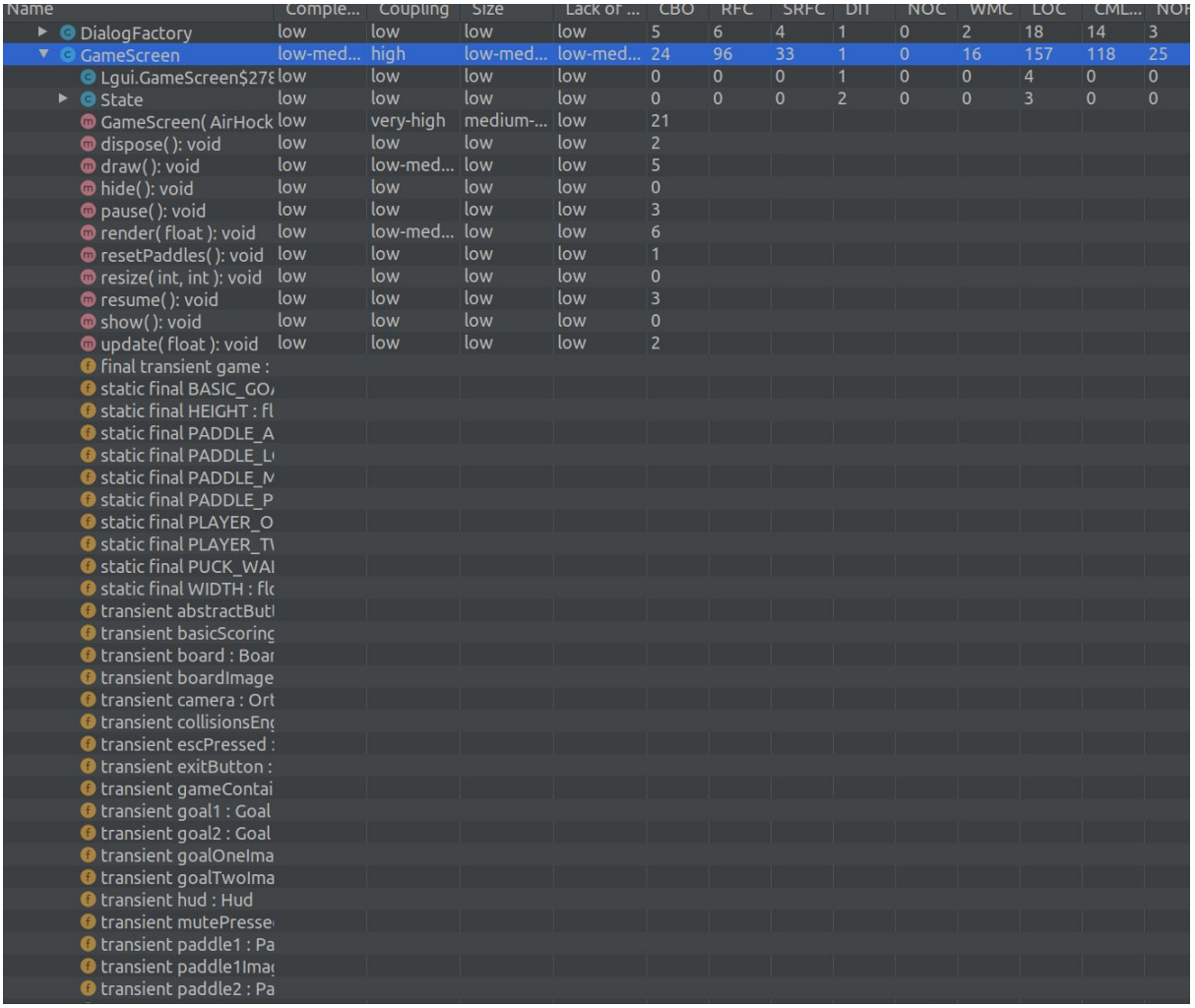
Class improvements

Game Screen previously had very high coupling.



