CS3217

Problem Set 4

Matric No	Name
A0259195N	**JUSTIN CHEAH YUN FEI**

Total Score: **64** /130

Item	Score	Remarks
Problem 1	3 /10	-7: Minimal, essentially not helpful * Outdated diagrams are not useful to the reader. * How do the new PS4 features (e.g. power ups, resizing, rotation, etc.) work? No explanation on the control flow/data flow and no diagram to illustrate. * How did you implement collisions in your physics engine? Also, seems like there's no improvement in the PS2 section of the DG, after previous grader provided the following comments: * Instead of only providing a link to the variation of MVVM architecture that you used, explain briefly on what this architecture is about and what makes this variation special. * A few sequence diagrams describing the most important flows in your application * Pictures showing which views corresponds to which view class to see what they look like
Problem 2	4 /4	
Problem 3	2 /4	-2: There is no special visual effect when ball enters the bucket. The number of balls left going up by 1 is not considered a *special effect*. It should be clearly visible and a visual special effect.
Problem 4	4 /4	
Problem 5.1	4 /8	-4: Game crashed when I activate green peg explosion with about only 15 green pegs. This performance issue wasn't documented in README
Problem 5.2	8 /8	
Problem 5 subtotal	12 /16	
Problem 6	2 /4	-2: Rectangular block collision not implemented well - very often ball can overlap with block while colliding and doesn't bounce off corners properly. Stubborn pegs can also overlap with the block in some cases.
Problem 7	4 /4	
Problem 8	4 /4	
Problem 9	8 /8	
Problem 10	4 /8	-2: Stubborn peg doesn't lose HP when hit wall. -2: The showing of HP dealt upon collision is erratic - sometimes it doesn't show, sometimes it shows 0 when it isn't 0.
Problem 11	8 /8	

CS3217

Problem Set 4

Matric No	Name
A0259195N	**JUSTIN CHEAH YUN FEI**

Total Score: **64** /130

Item	Score	Remarks
Problem 12	4 /4	
Problem 13	28 /40	
Bells and whistles		
Problem 14	4 /8	-2: Relatively ok set of tests for PS4 features. Missing some edge cases. Ok set of test cases for PS2 and PS3, but seems like previous graders suggestions were not implemented. -2: I don't see any unit tests at all. I'm not sure if you intended it to be such that the unit tests were interespersed in the expository tests. Writing something like "newPoint.x == oldPoint.x and newPoint.y - oldPoint.y < size / 2." doesn't mean that what you wrote is a unit test. What method is being tested in this "unit test"? Also having such a detail in the integration test is useless, as a QA testing the feature how am I supposed to know what is the values of newPoint and oldPoint etc.
Problem 15	4 /4	Great, seems like you learnt a lot from this module!
Issues	-24	-5: Memory leak (check memory graph in debugger) -2: Your ball speed is really too fast, such that sometimes it goes through objects or collides in very strange manners. -5: overlaps method not scalable (using if-else statements). Consider using double-dispatch pattern. -5: effectWhenActivated methods have direct access to gameStateManager. Why? Could have just passed callback function or use delegates/observer pattern. -5: Singleton antipattern in LocalPersistenceManager. Means anywhere in the code people can mess with your persistence. -2: Preloaded levels don't appear on select level screen.
Reflection bonus	3 /3	

CS3217

Problem Set 4

Matric No	Name
A0259195N	**JUSTIN CHEAH YUN FEI**

Total Score: **64** /130

Item	Score	Remarks
Coding style deduction	-10	-1: Magic values (e.g. in Board.swift) -1: print statement not removed -1: Different structs/classes into single file (e.g. ScreenPosition in GameVm) -4: SwiftLint errors * Type Name Violation * Identifier Name Violation -3: SwiftLint warnings * Todo Violation * Multiple Closures with Trailing Closure Violation * Identifier Name Violation
Late penalty	0	
Playtesting comments		It was ok-ish. I didn't really like how fast the ball was moving (it makes your app look very buggy) and the mechanism for activating the power up. Because let's say I put the green peg very near the cannon. When it hits, I want it to activate immediately. But now I can't because I need time to move my finger from shooting to pressing the power up button. But besides those, I think it was an ok experience.
Comments		Good job on completing PS4! I read your reflection, seems like you had to do a lot of refactoring, which I think is good because you took the previous graders' comments into consideration (e.g. tightly coupled Physics Engine and Game Engine). Seems like you learnt a lot, which is good! Hopefully you can apply your learnings to the final project. All the best!