PROJECT Design Documentation

Team Information

- Team name: Incognito - 11b
- Team members
    - Jacquelyn Leung
    - Mallory Bridge
    - Anthony Ferraioli
    - Celeste Gambardella
    - Kelly Vo

Executive Summary

WebCheckers is a webapp to play a game of checkers with other people. You have the option to watch games live or play games against others.

Purpose

> *The purpose of the project is to work together in an agile team and create a game of WebCheckers.*

Glossary and Acronyms

TERM	DEFINITION
VO	Value Object
UI	User Interface
AI	Artificial Intelligence
OS	Operating System
HW	Hardware

Requirements

This section describes the features of the application.

Sign In

- *When a player is not signed in they have the option to sign in.*
- *If a player chooses to sign in they are redirected to a sign in page*
- *The sign in page displays two text boxes one for a username and password*
- *When the sign in button is clicked we check to see if a valid username and password was entered.*
- *If there is an invalid username or password an error message will be displayed.*
- *When a valid username and password is entered the user will be redirected back to the homepage.*
- *A user will be able to see other players logged if they are logged in otherwise they will just see the number of players available.*

Start a Game

- *A user will be able to select a player to start a game if they are logged in.*
- *If the player selected is in a game already then the user will see an error message.*
- *If a user selects a player not in a game then that player and user will both be redirected back to game view.*
- *A player will be only be able to drag and drop a piece on valid spaces.*
- *A player will only be able to move there pieces.*

Simple Move

- *A user will be able to move along the board when it is their turn.*
- *A user can backup their move, and submit their turn.*
- *A user can also be kinged when at a king spot and move backwards when kinged.*

Jump Move

- *A user will be able to jump over an opponent and capture their opponent piece*
- *A user will be able to make a single jump or multiple jumps.*
- *A user will be able to backup one jump at a time.*

End Game

- *A user will be able to win, lose, or tie a game.*
- *A USer will be able to resign a game be lose when resigned*
- *A user is only able to resign a game when it is their turn.*

> - *A user wins by capturing all opponents pieces or making their opponent no longer able to move*
>
> *AI Enhancement*
>
> - *A user will be able to win, lose, or tie a game.*
> - *A USer will be able to resign a game be lose when resigned*
> - *A user is only able to resign a game when it is their turn.*
>
> *Spectator Enhancement*
>
> - *A user will be able to win, lose, or tie a game.*
> - *A USer will be able to resign a game be lose when resigned*
> - *A user is only able to resign a game when it is their turn.*

Definition of MVP

> *Where developers provide the smallest amount of features to help satisfy new customers and get feedback for new enhancements that can be implemented in future product development.*

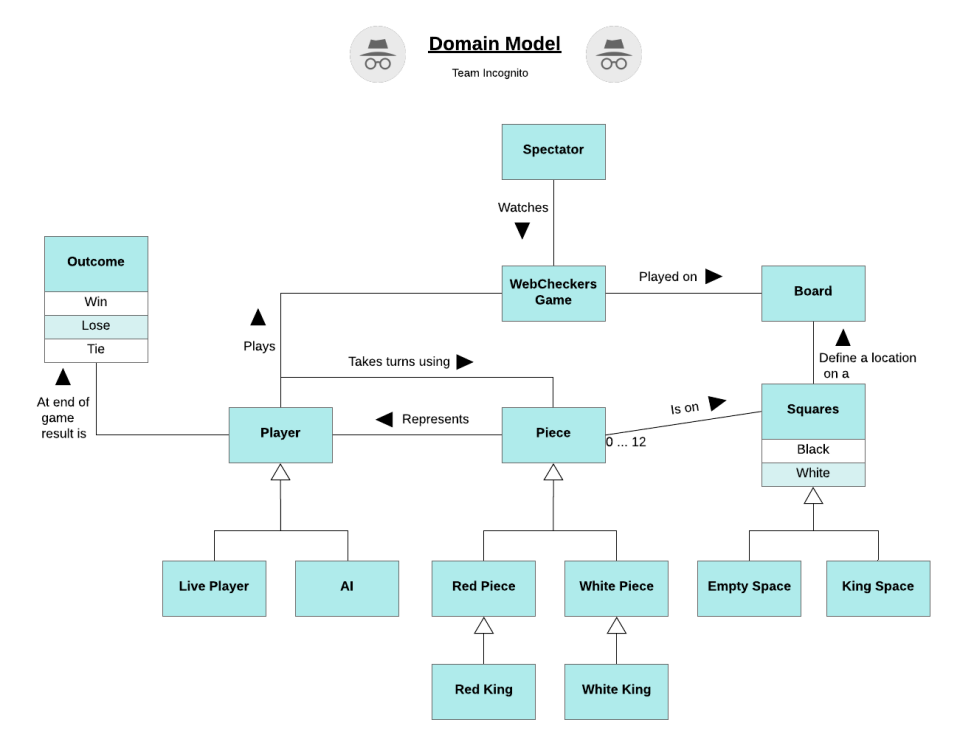
MVP Features

> *Sign In*
>
> > - *A player must be able to sign into WebCheckers.*
>
> *Start Game*
>
> > - *A Player must be able to start a game with an opponent.*
>
> *Sign Out*
>
> > - *A Player must be able to sign out of a game.*
>
> *Resignation*
>
> > - *A player must be able to resign from a game when it is their turn.*
>
> *Game Play*
>
> > - *A player must be able to play a game with an opponent.*

Roadmap of Enhancements

> *Spectator Mode*
>
> > - *A player is able to spectate a live game.*
> > - *A player is not able to be put in a game when spectating.*
> > - *A player is not able to make any moves while spectating.*
> > - *A player can see the time the last move was made.*
>
> *AI*
>
> > - *A player is able to play a game with an AI.*
> > - *Multiple players can play an AI game.*
> > - *The AI is able to jump, move, and be kinged on the board.*

Application Domain



> *The domain model provides a common understanding between a customer and the developers of the scope and major entities that exist in the system.*

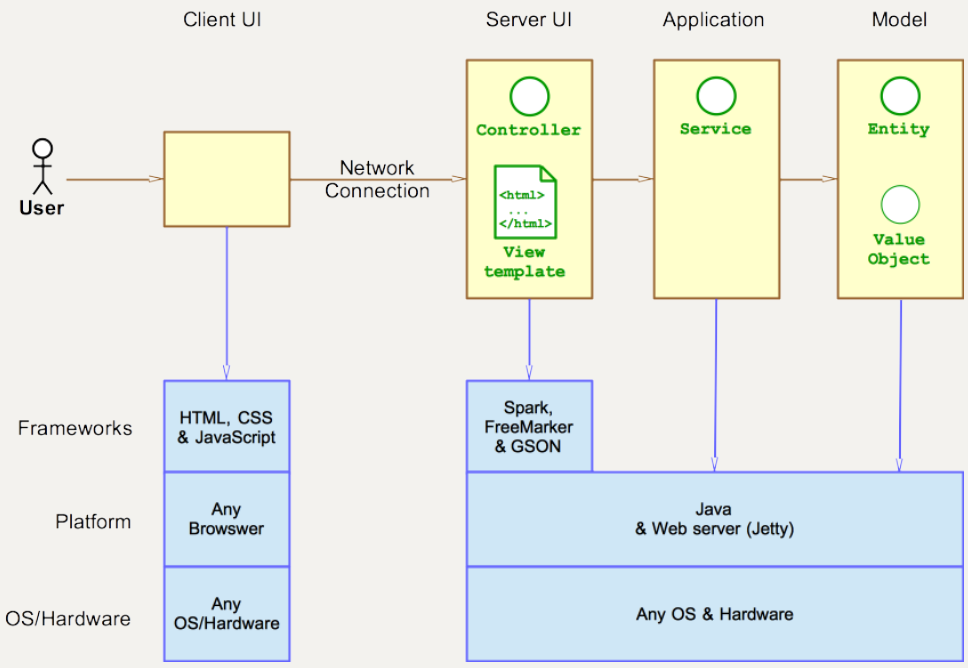
> *We have created our domain model in hopes to help the customer understand what we was trying to accomplish as we begin our development process. Within our domain model we have included the following elements:* ***Domain Model Overview***
>
> > - *Spectator that watches the WebCheckers game*

