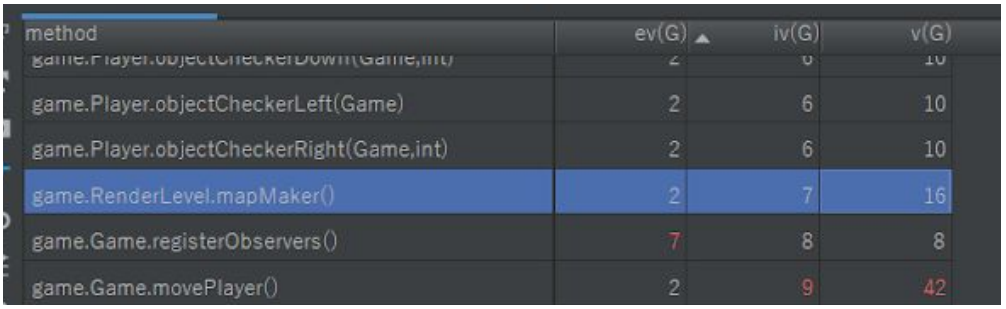
Assignment 4 Exercise 2

We used metricReloaded and CodeMR to calculate code metrics of our code, here are some methods and classes which we thought could be improved.

1) **movePlayer()** in Game was decreased by taking some statements and making them into seperate method which reduced the complexity. So we did extract method refactoring.



method	ev(G)	iv(G)	v(G)
game.Player.objectCheckerLeft(Game)	2	6	10
game.Player.objectCheckerRight(Game,int)	2	6	10
game.RenderLevel.mapMaker()	2	7	16
game.Game.registerObservers()	7	8	8
game.Game.movePlayer()	2	9	42

method	ev(G)	iv(G)	v(G)
game.Game.leftWall()	1	2	2
game.Game.lose()	3	4	4
game.Game.make()	2	2	2
game.Game.moveGhosts()	1	5	5
game.Game.movePlayer()	1	8	13

2) **StartScreen()** constructor was complex so we again did extract **method refactoring**

method	ev(G)	iv(G)	v(G)
Register.Register()	1	9	9
StartScreen.main(String[])	1	1	1
StartScreen.StartScreen()	1	11	11
Total	**259**	**400**	**534**
Average	1.23	1.90	2.53

method	ev(G)	iv(G)	v(G)
StartScreen.lostFocusUsername()	1	2	2
StartScreen.main(String[])	1	1	1
StartScreen.StartScreen()	1	5	5
Total	**262**	**403**	**537**
Average	1.22	1.88	2.51

3) We did extract method refactoring for **mapMaker** in **render level**. We decreased complexity throughout quite a bit.

method	ev(G)	iv(G)	v(G)
game.FruitPellet.points()	4	1	4
game.Game.movePlayer()	2	9	42
game.Game.registerObservers()	7	8	8
game.RenderLevel.mapMaker()	2	7	16

method	ev(G)	iv(G)	v(G)
game.RenderLevel.getPixels()	1	1	1
game.RenderLevel.mapMaker()	1	6	6

4) We did extract method refactoring for **render()** in **render level**. We decreased complexity throughout quite a bit.

method	ev(G)	iv(G)	v(G)
game.FruitPellet.points()	4	1	4
game.Game.movePlayer()	1	8	13
game.Game.registerObservers()	7	8	8
game.RenderLevel.render(Graphics)	1	12	12

method	ev(G)	iv(G)	v(G)
game.RenderLevel.randySetter(int,int)	1	2	2
game.RenderLevel.render(Graphics)	1	3	3

5) We also did extract method refactoring for **moveGhost()** in both **Pinky** and **Blinky**. As a result, the complexity has been brought down considerably. Any more refactoring was sadly not possible with how we coded the classes.

method	ev(G)	iv(G)	v(G)
ghost.Pinky.moveGhost(int,int)	4	20	31
ghost.Pinky.Pinky(int,int,SpriteSheet)	1	1	1
ghost.Pinky.getType()	1	1	1
Total	**6**	**22**	**33**
Average	2,00	7,33	11,00

method	ev(G)	iv(G)	v(G)
ghost.Pinky.moveGhost(int,int)	2	10	13
ghost.Pinky.getNewDestination(Point)	2	2	6
ghost.Pinky.movePinky(int,int,int)	2	2	6
ghost.Pinky.oneValidMove(int,boolean,boolean,boolear	5	1	5
ghost.Pinky.Pinky(int,int,SpriteSheet)	1	1	1
ghost.Pinky.getType()	1	1	1
Total	**13**	**17**	**32**
Average	2,17	2,83	5,33

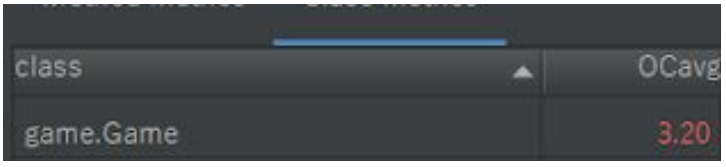
Classes Refactoring

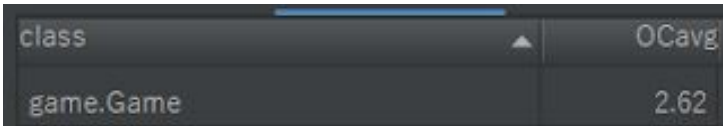
1) **Level** class which had **high lack of cohesion**. So we systematically chose some methods from level and put them in a **new class RenderLevel** and thus reduced lack of cohesion.

ID	CLASS	COUPLING	COMPLEXITY	LACK OF COHESION	SIZE	LOC	COMPLEXITY	COUPLING	LACK OF COHESION	SIZE
1	Player					291	medium-high	medium-high	medium-high	low-medium
2	Game					282	medium-high	low-medium	medium-high	low-medium
3	Level					132	low-medium	low-medium	high	low-medium

ID	CLASS	COUPLING	COMPLEXITY	LACK OF COHESION	SIZE	LOC	COMPLEXITY	COUPLING	LACK OF COHESION	SIZE
1	Game					264	medium-high	low-medium	medium-high	low-medium
2	Level					102	low-medium	low-medium	medium-high	low-medium

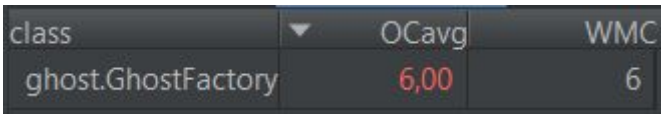
2) The average operational complexity of game class was more than threshold so we decreased it by extract method refactoring

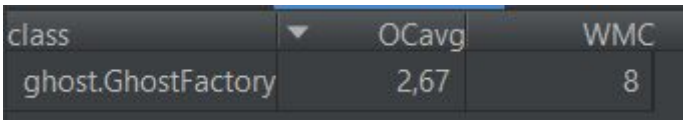




3) The average operational complexity of RenderLevel class was more than threshold so we decreased by some extract method refactoring

4) The average operational complexity of **GhostFactory** class was more than threshold so we decreased by some extract method refactoring





5) The operational complexity of class **Level** and weight method complexity of cwas quite above the threshold. We were able to decrease that significantly by dividing some of the tasks of Level into renderLevel.