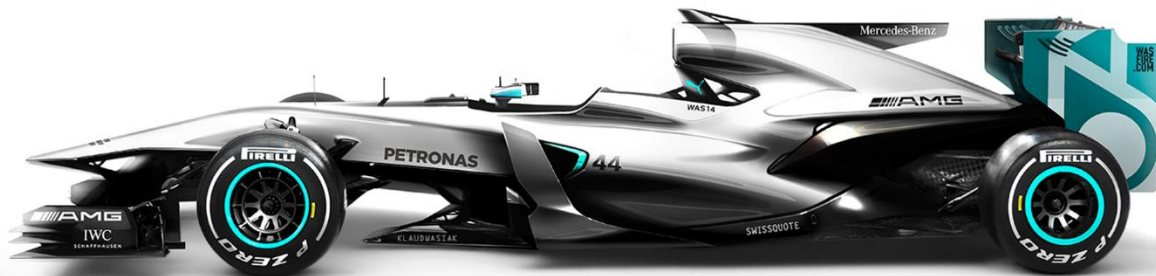


Elias Ait Larbi - eaitlarbi
Tim Pelser - tpelser
Wouter van Lil - wvanlil
David Happel – dvhappel
30/1/2017

F1 Manager

F1 Manager *TU Delft edition*



"You don't drive races on paper"

General

We were asked to build a F1 manager game in java. You should be able to buy drivers and engines, change strategy, upgrade staff, manage a budget and run a race. There is a total of five sprints of two weeks. At the end of three of those sprints there was a demo. First we created a UML to generally map out the structure of our code. For version control we used Github with Zenhub as an extension. We also used a slack channel and a Whatsapp group to communicate. Generally the collaboration went well. All of us listened to others and were heard. For us version control was a necessary evil. The first few weeks were almost solely devoted to getting to know Git, Github and maven. Even though a lot of time was spent managing/assigning issues and trying to merge branches, without Git we could not have made the game.

Design process

Problems: On our way to a finished project, we encountered a lot of problems. For example, a problem that has taken a lot of time to solve is the “infinite json loop” problem. We have a team with a list of drivers as an attribute. These drivers in turn had a team as one of their attributes. When saving the game, these classes had to be converted to json. But because of this circular reference this could not be done. A lot of things were tried to fix this but nothing worked as we wanted it to. Eventually we decided to change the driver.team attribute to a string: driver.teamid. The problem with this is that we had to change a lot of our already written code to make this work, which took a lot of time. Another problem we ran into had to do with reaching classes. We never really thought about making a central place where we could reference to our player list and save/load methods. Our solution was a central Game class that has static methods(load/new game) and could also be instantiated with attributes like the list with drivers and the list with all teams. Basically “the game state”. This “game” object is first instantiated in the controller and then assigned to the static “main” class so it could be referenced by any class. At the end of week 7 we had some branches that had a lot of improvements over the master but were never merged. Those branches had stepped off of the idea of “one branch per issue”. These branches became very hard to merge, because they differed too much from the master. This was not very productive when you wanted to add upon the newly added changes of those branches. Because you could not simply split a new branch from the master.

