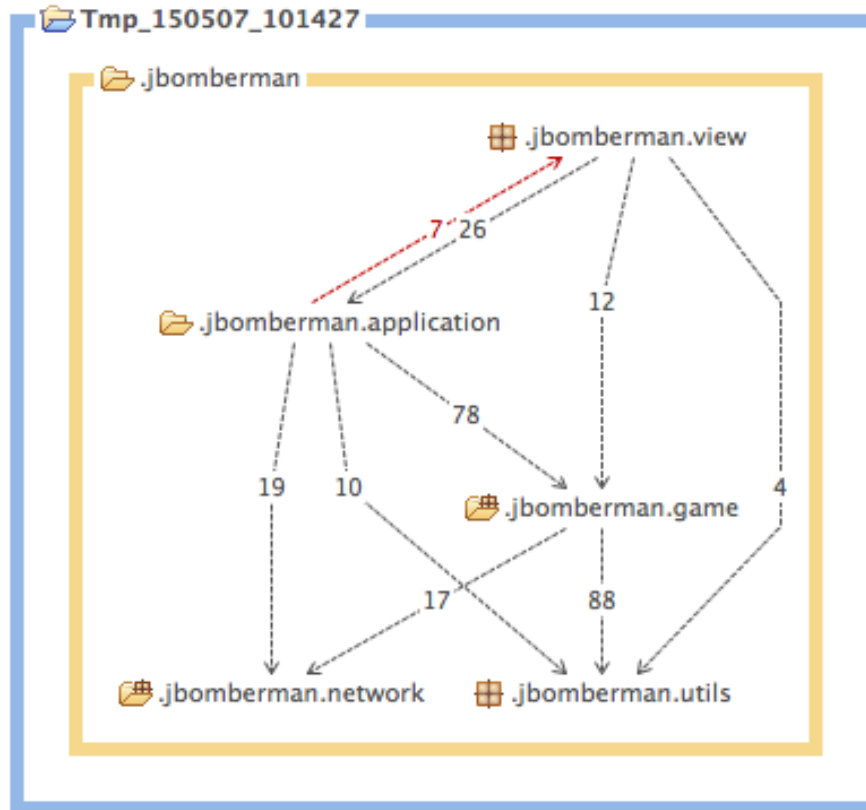
2.4 Übersicht

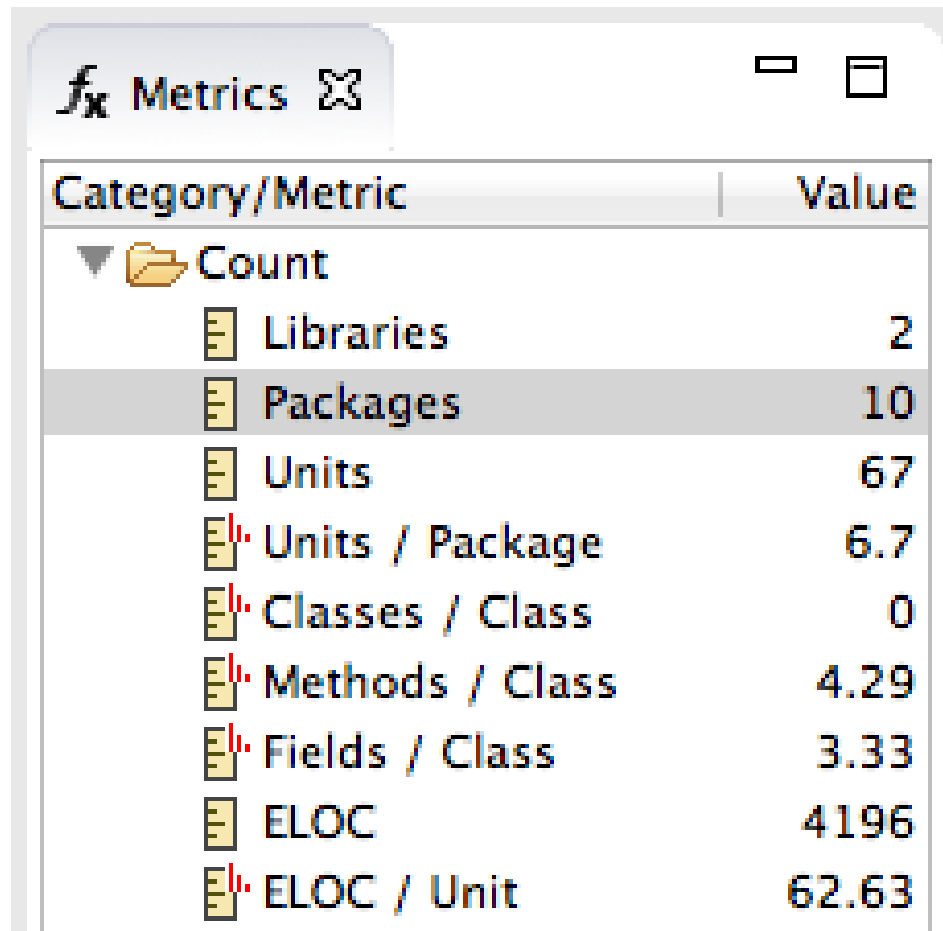
Es werden während dem Projekt verschiedene Metriken verwendet um auf Fehler oder ähnliches zu prüfen.

