

Suggested, reviewed, changed (or suggested to do no changes, or didn't change after review)					
Needs implementation					
Needs review			Needs suggestions		
Class	Reviewers	Main writer	Comments (please denote place, eg. methodName)	Result of discussion with group/ main Writer	Adjusted / refactored?
Button	Stefan, Alex	Mark	Stefan: isMouseEvent() could just return what's between the brackets of the if-statement, couldn't it?	Mark: yes, it could. Can be refactored. I probably have that in others spots as well. Oops.	Stefan: executed proposed changes
			Stefan: Can Button not extend MyRectangle? Instead of returning a entirely new MyRectangle at the getRectangle method.	Mark: no, Button has no association with rectangle beyond its use for mouse-over's. I don't think getRectangle is actually used... is it?	Stefan: left the code unchanged
			Alex: RND.getInstance very repetitious	Mark: findBugs does not like it if you store instances in a variable. Please don't fix.	
			Alex: have overloaded drawColor use first drawColor?	Mark: when you do this slick2D starts vomiting. Actually tried.	
GameState	Alex, Mark	All	Bart: after the split everything looks good		Class split into helper classes.
MainGame	Alex, Mark	Bart	Stefan: Nothing to add, initStateList looks a bit ugly but that particular code needs execution and checkstyle doesn't care.		
MenuGameOverState	Bart,Alex	Mark	Menno: exit, change 2*2*2 into 8. processButtons, change else if. May need to put the things in between into methods then.	If the complexity of processButtons is too high, yes this is a good idea. Just make sure the naming is consistent.	Changed 2*2*2 into 8. Refactored processButtons using methods for each option
MenuMainState	Bart,Alex	Mark	Menno: exit, change 2*2*2 into 8. processButtons, change else if. May need to put the things in between into methods then.	Same as for menuGameOverState!	Changed 2*2*2 into 8. Refactored processButtons using methods for each option
			Alex: In general for quite a few methods across all menus: unused container variables (probably after RND singleton)	Mark: these are methods such as Render(), which are specified by slick2D. They cannot be modified.	
MenuMultiPlayerState	Menno, Bart	Mark	Menno: Consider switch in processButtons	Same as for menuGameOverState!	Changed 2*2*2 into 8. Refactored processButtons using methods for each option
MenuSettingsState	Bart,Alex	Mark	Menno: Switch in processButtons, processButtons2, processColorButtons, drawSprites. Delete outcommented code in drawColorControls	A switch would be nice. However, a way to reduce duplicate code would be even better. And the commented code can go, yes.	Refactored into smaller methods. Deleted outcommented code
Popup	Bart,Alex	Mark	Stefan: Nothing to add		
RND	Menno, Bart	Mark	Menno: Consider renaming the instance to something like "instance". Do not put three statements on one line (init). Is it possible to make the drawColor functions use each other? Like, drawColor(RenderOptions) calls drawColor(RenderOptions, width, height) with default values? Something like this. Some of this code looks really similar. Rename text() to drawText? Might consider overloading this method instead of new method textSpecifiedColor, textNoColor, drawPowerup has a huge else if statement! Could refactor this into letting PowerupType return its own images. (MARK I DID THIS WHY DID YOU DELETE IT?? sadface) Also, multiple statements on one line, should not be done.	1) I'll look into merging draw functions later. Might be possible! 2) I'd rather you don't rename text(). It will result in a hilarious amount of checkstyle warnings for line length. 3) Same for text() as draw(). Might be merged, if possible. I'd rather do this myself. 4) I did that because it gave us glorious 100% powerup coverage etc. Of course, it's a terrible thing to do. Still did it. Can undo it.	Changed name to instance. Put multiple statements on multiple lines. Put powerup images in Powerup class.
Seperator	Bart,Alex	Mark	Menno: No problems		
			Alex: A few unused methods	Mark: noted, removing...	Mark: removed unused methods.