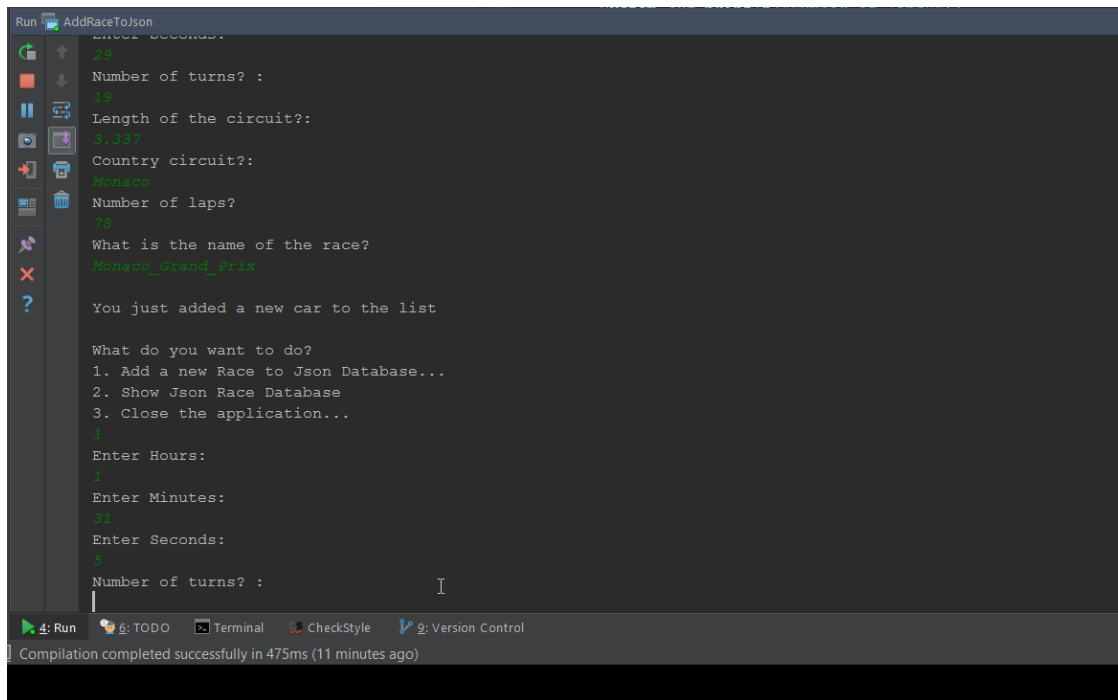
Technologies: *Graphics:* For graphics we considered a lot of options but we decided to use javaFX pretty quickly because there seemed to be the most documentation and forum posts about it. Also both the presentation and our Project mentor mentioned javaFX as a good choice.

File System: For saving our files json was the obvious choice. It is easily readable and converts very easily to java classes with attributes. For this conversion we settled with Gson. Gson is made by google and is used a lot to handle json files in java. Both of these facts made us confident that Gson is a solid and well-made tool to reliably convert to and from json.

Testing: To test our application we went with Junit. We all were already familiar with Junit for testing because of the Object Oriented Programming course in the first quarter. We did however need some more functionality to test some methods. That is why we also used Mockito to mock classes in our tests.

Other: We also used a framework called “Guava” to be able to easily implement preconditions to methods in our code. This makes our code a lot more reliable and readable.

Extra's: Tools: To make our lives easier, we wrote a little console application to enable us to quickly add data to JSON files.



```
Run AddRaceToJson
Enter Seconds:
25
Number of turns? :
15
Length of the circuit?:
3.333
Country circuit?:
Monaco
Number of laps?
75
What is the name of the race?
Monaco Grand Prix
You just added a new car to the list

What do you want to do?
1. Add a new Race to Json Database...
2. Show Json Race Database
3. Close the application...
1
Enter Hours:
1
Enter Minutes:
25
Enter Seconds:
25
Number of turns? :
1
```

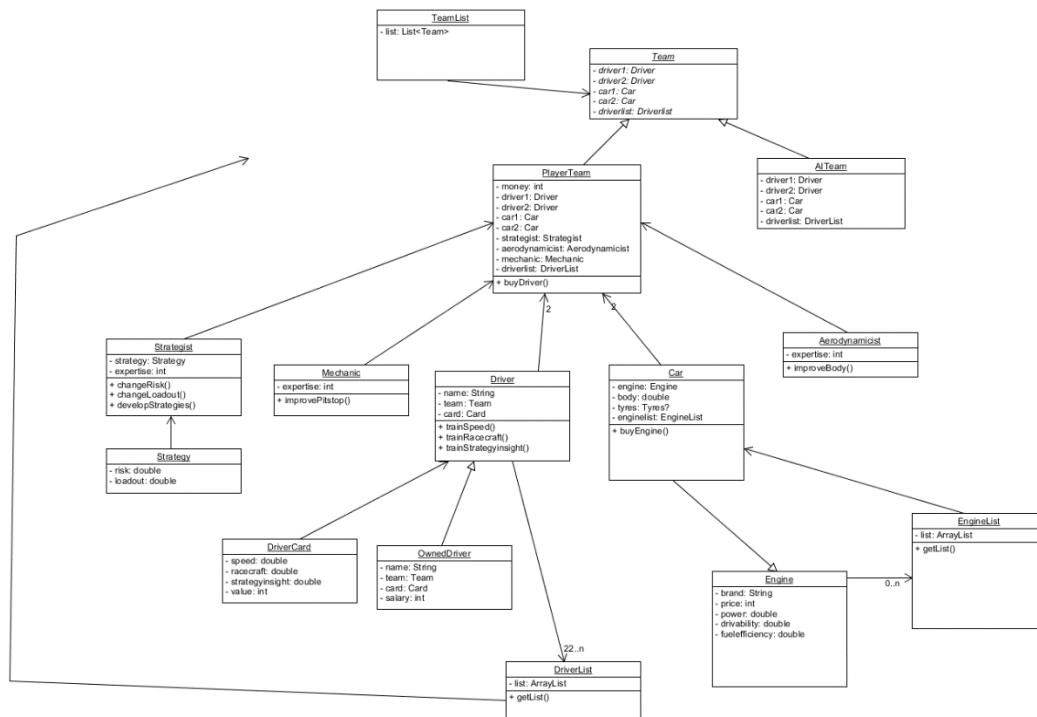
SVG Paths: We added SVG Paths to give the player a visual representation of the upcoming race. We didn't find enough (good) paths for all races on the internet though, so one thing we learned from this is that having consistent resources is important when developing a game.