> - *WebCheckers game that is played on a Board*
> - *Squares that defines a location on the Board*
> - *Piece(s) that are on a Square*
> - *Piece that represents a player*
> - *Player taking turns using a Piece*
> - *Player playing WebCheckers*
> - *Player getting end of game result*

Architecture and Design

Summary

The following Tiers/Layers model shows a high-level view of the webapp's architecture.



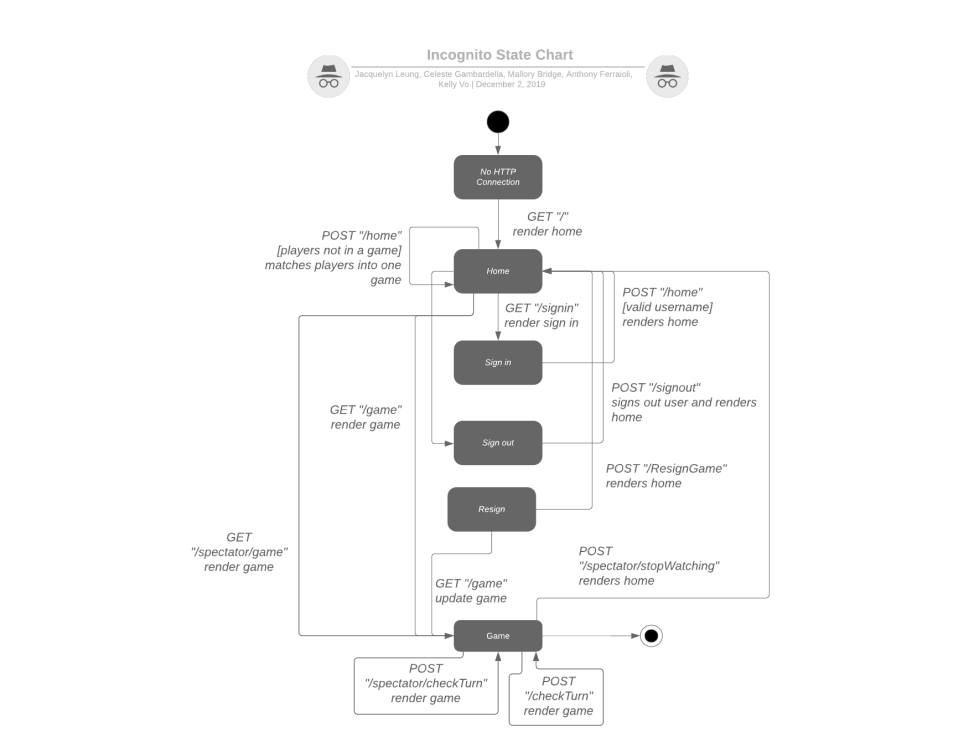
As a web application, the user interacts with the system using a browser. The client-side of the UI is composed of HTML pages with some minimal CSS for styling the page. There is also some JavaScript that has been provided to the team by the architect.

The server-side tiers include the UI Tier that is composed of UI Controllers and Views. Controllers are built using the Spark framework and View are built using the FreeMarker framework. The Application and Model tiers are built using plain-old Java objects (POJOs).

Details of the components within these tiers are supplied below.

Overview of User Interface

This section describes the web interface flow; this is how the user views and interacts with the WebCheckers application.

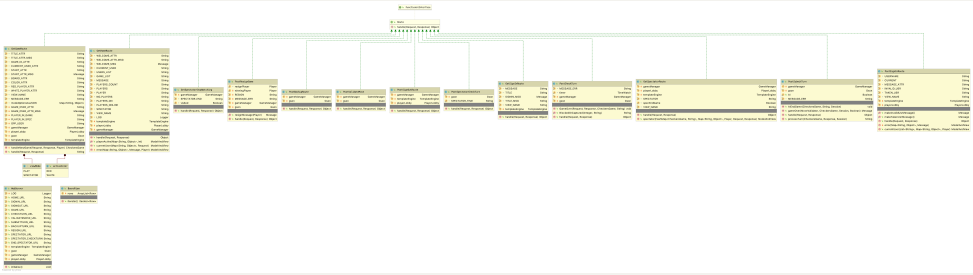


> *Since a web page can only have one state or page at a given time we have designed a flowchart of where each page should go to after a certain amount of time or when the user interacts with the web application.*
>
> *No HTTP Connection*
>
> - *Begin a connection when the user goes to the web page.*
>
> *Home*
>
> - *The user will see the welcome/home page.*
> - *The number of players that are already signed in are displayed.*
>
> *Sign In*
>
> - *They are given an option to sign in.*
> - *If they choose to sign in a new page will be rendered with text boxes asking for a username and password.*
> - *If the username or password is invalid the user will see an error message appear on the page.*

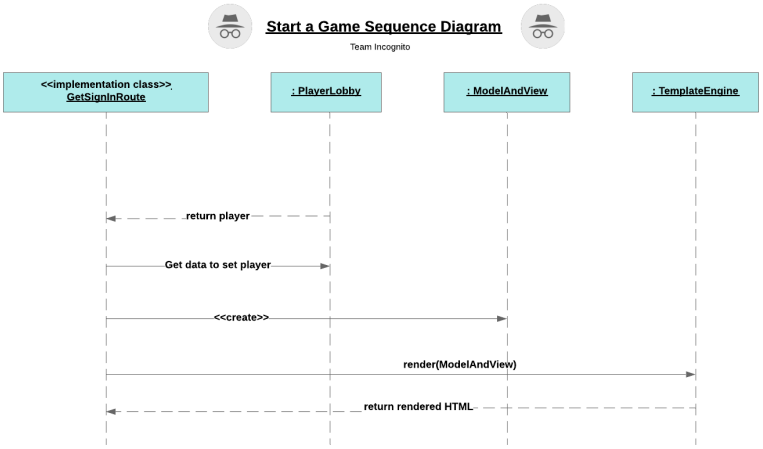
> *Home*
>
> - *If a valid username and password the user will be redirected make to the homepage and be able to see the other players logged into the game.*
> - *Now the user will be able to select another player which will redirect a new game page.*
>
> *Game*
>
> - *The other player selected will also be redirected to the same game.*

