

# Sprint Retrospective, Iteration #2

Context Project: Bust-a-move game

Group: 3

User Story #	Task #	Task Assigned To	Estimated Effort per Task (in hours)	Actual Effort per Task (in hours)	Done (yes / no)	Notes
I would like to see the current classes split into smaller, more specific classes. These classes must follow the RDD.	Separate the bubble storage from the Arena class	Jason	2	4	yes	For future reference, always refactor in batches and check whether it's still working or not...
	Move collision detection into its own class	Jason	1	2	yes	
	Use Point class instead of x,y variables	Justin	1/2	1	yes	We should add more methods using it in the future but it is fine for now.
	Create a Screen interface class	Justin	1/2	1/60	yes	Created but currently no use for and therefore gives a PMD error.
I would like to see 60% of the checkstyle errors to be fixed	Resolve all checkstyle errors in newly created class	Everyone	6	12	yes	Newly created classes do contain many checkstyle errors
	Fix checkstyle errors in smaller classes	Winer	2	6	Yes	Reduced 900+ checkstyle errors to 250
I would like to see 40% of all the code tested	Test 75% of the Bubble storage class	Winer	2	-	No	BubbleStorage not completed yet. Spend a lot more time than expected on solving checkstyle errors and unit testing.
	Test 75% of the Bubble class	Winer	2	6	No	Line coverage 36% (of 75%)

	Test 75% of the Cannon class.	Justin	1	1/2	no	Untestable without mockito. Cannon class is too dependent on other classes. So we will probably be adding mockito next sprint to test more of the classes.
I would like to see a Observer pattern implemented in the program	Make Score class Observable	Maurice	2	1	yes	Score class is now an observable class.
	Make the VictoryScreen an Observer	Maurice	½	½	yes	Screen observes score class
	Make the DefeatScreen an Observer	Maurice	½	½	yes	Screen observes score class
	Make the PauseScreen an Observer	Maurice	½	½	yes	Screen observes score class
	Make the Game an Observer	Maurice	½	½	yes	game observes score class
I would like to see the Decorator pattern in the program	Create a PowerUp decorator	Calvin	2		yes	PowerUp implemented successfully using Decorator pattern
I would like to see a shorter package name.	Rename the packages to not have 'bust_a_move20162017. bust_a_move_framework'	Maurice	½	1	yes	Renamed to bust_a_move
I would like to see less packages	Put together packages which are closely related	Maurice	½	½	yes	Amount of packages decreased from 10 to 7
I would like to see 100% of the PMD errors fixed	Resolve all PMD violations	Calvin	2	0	no	Focused on Checkstyle errors this sprint
I would like 80% of the Findbug errors fixed.	Resolve the FindBugs bugs(bring down to at most 2 bugs)	Calvin	2	0	no	Focused on Checkstyle errors this sprint

I would like to see document of the design patterns implemented in the sprint	Create sequence diagram of the design patterns	Justin	1	1	yes	
	Create class diagram of the design patterns	Justin	1	1	yes	
	Write a description of why and how the pattern is implemented	Justin	1	1	yes	
I would like to see bubbles with special effect introduced in the game	Define 4 special effects	Calvin	1	1	yes	Defined 6 special effects: OBomb, RowBomb, SpeedUp, SlowDown, LongScope, ShortScope
	Implement special effects	Calvin	3	4	yes	Successfully implemented 6 special effects
I would like to see a documentation of the special effect of the bubbles	Write a requirements document	Calvin	1	1	yes	Done
	Create an UML of the new classes	Calvin	1	1	yes	Done, with class and sequence diagrams
I would like to see a document of code restructure	Create an UML of the restructured classes	Jason	1	1	yes	
	Create a class diagram of the restructured classes	Jason	1	1	yes	
	Write a requirements document	Jason	1	1	Yes	In repo docs folder

## Adjustments for the next Sprint

Fix most of the remaining checkstyle, PMD and findbugs errors. Make more testers for the classes including the one that weren't finished this sprint.