Splash Loading Screen: We added a loading screen just for the sake of improving the over-all user experience.

UI & Design: We kept improving the design of the User Interface until we were satisfied with it. We have added our own buttons for minimizing and closing the application. We removed the default window border while keeping the functionality to change the location of the application.

Load/Save game: We added the functionality to give the player the ability to load a game from an earlier save, or save the current game.

UML: We used UML to map out the structure of our program. This was good at the start of the project because it helped us visualize the project in our heads. However, along the way we encountered a lot of things that had to be changed and added. We had to make certain classes static and add a game class. This also meant that, close to the end of the project, we lost the overview a little. It often was not very clear where certain methods were and how to access them. But eventually we always managed to figure it out.



Points for improvement

Since we used checkstyle to make sure that the code looks clean and is easy to understand when read, the written code can hardly be improved. Checkstyle made sure that everyone was on the same page concerning notations and indentations.

Parts of the code that could be improved on would probably be the re-use of code. A few times it occurs that code is duplicated, which isn't necessary of course since there could be written a method for this purpose. Changing this wouldn't have a huge impact, since the size of the project is rather small, and creating one of these methods would only result in a few lines less of code.

Another point that could be improved on would be the tests. During the project we've noticed how important tests are. Methods that looked like they were working, were actually doing something completely wrong, signs in an equation would be reversed (e.g. '>' instead of '<'). Tests made these mistakes very clear. There are however methods for which tests are a lot harder to write, and it is questionable if the time spent writing the tests is actually worth it. Testing methods for an object which has many relations to other objects is a cumbersome task. Certain things make this easier, such as Mockito, which mocks an object. Mockito, however, isn't perfect either. Since it mocks an object, the mock isn't a perfect representation of what the object actually is. Certain attributes or functions can be lost in this process. The time it takes to write these tests could also be used to manually go over the method and try to find any errors by hand. Of course this way of testing is more subject to errors. In conclusion, we think that the more tests the better, but the time it takes to write them should be weighed against the benefits.

In the first few weeks, we decided to get together an extra day in addition to the regular schedule. We rented a room in the EWI and discussed what problems we had and how to overcome them. This was very useful in the earlier weeks, since there was much planning to do. In the later weeks of the project these meetings were less frequent. With the progress that was made during these sessions, an improvement would be to keep these sessions frequently, even in the later stages of the project.

At the last meeting with our TA, he informed us that this project had sprints of one week before, instead of the sprints of two weeks. Having two weeks makes goals for sprints a lot larger. Since these goals for two weeks are somewhat large, and set in advance, the situation happened where during the sprint we found out new things that made either the set goals obsolete or inaccurately chosen since a goal was more time consuming than anticipated. The two-weekly sprints also made the structure of the meetings on Mondays a little odd. One week there was little to be discussed, whereas for other weeks an agenda had to be made for the meeting and someone had to write down what was talked about. This inconsistency caused some trouble. We think that weekly sprints give an overview that's more clear, and a better structure.