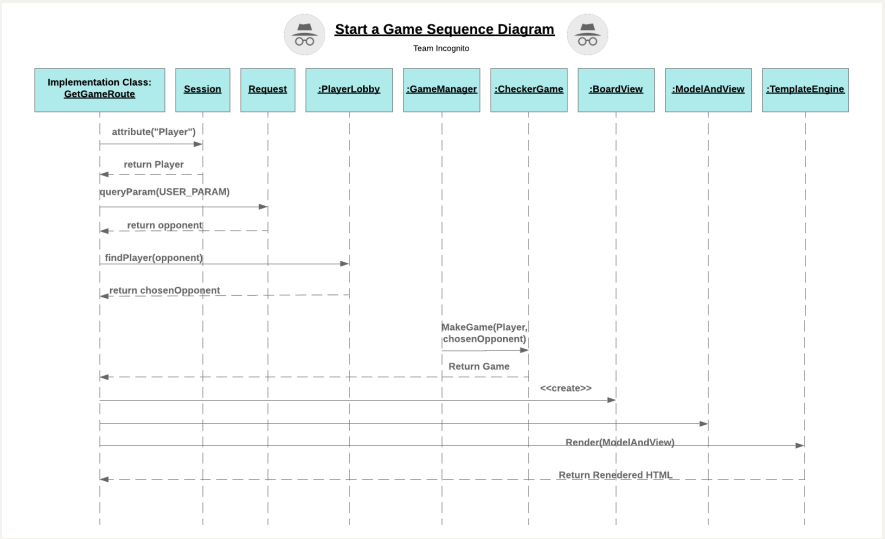
UI Tier

> *This level of the application contains everything the user will interact with. It will interact with the Application and Model tiers. The following UML diagram shows the classes used in the UI*



> *A Sequence diagram of the start a game and sign in was created to showcase a better understanding of the functionality of how start a game and sign in works.*

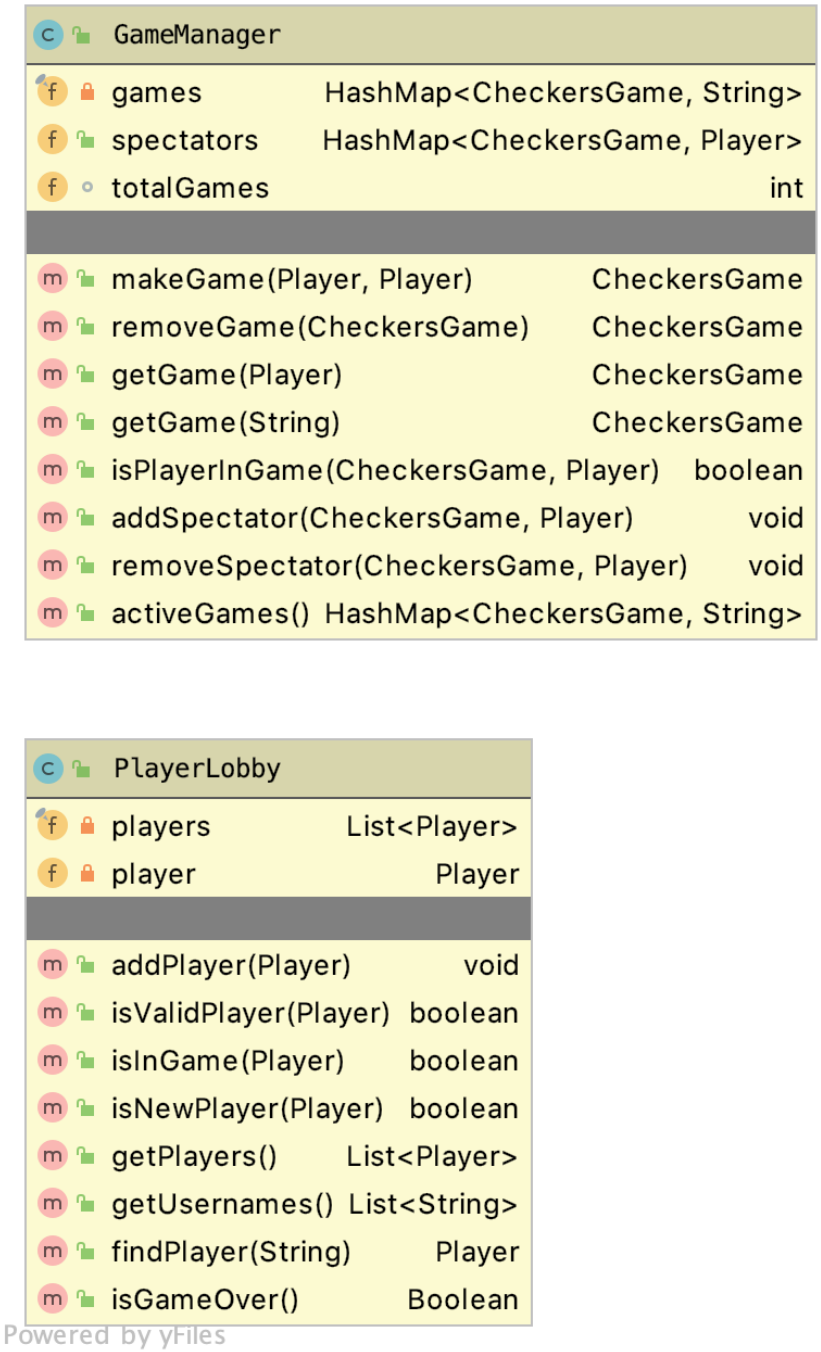




> *These diagrams breaks down what each class handles or is expecting and how they connect to each other. You are able to follow the arrows to see how GetGameRoute uses other aspects to get a game and show it to the user.*