- STAN (Strukturanalyse)
- Findbugs
- Jacoco
- Eclipse Metrics

2.5 STAN

STAN wurde zur Strukturanalyse verwendet und um Abhängigkeiten zwischen den einzelnen Packages aufzuzeigen.





The screenshot shows a window titled 'fx Metrics' with a table of metrics. The table has two columns: 'Category/Metric' and 'Value'. The 'Count' category is expanded, showing a list of metrics. The 'Packages' metric is highlighted. The metrics and their values are: Libraries (2), Packages (10), Units (67), Units / Package (6.7), Classes / Class (0), Methods / Class (4.29), Fields / Class (3.33), ELOC (4196), and ELOC / Unit (62.63). Each metric has a small icon to its left, and some have a red vertical bar next to the icon.

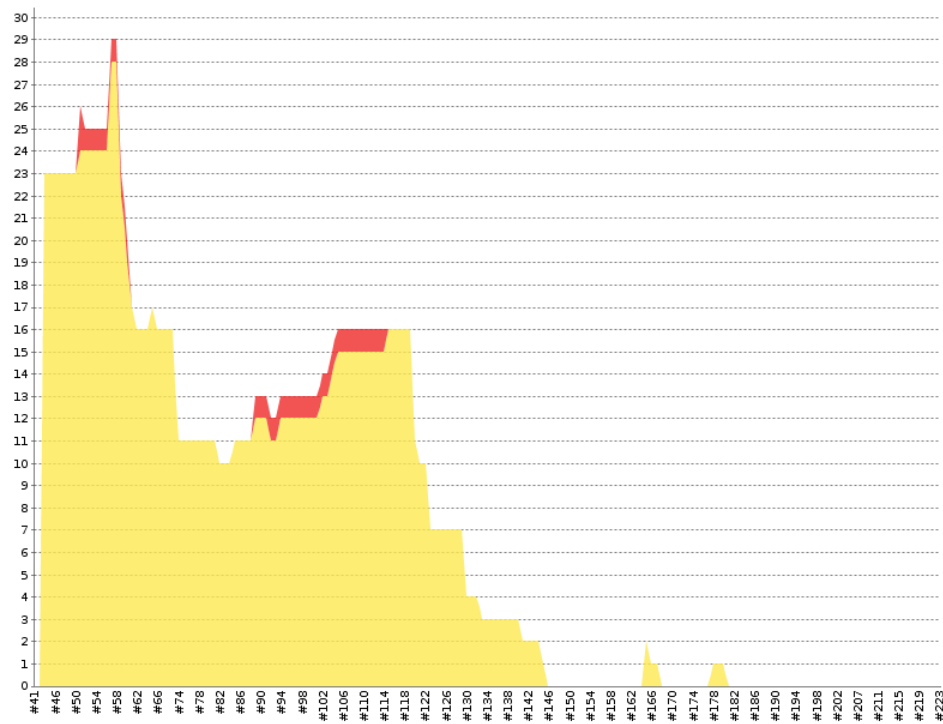
Category/Metric	Value
▼ Count	
Libraries	2
Packages	10
Units	67
Units / Package	6.7
Classes / Class	0
Methods / Class	4.29
Fields / Class	3.33
ELOC	4196
ELOC / Unit	62.63

2.6 Beschreibung

Zwar konnten einige Abhängigkeiten gelöst werden (zwischen utils und Game) -> jedoch konnte aus Zeitgründen und nicht sofortiges funktionieren die Abhängigkeit zwischen Application und View nicht gelöst werden.