Individual feedback

Elias: “I think that I have worked hard on this project. I was trying to not just write code, but also test it and looking for ways to improve it and make it more efficient. I worked on both the backend and the UI, but I think that my preference goes to writing code in the backend. We worked with JavaFX and even though it was interesting to learn about, I’d rather write code with the logic behind it instead of connecting the logic to the UI. The goal of the project wasn’t just learning how to program, though, but also learning the art of Software Engineering. We worked in a small group of 4 people, and I think that’s a good size for a Software Development team. I enjoyed working with my teammates on the project. Especially the meet-ups and brainstorming part was a good experience. Not everyone did as much, though, which I found unfortunate. My experience is that delivering software that is useful to people and consistent isn’t as easy as having some meetups and writing code. Planning, managing, testing and meeting-up are all important aspects in Software Development. This was my first experience with this workflow, and I have experienced how this workflow makes developing software actually more efficient.”

David: “In general I believe it went pretty well. I did committed third most lines in the team (of four). With quite a bit of distance between me and the second and first. However, I do not believe I didn’t do enough work. There are definitely members in the team that did more work than me, but I was always (like the others) involved and up to date with the decisions and changes made. And a lot of my tasks were not very line-heavy but more research-heavy (like UI).

After a few weeks, there were a few weeks in which I should have done more though. The bulk of my work was done in the first few, and mostly the last two weeks of the project. Apart from several other tasks, I mostly focused on the UI in the project. This includes anything that has to do with user interaction with the program. Like Populating lists and managing button clicks. Elias is the other team member that worked on the ui a lot. Of course we occasionally had some conflicts about the way things should be handled best, but we always managed to figure the best solution out together. We once spent hours on skype (while I looked at his screen via skype) trying to figure out the best way to change our single (ui)controller class to multiple controllers that could communicate with each other. Overall I think the project worked pretty well and I am happy with the result.”

Tim: “This project has been an adventure to me. Learning new coding techniques, getting familiar with new software and finding a great way to work as a team are just some examples of what this project has given me. I want to point out that this truly is a brilliant way to learn new things!

Getting more into detail of what I learned, I think version control, Github, is an essential skill to any computer engineer and I feel like this project has really given me the opportunity to familiarize myself with it. It is frustrating in the beginning but after a few weeks you start to get the hang of it and ask yourself “how could I ever live without it?”. Also, I made the decision to step away from Eclipse (after the TA’s recommendation) to IntelliJ and I don’t see myself coming back anytime soon since it is a fantastic program! I am now able to work with more tools thanks to this project and I haven’t even talked about Maven, JavaFX, and all the other tools we used to create our game!

Working as a team was very pleasant for me because I was surrounded by people who had about the same coding skills and this meant everybody could keep up and when need help each other. Overall, I must say that this project was both fun and interesting and I think a lot of what I learned during this semester will come in handy during the rest of my bachelor.”

Wouter: “At the start of the project, I was full of ideas. Ranging from features to be implemented to the structure that would be necessary. Quickly I realized that this was a lot more work than I thought. Writing a Formula-1 Manager isn’t the same as writing a House Catalog that was required for the test in the first quarter. We went ahead and tried to write some code to make progress, only to find that we needed more and more classes to link everything together. I’d preferably work on a project with a good preparation, having everything sorted out before starting to code gives a good overview of what fits where. Unfortunately this was not the way this project was going to be, since a more gradual approach was suggested. This took some time getting used to.

When writing code, it would sometimes occur that someone else was also working on this. This was miscommunicated, which caused some frustration. Eventually we worked things out by using issues on GitHub. This made it clear for everyone what had to be done, since the tasks could be assigned.

Another issue I had was that the code others had written was not clearly described. Especially at the start of the project we used poorly written commit messages, and our pull requests would be extremely large so that going over all of the written code and checking everything in detail was a lot of trouble. Eventually the pull requests became smaller, and the added code was better defined.

Overall I think the main thing the project has taught me is how working in a group is done and how some of the most important tools work, such as git.”