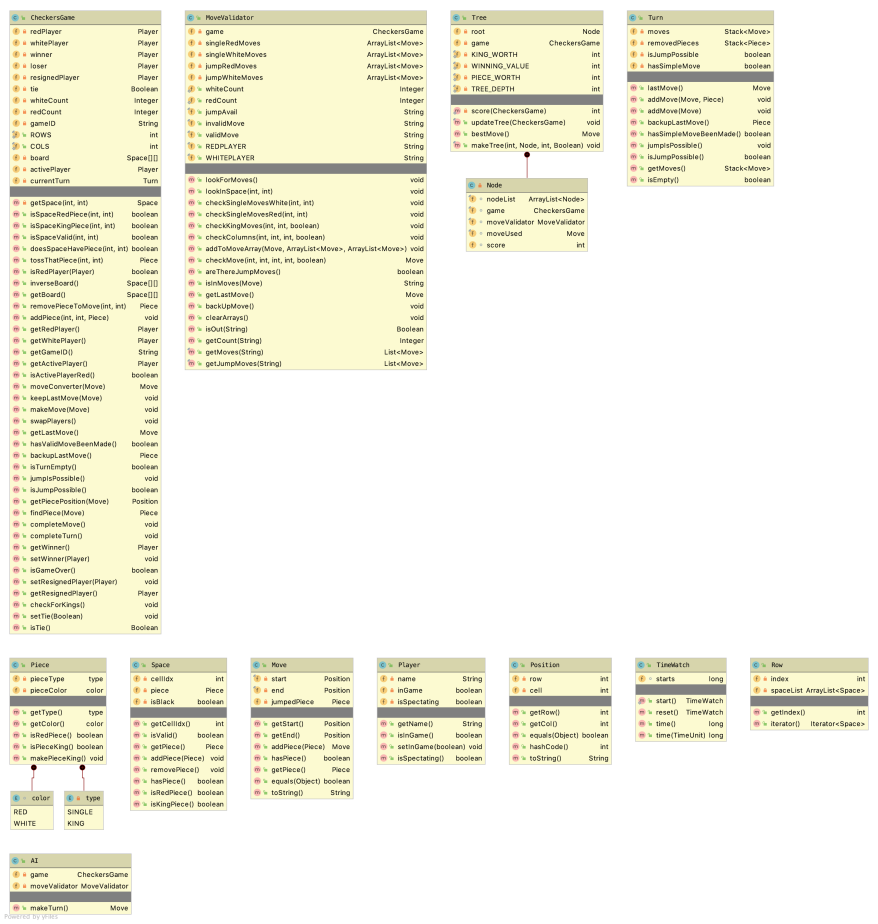
Application Tier

> *The Application Tier provides support and functionality for the game. It is designed to handle the logic of game and act as the middle man between the Model Tier and the UI Tier. The classes currently stored within the Application Tier in the UML diagram below.*



Model Tier

> *The model tier is the basic structure for the game. We use a Board View, Checkers Game, Move, Piece, Player, Position, Row, Space and Turn in our model tier. The classes that are within the Model tier are shown in the UML Diagram below.*



Design Improvements

> *To improve our design we should adhere to Object-Oriented design principles more. To adhere to the controller principle we should ensure that we do not make multiple unnecessary controllers and try to combine controllers together. For example keeping Signing in and out in the same controller. To adhere to the polymorphism principle we should use method overloading to have different arguments within the parameters. We currently do not adhere to the Liskov Principle and to do this we can add in pre and post conditions that will help improve our adherence as well as having subclasses properly extending super classes will allow our team to have more adherence to the principle.*
>
> *To improve our usage of the open/closed principle, we can create more classes as the base for other classes that share common properties. This would help limit the code in each class and lower cohesion. We also currently do not follow pure fabrication. We can use pure fabrication in the future for our design to lower cohesion and clean up the overall readability with our code. Classes can be made to calculate or examine movement options, or just test the possibility of decisions made by a player in the game. They would have no physical representation on the UI, but they would be referenceable in multiple scenarios and aid future class codings.*

Testing

Acceptance Testing

> *We had acceptance criteria that we used to create acceptance tests in Spring 1 and 2. During Sprint 1, we were unable to finish the acceptance criteria that stated that a user would be able to be automatically directed to start a game. We were able to pass all other acceptance criteria tests in Sprint 1. In Sprint 2, we were able to pass all acceptance criteria from Sprint 1 as well as new acceptance criteria from other user stories added to the game.*

User Story	Acceptance Criterion	Sprint 1	Tester initials; date; comments (required if test failed)	Sprint 2	Tester initials; date; comments (required if test failed)	Sprint 3	Tester initials; date; comments (required if test failed)
As a Player I want to sign-in so that can play a game of checkers.	Given that I have not yet signed in when I see the Home page then I must see a means to sign-in. (such as a link or button)	Pass	JL, 10/10, I see the sign in button that I can click on.	Pass	JL; 11/2	Pass	JL; 11/24
	Given that I am not signed-in when I do click on the sign-in link then I expect to be taken to the Sign-in page, with a means to enter a player name.	Pass	JL, 10/10 I see the sign in button that I can click on. And I can create my own user name and password	Pass	JL; 11/2	Pass	JL; 11/24
	Given that no one else is using my name when I enter my name containing at least one alphanumeric character and optionally spaces in the sign-in form and click the Sign-in button then I expect the system to reserve my name and navigate back to the Home page.	Pass	JL, 10/10 I was redirected to the homepage and my name was saved	Pass	JL; 11/2	Pass	JL; 11/24
	Given that I am on the Sign-in page when I enter a name that does not contain at least one alphanumeric character or contains one or more characters that are not alphanumeric or spaces and click the Sign-in button then I expect the system to reject this name and return the Sign-in page.	Pass	MB, 10/10	Pass	JL; 11/2	Pass	JL; 11/24
	Given that someone else with my name has signed-in when I enter my name in the Sign-in form and click the Sign-in button then I expect the system to reject my sign-in and return the Sign-in page for me to try a different name.	Pass	JL, 10/10 I was redirected to the sign in page when I used someone elses username	Pass	JL; 11/2	Pass	JL; 11/24
	Given that I am signed-in when I navigate to the Home page then I expect to see a list of all other signed-in players. (NOTE: In the next story you will use this list to pick opponents for checkers	Pass	MB, 10/10	Pass	JL; 11/2	Pass	JL; 11/24
	Given that I am not signed-in when I navigate to the Home page then I expect to see a message of how many players are signed-in but not a list of them (for privacy reasons).	Pass	MB, 10/10	Pass	JL; 11/2	Pass	JL; 11/24
As a Player I want to start a game so that I can play checkers with an opponent.	Given that I'm signed in when I view the Home page then I can start a game by selecting a player listed on the Home page.	Pass	CG/KV, 10/10	Pass	JL; 11/2	Pass	JL; 11/24
	Given that the player I selected isn't yet in a game when I select that player then the system will begin a checkers game and assign me as the starting (Red) player and my opponent as the White	Pass	CG/KV, 10/10	Pass	JL; 11/2	Pass	JL; 11/24
	Given that the player I selected is already in a game when I select that player then the system will return me to the Home page with an error message.	Pass	CG/KV, 10/10	Pass	JL; 11/2	Pass	JL; 11/24
	Given that I'm waiting for a game when another player selects a game with me then the system will automatically send me to the Game View as the White player. NOTE: the `home.fit` HTML includes a `<meta>` tag that tells the browser to refresh the game every 5 seconds; thus you need to update the `GetHomeRoute` controller to handle the situation when a player is assigned a game.	Fail	CG/KV, 10/10 Didn't update opponet's game view screen	Pass	CG/KV, 10/27	Pass	JL; 11/24
	Given a valid, initial game board when I drag a piece to a white space then the piece should not be droppable.	Pass	JL, 10/10 I am unable to drop the piece	Pass	JL; 11/2	Pass	JL; 11/24
	Given a valid, initial game board when I drag a piece to an occupied space then the piece should not be droppable.	Pass	JL, 10/10 I am unable to drop the piece	Pass	JL; 11/2	Pass	JL; 11/24
	Given a valid, initial game board when I drag a piece to an open space then the piece should be droppable. NOTE: In this Story the drop action should not do anything; piece movement will be the focus of future stories.	Pass	JL, 10/10 It is droppable, but there is an error saying it was not implemented yet	Pass	JL; 11/2	Pass	JL; 11/24
	Given a valid, initial game board when I view the board in the browser then my pieces are oriented on the bottom of the board grid just like it would see the board if I were playing in the physical world.	Pass	JL, 10/10 I see on the board my pieces are at the bottom	Pass	JL; 11/2	Pass	JL; 11/24