Name	Comple...	Coupling	Size	Lack of ...	CBO	RFC	SRFC	DIT	NOC	WMC	LOC	CML...	NOF
DialogFactory	low	low	low	low	5	6	4	1	0	2	18	14	3
GameScreen	low-med...	high	low-med...	low-med...	24	96	33	1	0	16	157	118	25
Lgui.GameScreen$27	low	low	low	low	0	0	0	1	0	0	4	0	0
State	low	low	low	low	0	0	0	2	0	0	3	0	0
GameScreen(AirHock	low	very-high	medium-...	low	21								
dispose( ): void	low	low	low	low	2								
draw( ): void	low	low-med...	low	low	5								
hide( ): void	low	low	low	low	0								
pause( ): void	low	low	low	low	3								
render(float): void	low	low-med...	low	low	6								
resetPaddles( ): void	low	low	low	low	1								
resize(int, int): void	low	low	low	low	0								
resume( ): void	low	low	low	low	3								
show( ): void	low	low	low	low	0								
update(float): void	low	low	low	low	2								
final transient game :													
static final BASIC_GO													
static final HEIGHT : fl													
static final PADDLE_A													
static final PADDLE_L													
static final PADDLE_M													
static final PADDLE_P													
static final PLAYER_O													
static final PLAYER_T													
static final PUCK_WAI													
static final WIDTH : flo													
transient abstractBut													
transient basicScorinç													
transient board : Boar													
transient boardImage													
transient camera : Ort													
transient collisionsEn													
transient escPressed :													
transient exitButton :													
transient gameContai													
transient goal1 : Goal													
transient goal2 : Goal													
transient goalOneIma													
transient goalTwoIma													
transient hud : Hud													
transient mutePresse													
transient paddle1 : Pa													
transient paddle1Ima													
transient paddle2 : Pa													

We improved the coupling by moving many of the game dependent classes (Puck, Paddle, Board, etc.) into a separate object that holds these (Game Container). Even so, the complexity of this class increased, as we actually removed methods from this class. RFC went down, and LOC also went down significantly.

Name	Complexity	Coupling	Size	Lack of Cohesion	CBO	RFC	SRFC
GameScreen	medium-high	medium-high	low-medium	low-medium	19	131	32
Lgui.GameScreen$19	low	low	low	low	0	0	0
State	low	low	low	low	0	0	0
GameScreen(AirHoc	low	very-high	low	low	14		
dispose( ): void	low	low	low	low	2		
draw( ): void	low	low-medium	low	low	6		
hide( ): void	low	low	low	low	0		
pause( ): void	low	low	low	low	3		
render(float): void	low	low-medium	low	low	6		
resize(int, int): void	low	low	low	low	0		
resume( ): void	low	low	low	low	3		
show( ): void	low	low	low	low	0		
update(float): void	low	low	low	low	2		
final transient game :							
static final HEIGHT : f							
static final WIDTH : fl							
transient abstractBut							
transient camera : Or							
transient collisionSou							
transient escPressed :							
transient exitButton :							
transient gameConta							
transient hud : Hud							
transient mutePresse							
transient resumeButt							
transient scoreSound							
transient stage : Stag							
transient state : State							