Scrumboard week 4, 7, 8 and 9

The image displays two screenshots of a Jira Scrumboard for the project 'F1-Manager'. The top screenshot shows the board for SPRINT 1, with columns for 'Ready to start', 'Backlog', 'In Progress', 'Review/QA', 'Done', and 'Closed'. Tasks are organized into sprints, with SPRINT 1 containing tasks like 'JSON databases', 'Create Formula for winning', 'Foundation', 'Mechanic', 'BuyDrivers & AiBuyDrivers', 'Strategist improve function', 'Class PlayerTeam has too many parameters, needs to be fixed', 'Team Class (Player/AI)', 'Create Driver Class', 'Java FX + GUI', and 'Create Engine Class'. The bottom screenshot shows the board for SPRINT 3, with columns for 'Backlog', 'Ready to start', 'In Progress', 'Review/QA', 'Done', and 'Closed'. Tasks include 'Finish team Equals tests', 'Upgradeable Classes', 'Tests for List classes', 'Final Report', 'Start race', 'Compose Team', 'Upgrades doen', 'Artificial Intelligence', 'Screenshot of test coverage', 'choosing drivers and their cars', 'Initial version UI tabs', 'Driver Upgradeable', 'Upgradeable Classes', 'Engine Price Test', 'Driver Value Test', 'Implement formula', 'Drivers overkopen', 'Minimale crew hebben', 'Removed duplicate', 'Game logics', 'Artificial Intelligence', and 'Mechanic now impleme upgradeable'. The interface includes navigation tabs for Code, Issues, Pull requests, Boards, Reports, Projects, and Wiki, along with filters for Unwatch, Unstar, and Fork.

This repository Search Pull requests Issues Gist ToDo

elarb / F1-Manager Private Unwatch 2 Unstar 2 Fork 0

<> Code Issues 20 Pull requests 0 Boards Reports Projects 0 Wiki

Repos (1/1) Show one Labels Milestones Assignees Epics Search (/) New Issue +

2 Backlog 1 Ready to start 6 In Progress 5 Review/QA 3 Done 7 Closed 55

F1-Manager #86 Tests for List classes SPRINT 3

F1-Manager #66 Start race SPRINT 3

F1-Manager #72 Finish team Equals tests FINAL SPRINT

F1-Manager #109 (de)serialisation of enums not working properly FINAL SPRINT

F1-Manager #107 Buy Driver Method misbehaving FINAL SPRINT

F1-Manager #56 Upgrades doen Artificial Intelligence

F1-Manager #59 choosing drivers and their cars Compose Team

F1-Manager #100 connect racer button and put raceresults into de raceresults list FINAL SPRINT