As a player, I want to sign out so that I can exit a game of checkers	Given an exisiting user when I sign out then I want to return to the home page			Pass	CG, 10/31	Pass	JL; 11/24
	Given that I am signed in when I finish a game then I must see a means to sign-out. (such as a link or button)					Pass	JL; 11/24
As a player, I want to make a single move so that I can move and play across the board				Pass	JL; 11/2	Pass	JL; 11/24
As a player, I want to make a single jump move so that I can capture an opponents piece				Pass	JL; 11/3	Pass	JL; 11/24
As a player, I want to make a double jump move so that I can capture multiple opponent pieces				Pass	JL; 11/3	Pass	JL; 11/24
As a player, I want to be kinged so I can move and jump diagonally forward and backwards				Pass	JL; 11/2	Pass	JL; 11/24
As a player, I want to only be allowed to jump when it is available.				Pass	JL; 11/3	Pass	JL; 11/24
As a player, I want to jump more than 2 times if it is available				Pass	JL; 11/3	Pass	JL; 11/24
As a player, If I want to make a double jump I wont be able to submit move in between jump				Fail	JL; 11/3, does not currently work	Pass	JL; 11/24
As a player, when I am kinged I want my move to be finished and I want to no longer be able to move.				Fail	JL; 11/3; move can go further after reaching king space	Pass	JL; 11/24
As a player, I want to be assigned to a color so that I know what player I				Pass	JL; 11/3	Pass	JL; 11/24
As a Player I want to know if another player has left the game so that I can play another game.				Fail	JL; 11/3, does not currently work	Pass	JL; 11/24
As a player, I want to forfeit so I can resign the game				Fail	JL; 11/3 fails but does not fail correctly	Pass	JL; 11/24
As a spectator I want to login so that I can watch a game of watchcheckers.						Pass	JL; 11/24
As a spectator, I do not want to be asked to play in a game						Pass	JL; 11/24
As a spectator I want to see want games are happening so that I can choose who to watch.						Pass	JL; 11/24
As a player, I want to play with an AI						Pass	JL; 11/24
As a player, I want to see a message of how I won or lost when I lose or win a game						Pass	JL; 11/24

Unit Testing and Code Coverage

> *Each team member created unit tests to test various parts of the program. Initially each member tested an item from the UI and something from the application or model tier. Afterwards, team members created more tests to try to gain high code coverage. We aimed for above 95% code coverage in the application model for the lines and methods. For the model tier we aimed for 95% code coverage and for the UI Tier we aimed for 90% code coverage. The figures shown below are the Jacoco analysis from the tests.*

[all classes]

Overall Coverage Summary

Package	Class, %	Method, %	Line, %
all classes	100% (34/ 34)	96.9% (187/ 193)	90.5% (960/ 1061)

Coverage Breakdown

Package	Class, %	Method, %	Line, %
com.webcheckers.appl	100% (2/ 2)	100% (18/ 18)	95.4% (62/ 65)
com.webcheckers.model	100% (15/ 15)	95.9% (117/ 122)	95.7% (468/ 489)
com.webcheckers.ui	100% (17/ 17)	98.1% (52/ 53)	84.8% (430/ 507)

generated on 2019-11-24 15:06

[all classes] [com.webcheckers.appl]

Coverage Summary for Package: com.webcheckers.appl

Package	Class, %	Method, %	Line, %
com.webcheckers.appl	100% (2/ 2)	100% (18/ 18)	95.4% (62/ 65)

Class	Class, %	Method, %	Line, %
GameManager	100% (1/ 1)	100% (9/ 9)	94.4% (34/ 36)
PlayerLobby	100% (1/ 1)	100% (9/ 9)	96.6% (28/ 29)

generated on 2019-11-24 15:06

[all classes] [com.webcheckers.model]

Coverage Summary for Package: com.webcheckers.model

Package	Class, %	Method, %	Line, %
com.webcheckers.model	100% (15/ 15)	95.9% (117/ 122)	95.7% (468/ 489)

Class	Class, %	Method, %	Line, %
AI	100% (1/ 1)	100% (2/ 2)	100% (7/ 7)
CheckersGame	100% (1/ 1)	94.9% (37/ 39)	97.9% (141/ 144)
Move	100% (1/ 1)	100% (9/ 9)	100% (24/ 24)
MoveValidator	100% (1/ 1)	100% (19/ 19)	95.7% (135/ 141)
Piece	100% (3/ 3)	87.5% (7/ 8)	91.7% (11/ 12)
Player	100% (1/ 1)	80% (4/ 5)	87.5% (7/ 8)
Position	100% (1/ 1)	100% (6/ 6)	92.9% (13/ 14)
Row	100% (1/ 1)	100% (3/ 3)	100% (9/ 9)
Space	100% (1/ 1)	100% (9/ 9)	100% (22/ 22)
TimeWatch	100% (1/ 1)	80% (4/ 5)	88.9% (8/ 9)
Tree	100% (2/ 2)	100% (7/ 7)	88.6% (62/ 70)
Turn	100% (1/ 1)	100% (10/ 10)	100% (29/ 29)

generated on 2019-11-24 15:06

[all classes] [com.webcheckers.ui]

Coverage Summary for Package: com.webcheckers.ui

Package	Class, %	Method, %	Line, %
com.webcheckers.ui	100% (17/ 17)	98.1% (52/ 53)	84.8% (430/ 507)

Class	Class, %	Method, %	Line, %
BoardView	100% (1/ 1)	100% (2/ 2)	100% (12/ 12)
GetGameRoute	100% (3/ 3)	100% (6/ 6)	73% (54/ 74)
GetHomeRoute	100% (1/ 1)	83.3% (5/ 6)	71% (49/ 69)
GetSignInRoute	100% (1/ 1)	100% (3/ 3)	100% (8/ 8)
GetSpectatorRoute	100% (1/ 1)	100% (3/ 3)	90.2% (46/ 51)
GetSpectatorStopWatching	100% (1/ 1)	100% (3/ 3)	90% (9/ 10)
PostBackupRoute	100% (1/ 1)	100% (2/ 2)	100% (12/ 12)
PostCheckTurn	100% (1/ 1)	100% (4/ 4)	97.9% (46/ 47)
PostResignGame	100% (1/ 1)	100% (3/ 3)	100% (27/ 27)
PostSignInRoute	100% (1/ 1)	100% (6/ 6)	91.9% (34/ 37)
PostSignOutRoute	100% (1/ 1)	100% (2/ 2)	100% (18/ 18)
PostSpectatorCheckTurn	100% (1/ 1)	100% (2/ 2)	90% (9/ 10)
PostSubmitTurn	100% (1/ 1)	100% (6/ 6)	83% (73/ 88)
PostValidateMove	100% (1/ 1)	100% (2/ 2)	42.1% (8/ 19)
WebServer	100% (1/ 1)	100% (3/ 3)	100% (25/ 25)