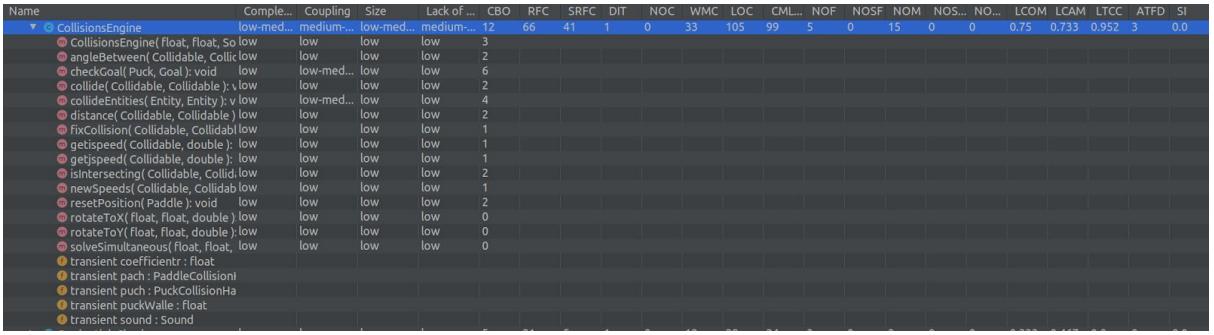
CollisionsEngine Class

Before: medium-high complexity

CollisionsEngine	medium-...	medium-...	low-med...	medium-...	12	66	50	1	0	52	150	146	3	0	2
CollisionsEngine(floa	low	low	low	low	1										
angleBetween(Collid	low	low	low	low	2										
checkGoal(Puck, Goal	low	low-med...	low	low	5										
collide(Collidable, Co	low	low	low	low	2										
collide(Paddle, Board	low	low	low	low	2										
collide(Puck, Board):	low	low	low	low	2										
collideEntities(Entity,	low	low	low	low	2										
distance(Collidable, C	low	low	low	low	2										
fixCollision(Collidable	low	low	low	low	1										
fixPaddlePosition(Pa	low	low	low	low	2										
fixPaddleXposition(P	low	low	low	low	3										
fixPaddleYPosition(P	low	low-med...	low	low	4										
fixPuckPosition(Puck,	low	low	low	low	2										
fixPuckXPosition(Puc	low	low-med...	low	low	5										
fixPuckYPosition(Puc	low	low-med...	low	low	5										
getispeed(Collidable,	low	low	low	low	1										
getjspeed(Collidable,	low	low	low	low	1										
isIntersecting(Collida	low	low	low	low	2										
newSpeeds(Collidabl	low	low	low	low	1										
resetLeft(Puck): void	low	low	low	low	1										
resetPosition(Paddle	low	low	low	low	2										
resetPuckPosition(Pu	low	low	low	low	1										
resetRight(Puck): voi	low	low	low	low	1										
rotateToX(float, float	low	low	low	low	0										
rotateToY(float, float	low	low	low	low	0										
solveSimultaneous(fl	low	low	low	low	0										
transient coefficientr															
transient puckWalle :															
transient sound : Sour															

Methods for the collisionsHandling with the board and Puck and Paddle respectively were refactored out into separate classes to reduce the amount of logic the collisions engine was responsible for. This improved the complexity as the collisions engine had fewer methods and was responsible for less.