F1-Manager #103 display game events on hometab FINAL SPRINT

F1-Manager #87 Final Report FINAL SPRINT

F1-Manager #108 Driver images FINAL SPRINT

F1-Manager #80 Driver Upgradeable SPRINT 3 Upgradeable Classes

F1-Manager #84 Upgradeable Classes SPRINT 3

F1-Manager #53 Implement formula SPRINT 3

F1-Manager #54 Drivers overkopen Artificial Intelligence

F1-Manager #57 Minimale crew hebben Artificial Intelligence

F1-Manager #51 Startscherm SPRINT 3

F1-Manager #76 Screenshot of test coverage FINAL SPRINT

F1-Manager #105 Update readme

F1-Manager #77 add newgame and loadgame

F1-Manager #102 added report

F1-Manager #106 removed /lib

F1-Manager #104 readme-updates

F1-Manager #101 Race logics

F1-Manager #99 Save and loadgame rein

F1-Manager #98 reimplement load and s into new ui FINAL SPRINT

F1-Manager #97 the tires slider now sets of the tires

F1-Manager #96 set tires with slider in ui

elarb / F1-Manager Private Unwatch 2 Unstar 2 Fork 0

<> Code Issues 13 Pull requests 1 Boards Reports Projects 0 Wiki

Repos (1/1) Show one Labels Milestones Assignees Epics Search (/) New Issue +

4 Backlog 4 Ready to start 5 In Progress 4 Review/QA 3 Closed 48 Add a Pipeline ...

F1-Manager #86 Tests for List classes SPRINT 3

F1-Manager #122 Bad SVG Paths enhancement

F1-Manager #134 Game saves and load shouldn't be from resources folder

F1-Manager #142 fix/fplistview-crewtab

F1-Manager #132 Engine List @Marketplace FINAL SPRINT

F1-Manager #133 Engine Buy Methods FINAL SPRINT

F1-Manager #138 Upgrade costs to low FINAL SPRINT

F1-Manager #135 Formula Rebalance FINAL SPRINT

F1-Manager #137 Per race bonus & costs FINAL SPRINT

F1-Manager #103 display game events on hometab FINAL SPRINT

F1-Manager #107 Buy Driver Method misbehaving FINAL SPRINT

F1-Manager #87 Final Report FINAL SPRINT

F1-Manager #136 Feature/ Popup when crew-member is max upgraded already FINAL SPRINT

F1-Manager #141 JFXListView not registering Mouse Events FINAL SPRINT

F1-Manager #140 Added files that were missing to the resources

F1-Manager #139 Font missing in resources FINAL SPRINT

F1-Manager #131 home-tab-tables + #126, #128, #129

F1-Manager #125 Undecorated window & App styling FINAL SPRINT

F1-Manager #127 Slider misplaced FINAL SPRINT

F1-Manager #124

Test coverage (each week)

Packages

[All](#)
[edu.tudelft.games.f1manager](#)

Coverage Report - All Packages

Package /	# Classes	Line Coverage		Branch Coverage		Complexity
All Packages	3	0%	0/12	N/A	N/A	1
edu.tudelft.games.f1manager	3	0%	0/12	N/A	N/A	1

Report generated by [Cobertura](#) 2.1.1 on 1/27/17 11:44 AM.

edu.tudelft.games.f1manager

Classes

[Constants](#) (0%)
[Controller](#) (0%)
[Main](#) (0%)

Packages

[All](#)
[edu.tudelft.games.f1manager](#)
[edu.tudelft.games.f1manager.core](#)

Coverage Report - All Packages

Package /	# Classes	Line Coverage		Branch Coverage		Complexity
All Packages	18	0%	0/155	0%	0/28	1.145
edu.tudelft.games.f1manager	2	0%	0/13	N/A	N/A	1
edu.tudelft.games.f1manager.core	16	0%	0/142	0%	0/28	1.152

Report generated by [Cobertura](#) 2.1.1 on 1/27/17 12:06 PM.

All Packages

Classes

[Aerodynamicist](#) (0%)
[AlTeam](#) (0%)
[AlTeamList](#) (0%)
[Car](#) (0%)
[CarList](#) (0%)
[Card](#) (0%)
[Constants](#) (0%)
[Controller](#) (0%)
[Driver](#) (0%)
[DriverList](#) (0%)
[Engine](#) (0%)
[Main](#) (0%)
[Mechanic](#) (0%)
[PlayerTeam](#) (0%)
[Stratelist](#) (0%)
[Team](#) (0%)
[Tyres](#) (0%)
[Upgradeable](#) (N/A)

Packages

All
[edu.tudelft.games.f1manager](#)
[edu.tudelft.games.f1manager.core](#)
[edu.tudelft.games.f1manager.game](#)
[edu.tudelft.games.f1manager.tools](#)

Coverage Report - edu.tudelft.games.f1manager

Package /	# Classes	Line Coverage	Branch Coverage	Complexity
edu.tudelft.games.f1manager	2	0% 0/13	N/A N/A	1
edu.tudelft.games.f1manager.core	19	33% 60/238	46% 9/62	1.354
edu.tudelft.games.f1manager.game	5	0% 0/64	0% 0/4	1.053
edu.tudelft.games.f1manager.tools	1	0% 0/49	0% 0/6	3
Classes in this Package /	Line Coverage	Branch Coverage	Complexity	
ClientController	0% 0/2	N/A N/A	1	
Main	0% 0/11	N/A N/A	1	

Report generated by [Cobertura](#) 2.1.1 on 1/27/17 12:09 PM.