generated on 2019-11-24 15:06

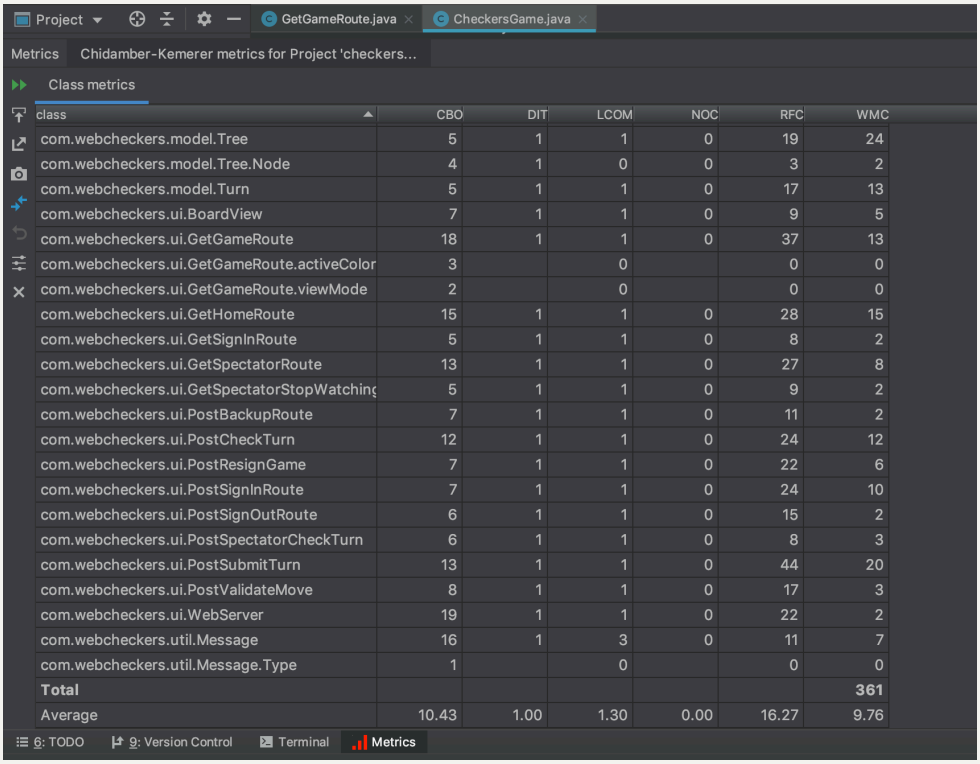
Code Metric Analysis

> *Code metrics produces a numerical measurement of a characteristic. This allows decisions to be made within the code. Our current code metrics is showcased below. Almost all metrics were in the preset range and there were hardly any measurements that did not meet the target. On the Complexity metrics, there were 7 functions that went over the target. The following were the functions that did not meet the target:*

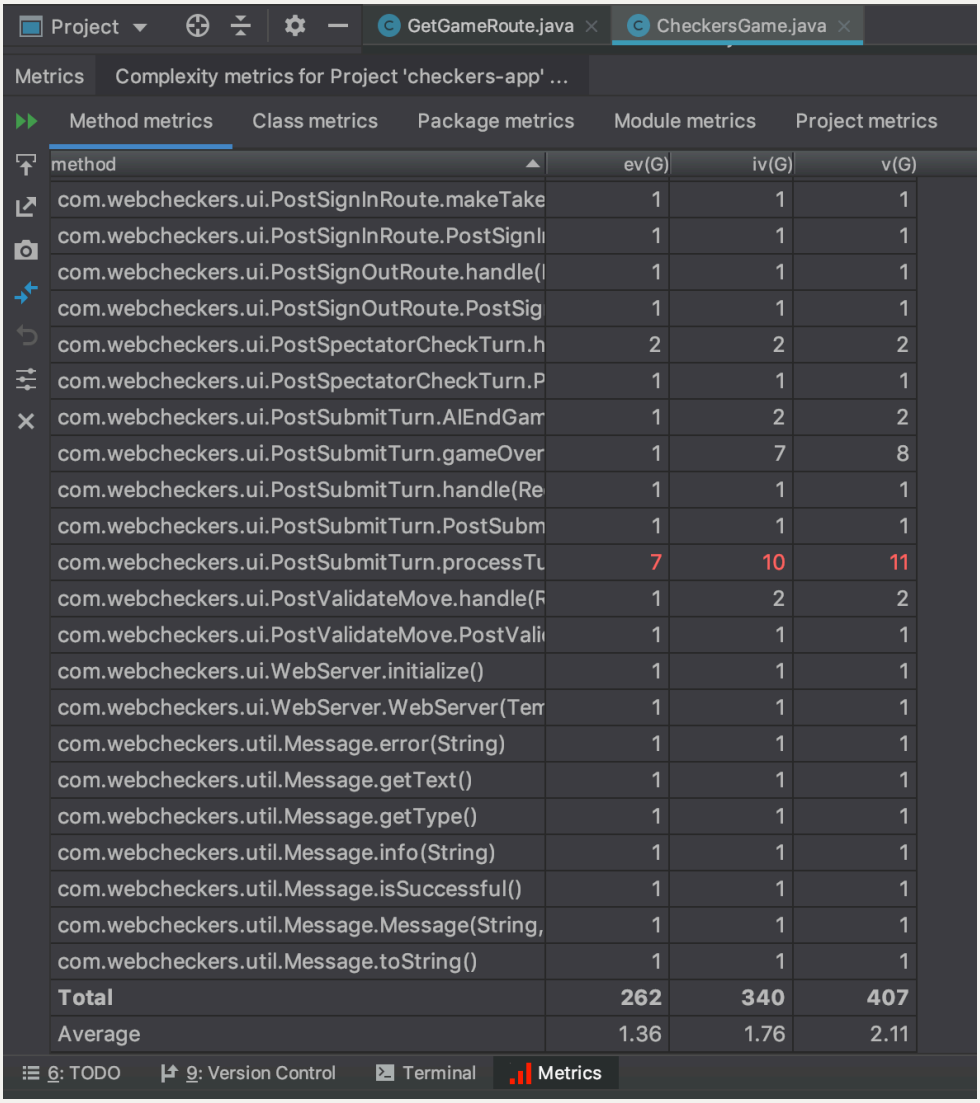
> - *Model.MoveValidator.checkMove*
>
>> - *The ev(G) and v(G) were out of bounds*
> - *Model.MoveValidator.isInMoves*
> - *The ev(G) was out of bounds*
>
> - *Model.Tree.makeTree*
>
>> - *the v(G) was out of bounds*
>
> - *UI.GetGameRoute.handleNewGame*
>
>> - *The ev(G) was out of bounds*
>
> - *UI.GetHomeRoute.handle*

> - *The ev(G) was out of bounds*
>
> - *UI.PostCheckTurn.handle*
>
>> - *The ev(G) was out of bounds*
>
> - *UI.PostSubmitTurn.processTurn*
>
>> - *The ev(G), iv(G), and v(G) were out of bounds*