3 Findbugs

Findbugs wurde zur statischen Code Analyse verwendet, um Fehler im Code aufzuzeigen.



3.1 Beschreibung

Am Anfang des Projekt waren noch einige Warnungen vorhanden, diese wurden jedoch dann nach und nach beseitigt. Zum Schluss bestehen keine Findbugs Warnungen mehr.

4 Jacoco

Jacoco wurde zur Analyse der Test Code Coverage gebraucht (als Ant Task).

4.1 Beschreibung

Jedoch wurde zu keinem Zeitpunkt eine genügend hohe Test Code Coverage erreicht was das einfügen des Graphen unnötig macht -> siehe Jenkins.

5 Eclipse Metrics

Eclipse Metrics wurde für die Statistik des Projekts zum Abschluss eingesetzt.

Metric	Total	Mean	Std. Dev.	Maximum	Resource causing Maximum	Method
► Number of Parameters (avg/max per method)		0.744	0.918	5	jbomberman/src/io/zonk/jbomberman/game/client...	Sprite
► Number of Static Attributes (avg/max per type)	24	0.407	0.692	4	jbomberman/src/io/zonk/jbomberman/application...	
► Effort Coupling (avg/max per packageFr...		3.214	3.447	11	jbomberman/src/io/zonk/jbomberman/game/client...	
► Specialization Index (avg/max per type)		0.127	0.327	1	jbomberman/src/io/zonk/jbomberman/game/Tim...	
► Number of Classes (avg/max per packageFr...	59	4.214	3.839	13	jbomberman/src/io/zonk/jbomberman/game/server...	
► Number of Attributes (avg/max per type)	171	2.898	3.986	24	jbomberman/src/io/zonk/jbomberman/view/Lobb...	
► Abstractness (avg/max per packageFragment)		0.094	0.255	1	jbomberman/src/io/zonk/jbomberman/network...	
► Normalized Distance (avg/max per package...		0.245	0.343	1	jbomberman/src/io/zonk/jbomberman/utills	
► Number of Static Methods (avg/max per type)	8	0.136	0.43	2	jbomberman/src/io/zonk/jbomberman/game/Acti...	
► Number of Interfaces (avg/max per package...	2	0.143	0.35	1	jbomberman/src/io/zonk/jbomberman/game	
► Total Lines of Code	3266					
► Weighted methods per Class (avg/max per type)	468	7.932	8.509	40	jbomberman/src/io/zonk/jbomberman/application...	
► Number of Methods (avg/max per type)	234	3.966	3.242	17	jbomberman/src/io/zonk/jbomberman/application...	
► Depth of Inheritance Tree (avg/max per type)		1.712	1.09	6	jbomberman/src/io/zonk/jbomberman/view/StartF...	
► Number of Packages	14					
► Instability (avg/max per packageFragment)		0.671	0.39	1	jbomberman/src/io/zonk/jbomberman/application...	
► McCabe Cyclomatic Complexity (avg/max p...		1.934	2.613	18	jbomberman/src/io/zonk/jbomberman/game/serv...	tick
► Nested Block Depth (avg/max per method)		1.302	0.741	5	jbomberman/src/io/zonk/jbomberman/application...	waitForPlayers
► Lack of Cohesion of Methods (avg/max per type)		0.297	0.312	0.896	jbomberman/src/io/zonk/jbomberman/view/Lobb...	
► Method Lines of Code (avg/max per method)	1945	8.037	20.25	222	jbomberman/src/io/zonk/jbomberman/view/Lobb...	LobbyPanel
► Number of Overridden Methods (avg/max p...	9	0.153	0.404	2	jbomberman/src/io/zonk/jbomberman/game/client...	
► Afferent Coupling (avg/max per packageFra...		4.5	7.651	25	jbomberman/src/io/zonk/jbomberman/utills	
► Number of Children (avg/max per type)	17	0.288	1.378	8	jbomberman/src/io/zonk/jbomberman/game/client...	

5.1 Beschreibung

Dabei kommen wir auf eine Total Lines of Code von 3266.