All Packages

Classes

[AddCarToJeton](#) (0%)
[Aerodynamicist](#) (0%)
[AlTeam](#) (75%)
[AlTeamList](#) (0%)
[Car](#) (0%)
[CarList](#) (0%)
[Card](#) (0%)
[Circuit](#) (0%)
[ClientController](#) (0%)
[Constants](#) (0%)
[Driver](#) (50%)
[DriverList](#) (0%)
[DriverResult](#) (0%)
[Engine](#) (0%)
[EngineList](#) (0%)
[Main](#) (0%)
[Mechanic](#) (0%)
[PlayerTeam](#) (83%)
[Race](#) (0%)
[RacesList](#) (0%)
[Season](#) (0%)
[Strategist](#) (86%)
[Team](#) (36%)
[Tyres](#) (0%)
[Upgradable](#) (N/A)

Packages

All
[edu.tudelft.games.f1manager](#)
[edu.tudelft.games.f1manager.core](#)
[edu.tudelft.games.f1manager.game](#)
[edu.tudelft.games.f1manager.tools](#)

Coverage Report - edu.tudelft.games.f1manager

Package /	# Classes	Line Coverage	Branch Coverage	Complexity
edu.tudelft.games.f1manager	2	0% 0/13	N/A N/A	1
edu.tudelft.games.f1manager.core	19	33% 60/238	46% 9/62	1.354
edu.tudelft.games.f1manager.game	5	0% 0/64	0% 0/4	1.053
edu.tudelft.games.f1manager.tools	1	0% 0/49	0% 0/6	3
Classes in this Package /		Line Coverage	Branch Coverage	Complexity
ClientController		0% 0/2	N/A N/A	1
Main		0% 0/11	N/A N/A	1

Report generated by [Cobertura](#) 2.1.1 on 1/27/17 12:09 PM.

All Packages

Classes

[AddCarToJeton](#) (0%)
[Aerodynamicist](#) (0%)
[AlTeam](#) (75%)
[AlTeamList](#) (0%)
[Car](#) (0%)
[CarList](#) (0%)
[Card](#) (0%)
[Circuit](#) (0%)
[ClientController](#) (0%)
[Constants](#) (0%)
[Driver](#) (50%)
[DriverList](#) (0%)
[DriverResult](#) (0%)
[Engine](#) (0%)
[EngineList](#) (0%)
[Main](#) (0%)
[Mechanic](#) (0%)
[PlayerTeam](#) (83%)
[Race](#) (0%)
[RacesList](#) (0%)
[Season](#) (0%)
[Strategist](#) (86%)
[Team](#) (36%)
[Tyres](#) (0%)
[Upgradable](#) (N/A)

Coverage Report - All Packages

Package /	# Classes	Line Coverage	Branch Coverage	Complexity
All Packages	30	23% 109/460	55% 43/78	1.317
edu.tudelft.games.f1manager	5	0% 0/51	N/A N/A	1
edu.tudelft.games.f1manager.core	18	49% 166/222	69% 43/62	1.38
edu.tudelft.games.f1manager.game	5	0% 0/64	0% 0/4	1.053
edu.tudelft.games.f1manager.tools	2	0% 0/103	0% 0/12	3

Report generated by [Cobertura](#) 2.1.1 on 12/19/16 3:11 PM.

Packages

[All](#)
[edu.tudelft.games.fimanager](#)
[edu.tudelft.games.fimanager.core](#)
[edu.tudelft.games.fimanager.game](#)
[edu.tudelft.games.fimanager.tools](#)

All Packages

Classes

[AddCarToJaxon](#) (0%)
[Aerodynamicist](#) (100%)
[AiTeam](#) (87%)
[AiTeamList](#) (100%)
[App](#) (0%)
[AppController](#) (0%)
[Car](#) (0%)
[CarList](#) (0%)
[Circuit](#) (0%)
[ClientController](#) (0%)
[Constants](#) (0%)
[Driver](#) (59%)
[DriverList](#) (91%)
[DriverResult](#) (0%)
[Engine](#) (40%)
[EngineList](#) (0%)
[Main](#) (0%)
[Mechanic](#) (100%)
[PlayerTeam](#) (90%)
[Race](#) (0%)
[RacesList](#) (0%)
[RandomDouble](#) (0%)
[Season](#) (84%)
[Strategist](#) (95%)
[Team](#) (60%)
[TeamInstanceCreator](#) (25%)
[Tyres](#) (75%)
[Upgradable](#) (N/A)