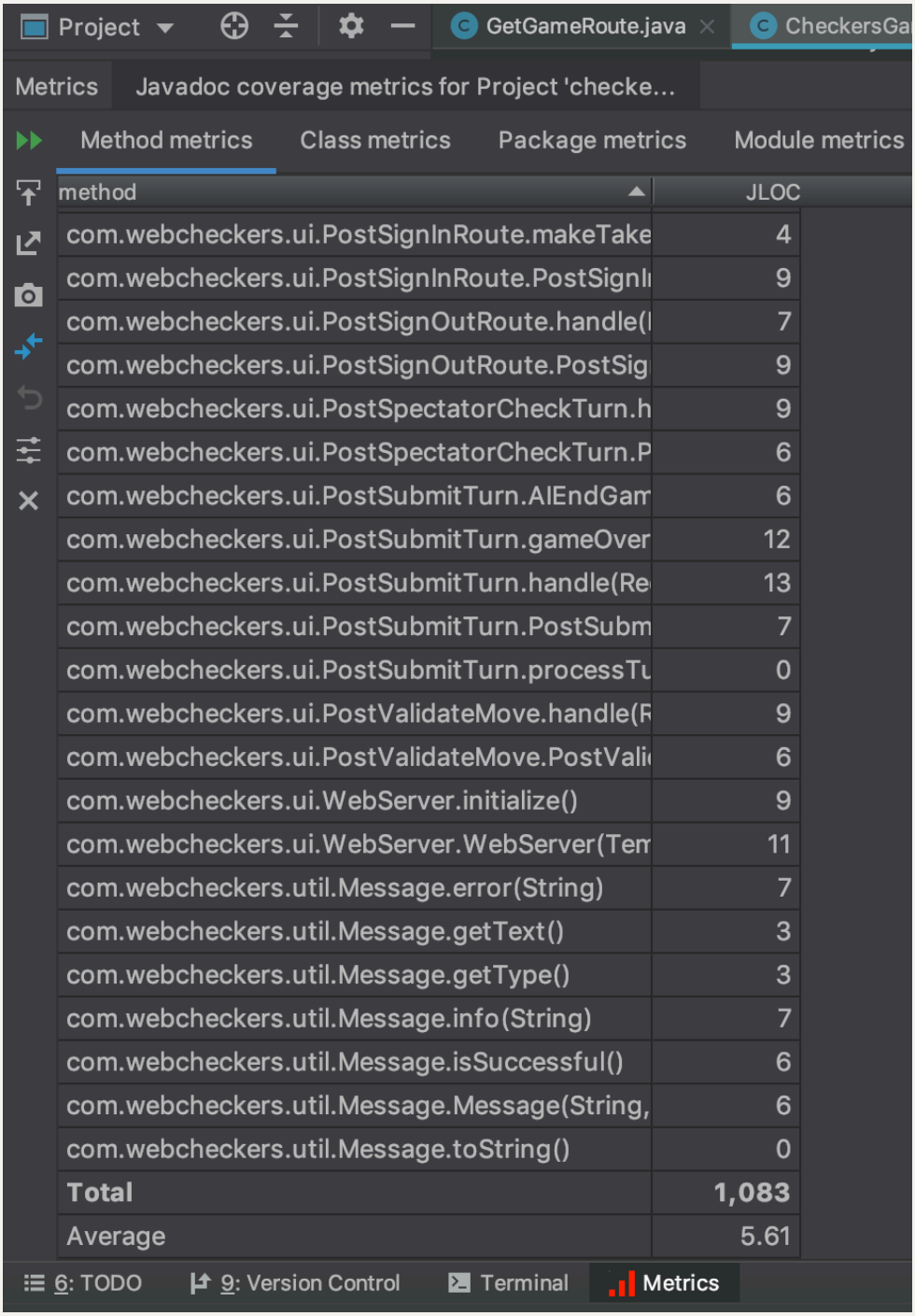
> *Looking into these metrics, these functions had high complexity due to their use within the code. To improve the complexity, we should use more helper functions and create different functions to help lesson the use of the functions that had high complexity. Some functions like handle in the UI cannot be improved due to their use being needed for the UI and unable to replicate or use other functions.*

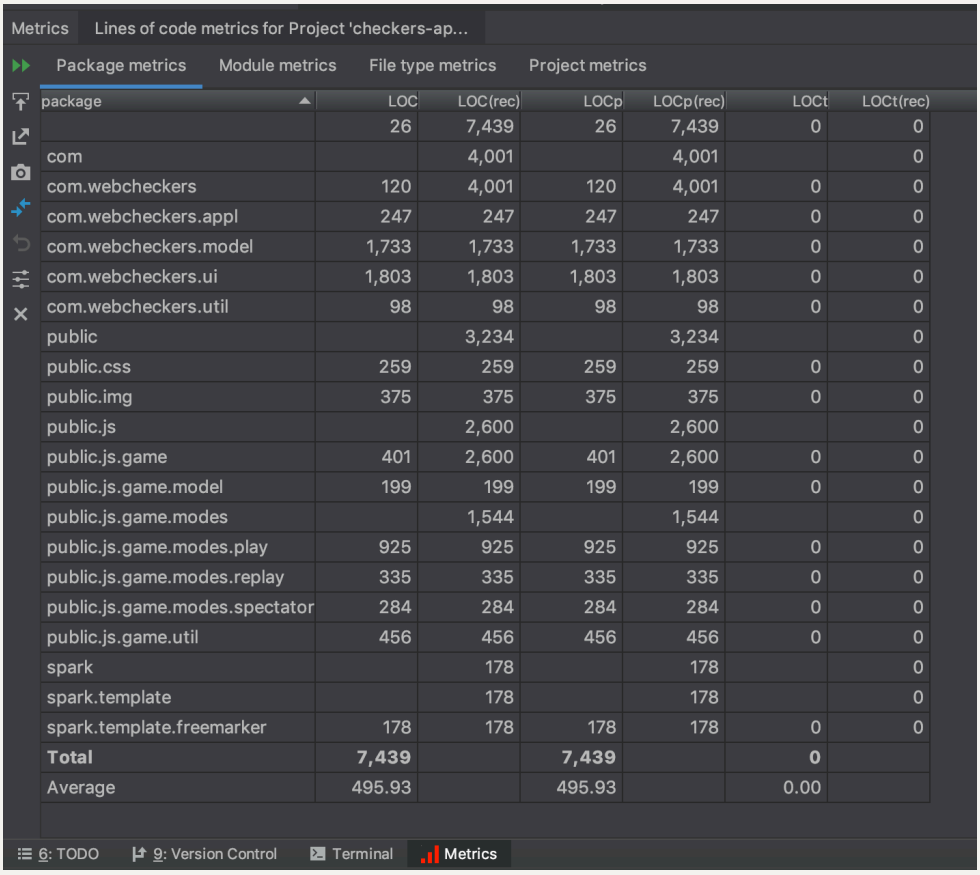


class	CBO	DIT	LCOM	NOC	RFC	WMC
com.webcheckers.model.Tree	5	1	1	0	19	24
com.webcheckers.model.Tree.Node	4	1	0	0	3	2
com.webcheckers.model.Turn	5	1	1	0	17	13
com.webcheckers.ui.BoardView	7	1	1	0	9	5
com.webcheckers.ui.GetGameRoute	18	1	1	0	37	13
com.webcheckers.ui.GetGameRoute.activeColor	3		0		0	0
com.webcheckers.ui.GetGameRoute.viewMode	2		0		0	0
com.webcheckers.ui.GetHomeRoute	15	1	1	0	28	15
com.webcheckers.ui.GetSignInRoute	5	1	1	0	8	2
com.webcheckers.ui.GetSpectatorRoute	13	1	1	0	27	8
com.webcheckers.ui.GetSpectatorStopWatching	5	1	1	0	9	2
com.webcheckers.ui.PostBackupRoute	7	1	1	0	11	2
com.webcheckers.ui.PostCheckTurn	12	1	1	0	24	12
com.webcheckers.ui.PostResignGame	7	1	1	0	22	6
com.webcheckers.ui.PostSignInRoute	7	1	1	0	24	10
com.webcheckers.ui.PostSignOutRoute	6	1	1	0	15	2
com.webcheckers.ui.PostSpectatorCheckTurn	6	1	1	0	8	3
com.webcheckers.ui.PostSubmitTurn	13	1	1	0	44	20
com.webcheckers.ui.PostValidateMove	8	1	1	0	17	3
com.webcheckers.ui.WebServer	19	1	1	0	22	2
com.webcheckers.util.Message	16	1	3	0	11	7
com.webcheckers.util.Message.Type	1		0		0	0
Total						**361**
Average	10.43	1.00	1.30	0.00	16.27	9.76