After: low-medium complexity

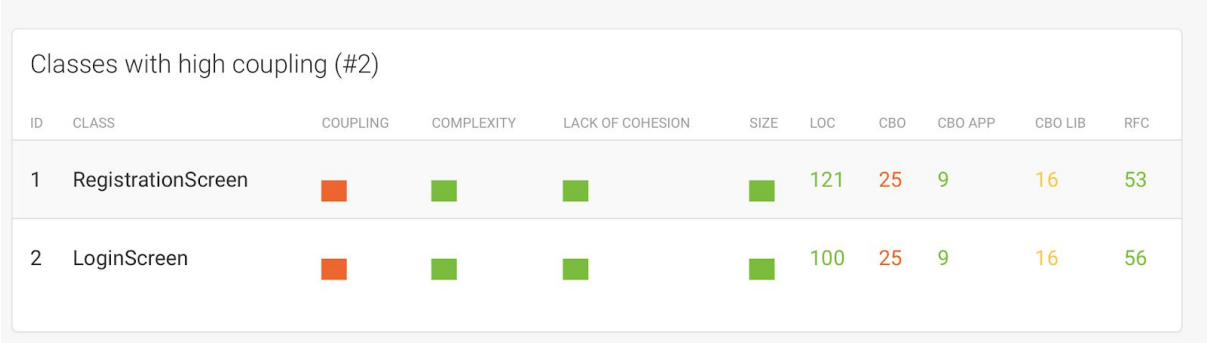


Name	Comple...	Coupling	Size	Lack of ...	CBO	RFC	SRFC	DIT	NOC	WMC	LOC	CML...	NOF	NOSF	NOM	NOS...	NO...	LCOM	LCAM	LTCC	ATFD	SI
CollisionsEngine	low-med...	medium-...	low-med...	medium-...	12	66	41	1	0	33	105	99	5	0	15	0	0	0.75	0.733	0.952	3	0.0
CollisionsEngine(float, float, So	low	low	low	low	3																	
angleBetween(Collidable, Collic	low	low	low	low	2																	
checkGoal(Puck, Goal): void	low	low-med...	low	low	6																	
collide(Collidable, Collidable): v	low	low	low	low	2																	
collideEntities(Entity, Entity): v	low	low-med...	low	low	4																	
distance(Collidable, Collidable)	low	low	low	low	2																	
fixCollision(Collidable, Collidabl	low	low	low	low	1																	
getispeed(Collidable, double):	low	low	low	low	1																	
getjspeed(Collidable, double):	low	low	low	low	1																	
isIntersecting(Collidable, Collid	low	low	low	low	2																	
newSpeeds(Collidable, Collidab	low	low	low	low	1																	
resetPosition(Paddle): void	low	low	low	low	2																	
rotateToX(float, float, double)	low	low	low	low	0																	
rotateToY(float, float, double):	low	low	low	low	0																	
solveSimultaneous(float, float,	low	low	low	low	0																	
transient coefficientr : float																						
transient pach : PaddleCollisionI																						
transient puch : PuckCollisionHa																						
transient puckWalle : float																						
transient sound : Sound																						

The lack of cohesion was also reduced, but not enough for codeMR to recognize it as a full step. LCOM and LCAM were reduced.

It was not possible to decrease the Lack of Cohesion more, as the Collisions engine has to divide up the different types of objects, and calculate the appropriate actions after these collisions.

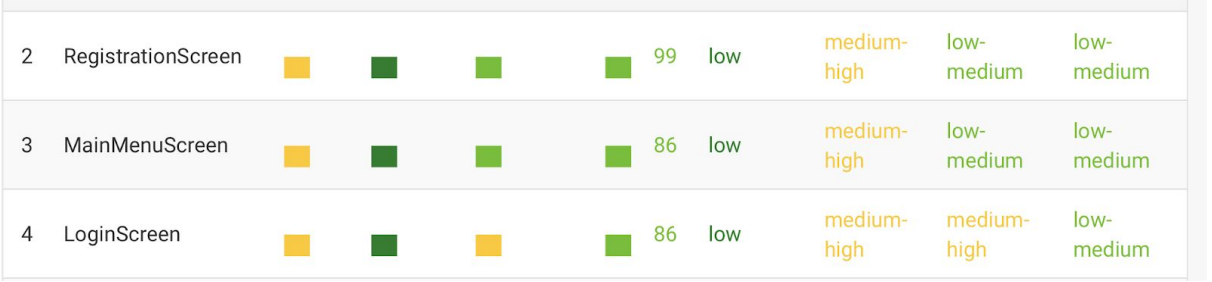
RegistrationScreen & LoginScreen
Improvements in coupling.
Before:



Classes with high coupling (#2)

ID	CLASS	COUPLING	COMPLEXITY	LACK OF COHESION	SIZE	LOC	CBO	CBO APP	CBO LIB	RFC
1	RegistrationScreen					121	25	9	16	53
2	LoginScreen					100	25	9	16	56

After:



2	RegistrationScreen					99	low	medium-high	low-medium	low-medium
3	MainMenuScreen					86	low	medium-high	low-medium	low-medium
4	LoginScreen					86	low	medium-high	medium-high	low-medium

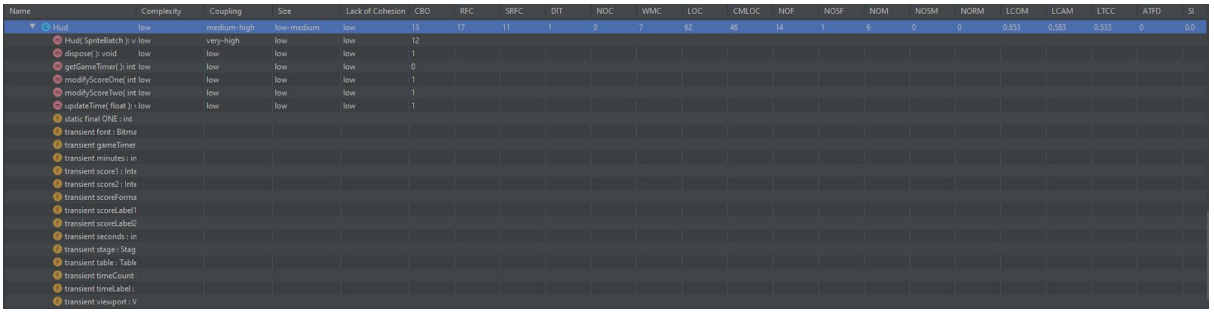
The main problem of these 2 classes was the huge amount of dependencies between them and CredentialsChecker, Dialog & Query classes.