Textfield	Bart,Alex	Mark	Stefan: Nothing to add		
Client	Stefan, Menno	Bart/Alex	Menno: In general: Write else if to switch, only put things that can throw an exception in try block, don't use message2 everywhere but just message = message.trim(). Don't feel like writing down every occurrence of these things. circleMessage, use more descriptive names instead of THREE, FOUR, etc. Same for powerupMessage, coinMessage. Why the initialization with Shield in addPowerup, dictatePowerup? Can remove that.	General things can be changed (switch, try, trim). I don't understand what you mean with more specific names than THREE (this is just the number 3 for accessing the array). The instantiation with shield is indeed wrong, it should be initialized with null, otherwise the nullcheck in the bottom of the method doesn't do anything.	
Connector	Alex, Bart	Menno	Stefan: sometimes message.trim() can be used instead of the local variable message2.	Agreed. Use message = message.trim instead of message2 =message.trim	
			Stefan: again, else-if statements that check string equals can be replaced with case statements	Agreed as well	
Host	Stefan, Menno	Bart/Alex	Stefan: sometimes message.trim() can be used instead of the local variable message2.	Yes you can change this (just remember to assign it to itself: message = message.trim()).	Stefan: executed proposed changes
			Stefan: getPowerupType, the else if can be replaced with a case statement I believe	You are right.	Stefan: executed proposed changes
Level	Bart, Mark	stefan	Bart: Looks fine, no clear problems		
level 1-10	Bart, Mark	Menno/stefan	Bart: Looks fine, no clear problems		
LevelFactory	Menno, Bart	stefan	Menno: No problems		
LevelFactoryMultiPlaye	Menno, Bart	Alex/stefan	Not sure if all of these things need to be in the try block.	The first statement (string assignment) can be excluded from the try block, the other 2 statements should remain in the try block (they can both throw exceptions)	Put the first statement out
LevelFactorySinglePla	Menno, Bart	stefan	Not sure if all of these things need to be in the try block.	The first statement (string assignment) can be excluded from the try block, the other 2 statements should remain in the try block (they can both throw exceptions)	Put the first statement out
BouncingCirle	Menno, Stefan	Bart	Stefan: is it polite to use getters and setters when it's about variables of the Object itself? Genuine question.	Since the variables I'm trying to access are from the superclass I can't access them directly (they are private)	Stefan: left the code unchanged
			Stefan: How does the xSpeed for circles which intersect with gates work?	I check if the circles belongs to the required list (this means it is placed left from the gate), if that is true I set their speed to a negative value (going left) and vice versa	Stefan: left the code unchanged
			Stefan: toString, unnecessary local variables?	True, you can change that	Stefan: executed proposed changes
			Stefan: is it necessary for super.clone to be there in the clone() method?	I don't think so, but please check after removing.	Stefan: removed and put back (by Mark) after it was clear the code should be left unchanged
			Stefan: getSpeedForRadius is a giant else if, couldn't it just be a case statement? (Same goes for the two methods below it)	Mark here: yes, this is necessary. See Findbugs for specifics.	Menno: Fixed
				Absolutely true	
			Menno: Do we still need both constructors? hashCode seems useless. If statements in handleXCalculations, gate part, could be combined.	HashCode is necessary for checkstyle (wants equals and hashCode to appear together). Old constructor (without id) is used in tests, so if you want to rewrite them be my guest :). If statements in handleXCalculations, gate part cannot be comined, the inner if statement has an else statement (for which the first if still has to be true). By comining them you will be doublechecking the first expression which is not efficient	Removed old constructor and rewrote tests
CircleList	Menno, Stefan	Bart	Stefan: could use a checkHigherId method I suppose, to get that if method that is present twice into a seperate method. Menno: Nothing to	Sure, good idea	Stefan: executed proposed changes
Coin	Menno, Bart	Mark/Alex	Menno: No problems	(+1) :p	