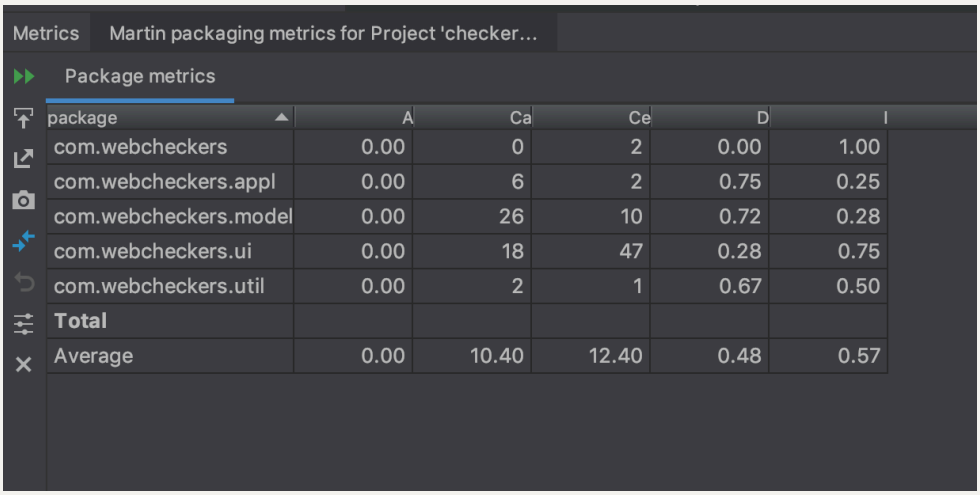


method	ev(G)	iv(G)	v(G)
com.webcheckers.ui.PostSignInRoute.makeTake	1	1	1
com.webcheckers.ui.PostSignInRoute.PostSignI	1	1	1
com.webcheckers.ui.PostSignOutRoute.handle(	1	1	1
com.webcheckers.ui.PostSignOutRoute.PostSig	1	1	1
com.webcheckers.ui.PostSpectatorCheckTurn.h	2	2	2
com.webcheckers.ui.PostSpectatorCheckTurn.P	1	1	1
com.webcheckers.ui.PostSubmitTurn.AIEndGam	1	2	2
com.webcheckers.ui.PostSubmitTurn.gameOver	1	7	8
com.webcheckers.ui.PostSubmitTurn.handle(Re	1	1	1
com.webcheckers.ui.PostSubmitTurn.PostSubm	1	1	1
com.webcheckers.ui.PostSubmitTurn.processTu	7	10	11
com.webcheckers.ui.PostValidateMove.handle(R	1	2	2
com.webcheckers.ui.PostValidateMove.PostVali	1	1	1
com.webcheckers.ui.WebServer.initialize()	1	1	1
com.webcheckers.ui.WebServer.WebServer(Tem	1	1	1
com.webcheckers.util.Message.error(String)	1	1	1
com.webcheckers.util.Message.getText()	1	1	1
com.webcheckers.util.Message.getType()	1	1	1
com.webcheckers.util.Message.info(String)	1	1	1
com.webcheckers.util.Message.isSuccessful()	1	1	1
com.webcheckers.util.Message.Message(String,	1	1	1
com.webcheckers.util.Message.toString()	1	1	1
Total	**262**	**340**	**407**
Average	1.36	1.76	2.11


Project
GetGameRoute.java
CheckersGa
Metrics
Javadoc coverage metrics for Project 'checke...
Method metrics
Class metrics
Package metrics
Module metrics
method
JLOC
com.webcheckers.ui.PostSignInRoute.makeTake 4
com.webcheckers.ui.PostSignInRoute.PostSignI 9
com.webcheckers.ui.PostSignOutRoute.handle(7
com.webcheckers.ui.PostSignOutRoute.PostSig 9
com.webcheckers.ui.PostSpectatorCheckTurn.h 9
com.webcheckers.ui.PostSpectatorCheckTurn.P 6
com.webcheckers.ui.PostSubmitTurn.AIEndGam 6
com.webcheckers.ui.PostSubmitTurn.gameOver 12
com.webcheckers.ui.PostSubmitTurn.handle(Re 13
com.webcheckers.ui.PostSubmitTurn.PostSubm 7
com.webcheckers.ui.PostSubmitTurn.processTu 0
com.webcheckers.ui.PostValidateMove.handle(R 9
com.webcheckers.ui.PostValidateMove.PostVali 6
com.webcheckers.ui.WebServer.initialize() 9
com.webcheckers.ui.WebServer.WebServer(Ter 11
com.webcheckers.util.Message.error(String) 7
com.webcheckers.util.Message.getText() 3
com.webcheckers.util.Message.getType() 3
com.webcheckers.util.Message.info(String) 7
com.webcheckers.util.Message.isSuccessful() 6
com.webcheckers.util.Message.Message(String, 6
com.webcheckers.util.Message.toString() 0
Total 1,083
Average 5.61
6: TODO
9: Version Control
Terminal
Metrics



package	LOC	LOC(rec)	LOCp	LOCp(rec)	LOCt	LOCt(rec)
	26	7,439	26	7,439	0	0
com		4,001		4,001		0
com.webcheckers	120	4,001	120	4,001	0	0
com.webcheckers.appl	247	247	247	247	0	0
com.webcheckers.model	1,733	1,733	1,733	1,733	0	0
com.webcheckers.ui	1,803	1,803	1,803	1,803	0	0
com.webcheckers.util	98	98	98	98	0	0
public		3,234		3,234		0
public.css	259	259	259	259	0	0
public.img	375	375	375	375	0	0
public.js		2,600		2,600		0
public.js.game	401	2,600	401	2,600	0	0
public.js.game.model	199	199	199	199	0	0
public.js.game.modes		1,544		1,544		0
public.js.game.modes.play	925	925	925	925	0	0
public.js.game.modes.replay	335	335	335	335	0	0
public.js.game.modes.spectator	284	284	284	284	0	0
public.js.game.util	456	456	456	456	0	0
spark		178		178		0
spark.template		178		178		0
spark.template.freemarker	178	178	178	178	0	0
Total	**7,439**		**7,439**		**0**	
Average	495.93		495.93		0.00	



package	A	Ca	Ce	D	I
com.webcheckers	0.00	0	2	0.00	1.00
com.webcheckers.appl	0.00	6	2	0.75	0.25
com.webcheckers.model	0.00	26	10	0.72	0.28
com.webcheckers.ui	0.00	18	47	0.28	0.75
com.webcheckers.util	0.00	2	1	0.67	0.50
Total					
Average	0.00	10.40	12.40	0.48	0.57