How the problem was solved: created a new class (SubmitAuthenticationCredentials) that would do the checks with the database. This new object is now used in Login Screen, as well as in Registration screen.

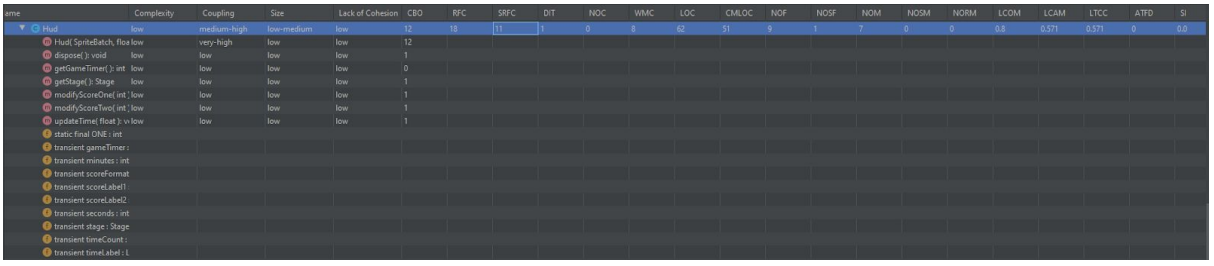
Hud class had medium-high coupling :

- Instead of using AirHockeyGame object to get the width and height of the game, these parameters are passed to the constructor of the Hud.
- Instead of using “Integer” class Hud uses primitive type “int”
- Attribute Stage is now private and there is now a getter to retrieve it from an object.

Hud class before changes:



Hud class after changes



The Coupling Between Objects metric decreased after refactoring.

Regarding class coupling in the GUI package,

low coupling is very hard to achieve since LibGDX screens require a lot of Objects (i.e. different buttons, different fields).

Note:

It was quite difficult to find tangible changes to make to the classes. We had just spent a lot of time optimizing and refactoring the code in the previous sprint to improve these exact issues that codemt was addressing. These metrics, and this exercise would have been much more effective earlier in this project.

Methods improvements

1) **Login Screen - constructor** used to have very high coupling.

Name	Complexity	Coupling	Size	Lack of Cohesion	CBO	RFC	SRFC	DIT
LoginScreen	low-medium	high	low-medium	low-medium	25	57	27	1
Lgui.LoginScreen$31	low	low	low	low	0	0	0	1
LoginScreen(AirHock	low	very-high	low	low	13			
dispose( ): void	low	low	low	low	0			
hide( ): void	low	low	low	low	0			
pause( ): void	low	low	low	low	0			
render(float): void	low	medium-high	low	low	10			
resize(int, int): void	low	low	low	low	0			
resume( ): void	low	low	low	low	0			
show( ): void	low	low	low	low	0			
submitCredentials( ):	low	medium-high	low	low	10			

After extracting parts of the method in other methods, the coupling is now medium-high.

Name	Complexity	Coupling	Size	Lack of Cohesion	CBO	RFC	SR
LoginScreen	low	high	low-medium	medium-high	25	39	31
Lgui.LoginScreen$3228;	low	low	low	low	0	0	0
LoginScreen(AirHockeyGame): void	low	medium-high	low	low	8		
addDoneButton( ): void	low	low	low	low	3		
addExitButton( ): void	low	low	low	low	3		
addImage(String): void	low	low	low	low	3		
createTextFields( ): void	low	low	low	low	3		
dispose( ): void	low	low	low	low	0		
hide( ): void	low	low	low	low	0		
pause( ): void	low	low	low	low	0		
render(float): void	low	medium-high	low	low	10		

2) **Login Screen - render** method

Before refactoring, the coupling was medium-high.
After refactoring, it dropped to low-medium:

Name	Complexity	Coupling	Size	Lack of Cohesion	CBO	RFC	SRFC	DIT
LoginScreen	low-medium	high	low-medium	medium-high	25	51	33	1
Lgui.LoginScreen$30	low	low	low	low	0	0	0	1
LoginScreen(AirHock	low	low-medium	low	low	6			
addDoneButton( ): vc	low	low	low	low	3			
addExitButton( ): voi	low	low	low	low	3			
addImage(String): v	low	low	low	low	3			
createTextFields( ): v	low	low	low	low	3			
dispose( ): void	low	low	low	low	0			
draw( ): void	low	low	low	low	3			
drawSpriteBatch( ): v	low	low-medium	low	low	5			
hide( ): void	low	low	low	low	0			
pause( ): void	low	low	low	low	0			
render(float): void	low	low-medium	low	low	4			
resize(int, int): void	low	low	low	low	0			
resume( ): void	low	low	low	low	0			
show( ): void	low	low	low	low	0			
submitCredentials( ):	low	medium-high	low	low	10			

3) RegistrationScreen - constructor

Before - the coupling of the constructor method was very-high:

Name	Complexity	Coupling	Size	Lack of Cohesion	CBO	RFC	SRFC	DIT
RegistrationScreen	low-medium	high	low-medium	low-medium	25	54	27	1
Lgui.RegistrationScre	low	low	low	low	0	0	0	1
RegistrationScreen(A	low	very-high	low-medium	low	13			
dispose( ): void	low	low	low	low	0			
hide( ): void	low	low	low	low	0			
pause( ): void	low	low	low	low	0			
render(float): void	low	medium-high	low	low	10			
resize(int, int): void	low	low	low	low	0			
resume( ): void	low	low	low	low	0			
show( ): void	low	low	low	low	0			
submitCredentials( ):	low	medium-high	low-medium	low	10			

After - the coupling dropped to medium-high:

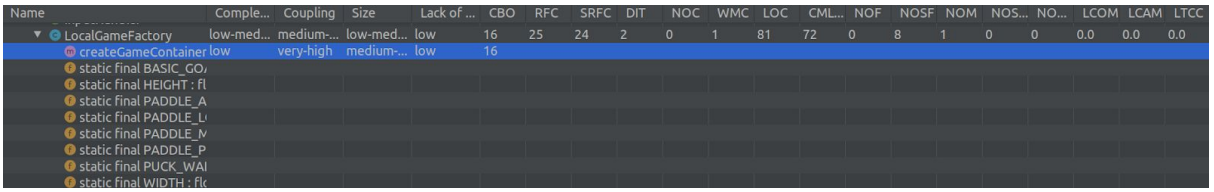
Name	Complexity	Coupling	Size	Lack of Coh...	CBO	RFC	SRFC	DIT
template								
gui	low	medium-high	medium-high	low				
RegistrationScreen	low-medium	high	low-medium	low-medium	25	62	33	1
Lgui.Registration!	low	low	low	low	0	0	0	1
RegistrationScree	low	medium-high	low	low	8			
createButtons( ):	low	low-medium	low	low	4			
createFields( ): voi	low	low	low	low	3			
dispose( ): void	low	low	low	low	0			
draw( ): void	low	low	low	low	3			
drawSpriteBatch(	low	low-medium	low	low	5			
hide( ): void	low	low	low	low	0			
pause( ): void	low	low	low	low	0			
render(float): voi	low	low-medium	low	low	4			
resize(int, int): vc	low	low	low	low	0			
resume( ): void	low	low	low	low	0			
show( ): void	low	low	low	low	0			
submitCredentials	low	medium-high	low-medium	low	10			

In order to decrease coupling in the constructor, we decided to separate from it the logic of creating fields and buttons into createFields() and createButtons() methods.

4) **RegistrationScreen - render** method
   For the render() method I separated it into a general draw() method which I separated further by putting the spriteBatch logic into the drawSpriteBatch() method.
   Before refactoring: medium-high coupling.
   After refactoring: low-medium coupling (see pictures above).

5) **LocalGameFactory - createGameContainerClass**

Before: Very high coupling.



After: Medium-high coupling. The creation of some objects was refactored into different methods so the createGameContainer function did less. This size of this method was also decreased.