FloatingScore	Menno, Bart	Mark	Menno: Do we really need all these constructors? I count five	Until java gets it's shit together with optional variables etc, sadly, yes. Now you can have a floatingScore where you can just stuff almost anything into it, which is really convenient. Of course, we could go through the code and extract circle.getscore by hand.	Left code unchanged
Gate	Mark,Alex	Stefan / Bart	Bart: everything looks good		
HighScores	Menno,Alex	Bart	Menno:Remove Logger getter/setter because it is not needed (+1 - Alex)	Yeah fine	Removed logger getter
HighScoresParser	Menno, Stefan	Bart	Menno:Remove setLogger method because it is unnecessary. Only put things that can throw an exception in try block (readHighScores, writeHighScores). Improve exception message in readHighScores. Write a non-custom exception message in writeHighScores. Put close() in finally block (readHighScores, writeHighScores). hs.setLogger is useless in readHighScores.	You can change all these things, just be careful with those exceptions (eclipse should not you when something can throw an exception)	removed logger setter. Left the try block alone, because otherwise 3 try blocks would emerge. Did put close in finally
			Stefan: Nothing to add		
LevelContainer	Alex, Bart	stefan	Bart: everything looks good		
Logger	Mark,Alex	Bart, Menno	Stefan: getCurrentTimeStamp.length can be halved by saying "return sdf.format(new Date());"	Sure	Stefan: executed proposed changes
			Bart: after the split everything looks good		
MyRectangle	Mark, Bart	Menno	Stefan: Nothing to add		

Player	Mark, Bart	Alex		Bart: after the split everything looks good		Class split into helper classes
PlayerList	Mark, Stefan	Bart		Stefan: why is super() in the constructor?	Automatically generated, can be removed	Stefan: executed proposed changes
				Stefan: the methods setPlayerImage and setPlayerImages are exactly the same thing.	Mark: oh wow how did that even happen. One of these can be removed.	
Score	Mark, Stefan	Bart		Stefan: why is super() in the constructor?	Automatically generated, can be removed	Stefan: executed proposed changes
ShutDownHook	Mark, Menno	Bart		Menno: No problems		
				Stefan: Nothing to add		
Weapon	Mark, Stefan	Alex		Stefan: why is super() in the constructor?	Automatically generated, can be removed	Stefan: executed proposed changes
				Stefan: in update, the this.visible could be assigned !(expressionCurrentlyInIf)? Implying this.visible is otherwise always true	Sounds good, even better would be this.visible = y >= ceiling.getHeight(). Then you avoid	Stefan: executed the changes that were proposed and it turned out to affect functionality. It was changed back.
WeaponList	Stefan, Mark	Bart		Stefan: why is super() in the constructor?	Automatically generated, can be removed	Stefan: executed proposed changes
				Stefan: are setLaserBeamImage and setLaserTipImage actually used? I don't think so (the setters aren't either actually)	Mark here: these are not used, but may very well be useful in the future. You can easily change the shape/color of the laser with these.	Stefan: left it in, in case people want to use it in the future
FastPowerup	Menno, Stefan	Bart		Menno: No problems, changed javadoc where needed		
				Stefan: Nothing to add		
FreezePowerup	Menno, Stefan	Bart		Menno: No problems, changed javadoc where needed		
				Stefan: Nothing to add		
InstantLaser	Menno, Stefan	Alex		Menno: No problems, changed javadoc where needed		
				Stefan: Nothing to add		
Powerup	Menno, Stefan	Alex		Menno: No problems, changed javadoc where needed		
				Stefan: Nothing to add		
SlowPowerup	Menno, Stefan	Bart		Menno: No problems, changed javadoc where needed		
				Stefan: Nothing to add		
SpeedPowerup	Menno, Stefan	Bart		Menno: No problems, changed javadoc where needed		
				Stefan: Nothing to add		
Spiky	Menno, Stefan	Alex		Menno: No problems, changed javadoc where needed		
				Stefan: in the update method, couldn't stayvisible just be assigned by saying: boolean stayVisible = this.isVisible(); (or using that, eliminating the local variable in its entirety)	Mark: no, this would change the behavior of stayVisible, because it can be switched to true from outside.	Stefan: left code unchanged.
Shield	Menno, Stefan	Alex		Menno: No problems, changed javadoc where needed		
				Stefan: Nothing to add		