Packages

[All](#)
[edu.tudelft.games.fimanager](#)
[edu.tudelft.games.fimanager.core](#)
[edu.tudelft.games.fimanager.game](#)
[edu.tudelft.games.fimanager.tools](#)

All Packages

Classes

[Aerodynamicist](#) (95%)
[AiTeam](#) (100%)
[AiTeamList](#) (74%)
[Car](#) (57%)
[Circuit](#) (4%)
[ClientController](#) (0%)
[ConfigurationTabController](#) (0%)
[Constants](#) (0%)
[CrewTabController](#) (0%)
[Driver](#) (70%)
[DriverList](#) (91%)
[DriverResult](#) (45%)
[Engine](#) (40%)
[EngineList](#) (0%)
[FadeApp](#) (0%)
[Game](#) (47%)
[GameEvent](#) (47%)
[GameEvents](#) (80%)
[HomeTabController](#) (0%)
[Main](#) (0%)
[MainController](#) (0%)
[Mechanic](#) (94%)
[NextRaceTabController](#) (0%)
[PlayerTeam](#) (70%)
[Race](#) (9%)
[RandomDouble](#) (75%)
[Season](#) (61%)
[SettingsTabController](#) (0%)

Coverage Report - All Packages

Package /	# Classes	Line Coverage	Branch Coverage	Complexity
All Packages	32	48% 31/673	70% 56/80	1.289
edu.tudelft.games.fimanager	4	0% 0/28	N/A N/A	1
edu.tudelft.games.fimanager.core	21	72% 109/267	80% 56/70	1.37
edu.tudelft.games.fimanager.game	5	22% 22/100	0% 0/4	1.05
edu.tudelft.games.fimanager.tools	2	0% 0/8	0% 0/6	2.333

Report generated by [Cobertura](#) 2.1.1 on 1/27/17 12:24 PM.

Coverage Report - All Packages

Package /	# Classes	Line Coverage	Branch Coverage	Complexity
All Packages	37	49% 36/735	60% 52/105	1.262
edu.tudelft.games.fimanager	11	0% 0/129	0% 0/4	1
edu.tudelft.games.fimanager.core	17	70% 249/344	79% 63/76	1.414
edu.tudelft.games.fimanager.game	8	45% 11/250	0% 0/20	1.036
edu.tudelft.games.fimanager.tools	1	75% 3/4	N/A N/A	1

Report generated by [Cobertura](#) 2.1.1 on 1/27/17 12:41 PM.

Packages

[All](#)
[edu.tudelft.games.fimanager](#)
[edu.tudelft.games.fimanager.core](#)
[edu.tudelft.games.fimanager.game](#)
[edu.tudelft.games.fimanager.tools](#)

Coverage Report - All Packages

Package /	# Classes	Line Coverage	Branch Coverage	Complexity
All Packages	43	33% 328/968	37% 15/172	1.272
edu.tudelft.games.fimanager	18	0% 0/490	0% 0/68	1.375
edu.tudelft.games.fimanager.core	17	72% 380/545	82% 64/76	1.375
edu.tudelft.games.fimanager.game	7	50% 75/149	16% 1/6	1.066
edu.tudelft.games.fimanager.tools	1	75% 3/4	N/A	1

Report generated by [Cobertura](#) 2.1.1 on 1/27/17 12:43 PM.

All Packages

Classes

[Aerodynamicist](#) (95%)
[AiTeam](#) (100%)
[AiTeamList](#) (70%)
[Aur](#) (0%)
[Car](#) (57%)
[Circuit](#) (4%)
[ClientController](#) (0%)
[Constants](#) (0%)
[CrewTabController](#) (0%)
[Driver](#) (75%)
[DriverList](#) (90%)
[DriverResult](#) (72%)
[Engine](#) (42%)
[EngineList](#) (0%)
[GameEvent](#) (62%)
[GameEvents](#) (78%)
[HomeTabController](#) (0%)
[MainController](#) (0%)
[MarketPlaceTabController](#) (0%)
[Mechanic](#) (94%)
[NextRaceTabController](#) (0%)
[PlayerTeam](#) (51%)
[Race](#) (9%)
[RandomDouble](#) (75%)
[Season](#) (64%)
[SettingTabController](#) (0%)
[Strategist](#) (94%)
[TableDriver](#) (0%)