Week 8 Status Report: Ascension

Our three target metrics were :

* McCabe Cyclomatic complexity
* Method Lines of Code
* Weighted methods per class.

We choose these metrics specifically because they focus on our codes effective simplicity. In other words these metrics will hopefully suggest how and where our code may or may not have become convoluted through the development process.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Metric | Total | Mean | Std. Dev. | Maximum | Resource Causing Maximum | Method |
| Number of Overridden Methods | 8 | .4 | 1.53 | 7 | PlayerDeckTest |  |
| Number of Attributes | 79 | 3.95 | 4.832 | 18 | Turn |  |
| Number of Children | 2 | 0.1 | 0.3 | 20 |  |  |
| Number of Classes | 20 | 20 | 0 | 20 |  |  |
| Method Lines of Code | 1702 | 9.562 | 17.239 | 140 | Turn | executeAction |
| Number of Methods | 171 | 8.55 | 6.682 | 24 | TurnTest |  |
| Nested Block Depth |  | 1.281 | 0.662 | 5 | Main | getCenterDeck |
| Depth of Inheritance Tree |  | 1.25 | 0.698 | 4 | Game |  |
| Number of Packages | 1 |  |  |  |  |  |
| Afferent Coupling |  | 0 | 0 | 0 |  |  |
| Number of Interfaces | 1 | 1 | 0 | 1 |  |  |
| McCabe Cyclomatic Complexity |  | 1.876 | 3.795 | 35 | Turn | executeAction |
| Total Lines of Code | 2388 |  |  |  |  |  |
| Instability |  | 1 | 0 | 1 |  |  |
| Number of Parameters |  | 0.871 | 1.579 | 8 | PlayerDeck | PlayerDeck |
| Lack of Cohesion of Methods |  | 0.384 | 0.363 | 0.9 | Action |  |
| Efferent Coupling |  | 7 | 0 | 7 |  |  |
| Number of Static Methods | 7 | 0.35 | 1.108 | 5 | Main |  |
| Normalized Distance |  | 0.04 | 0 | 0.04 |  |  |
| Abstractness |  | 0.04 | 0 | 0.04 |  |  |
| Specialization Index |  | 0.049 | 0.173 | 0.778 | PlayerDeckTest |  |
| Weighted methods per Class | 334 | 16.7 | 15.815 | 67 | Turn |  |
| Number of Static Attributes | 5 | 0.25 | 0.698 | 3 | Game |  |

According to our metrics our three worst areas are executeAction for both McCabe Cyclomatic complexity and Method Lines of Code, and then number of parameters, eight, possessed by PlayerDeck. Of our projected problem areas only one is considered a true problem by the java metric plugin. Beginning with the complexity I believe the assertion that McCabe CC is high in the method executeAction because executeAction has numerous paths depending wholly on which action is being attempted. This really isn’t a problem because to cause this method to lose its complexity one would need to revamp the code in such a way to be extraordinarily inefficient. The number of parameters is, by our metrics, suggested as a trouble area but in reality the eight parameters do not really add any complexity to the overall code since the constructor for player deck is called once for each player. Another place I thought would be a trouble area was the Weighted Methods per class (WMC) which is another section that if very high could mean our classes are too complicated and should be simplified. But was not in this case considered to be a problem, though 67 is a pretty high number of WMCs and will bear watching incase its complexity increases beyond an acceptable level.

Fun Metrics:

* Number of Cards - because the number of cards directly effects how interesting the game can become. More interesting = More fun.
* Number of times Played – More the game is played the more likely people are enjoying the game.
* Number of People Playing at any one time.
* Number of Games Finished – People who quit are more likely to have been dissatisfied.

Number of Lines per Team Member

Jack Petry: 250 lines

Kenneth Faulkner: 250 lines

Gabriel Glenn: 150 lines

Code Coverage at 80.2%