Workshop overview

**From:** Mike Talks [Avionic Manager]   
**Sent:** Thursday, 3 August 2017 12:51 a.m.  
**To:** Avionics Team  
**Subject:** Panther ODM module review

Hey Team

As you’re aware, for the last couple of months, one of our programmers - Jake D’Arby - has been working on the code for the Operational Data Manual ODM for the Panther aircraft.

The ODM is a key part of our system for mission planning on the aircraft. Simply put, it allows us to plan out a route, and work out if you have enough fuel. If the calculations don’t work properly, it means potentially having to ditch a $40 million aircraft, which isn’t a great look for us if we’re seen as responsible.

Jake said the module was working, and had finished doing unit tests on it. Unfortunately after winning the lottery rollover on Saturday, Jake decided to quit the company in spectacular fashion.

You’ll probably remember he did come in briefly last Monday to pick up some of his Star Wars collectables. And in fact I’d appreciate the employee responsible to take the video of him trying to defecate on our CEO’s car off of YouTube.

This component is currently already overdue – I need you to have a look into it, see if you can do some further testing, and get back to me on it. If it needs some additional work, I need some kind of plan about what that work is. This has really rattled the board who agree Jake should not have been working on such an important module so unsupervised, and now this area has a high degree of visibility.

I managed to talk Jake into providing some support (sadly under threat of legal action from us). The email below regarding what he’s built so far. I’m providing it to you unedited. Please do not share it.

Given the situation with Jake, we’d prefer to avoid any further interaction with him.

Thanks team, you have my full support to get this sorted,

Mike Talks

Avionics Manager, Panther Project

**From:** jakey\_boy@codewizz.com   
**Sent:** Wednesday, 2 August 2017 19:12 p.m.  
**To:** Mike Talks [Avionic Manager]  
**Subject:** Panther code explanation

Hey wage slave!

So today I picked up my Ferrari, and got myself a Rolls on back order. How’s the piece of crap you’re driving these days?



As promised, here’s a breakdown of what I built for the morons at the office. Everything’s under the **AircraftManagement.java** class – I’m only going to tell you about the public methods, everything is works and it has all the tests from the manual. DO NOT TOUCH THE PRIVATE METHODS – it works fine.

| **Method** | **Description** |
| --- | --- |
| getAllUpWeight() | Gives you the total weight of the aircraft, including equipment and fuel. |
| getRemainingFuel() | Returns how much fuels is left on the aircraft. |
| climb() | Puts the aircraft into a climb, removing fuel for the manoeuvre, and sets a new altitude. |
| dive() | Puts the aircraft into a climb and sets a new altitude. |
| aircraftLeg() | This works out the fuel consumption for a leg of a route.  You pass in the aircraft speed and the distance you’re travelling. It uses the altitude and current aircraft weight.  Then it subtracts the fuel used from the remaining fuel. |
| getFuelUsedLastManoeuvre() | Yeah, just added this to make easier to test. Just returns how much fuel the last manoeuvre (whether climb, dive or leg) used. Pretty sweet huh? |
| addA2AMissile()  addA2AMissile(num) | These add the weight of an A2A missile from the aircraft. You can actually provide a number, and it will add/remove that number of missiles to the aircraft, like,  addA2AMissile(4) |
| addDumbBomb()  addDumbBomb(num)  dropDumbBomb()  dropDumbBomb(num) | These add/remove the weight of a dumb bomb from the aircraft. As with the A2A, if you include an integer, it’ll add/remove that number of bombs. |
| addIntelliBomb()  dropIntelliBomb() | Add/remove the weight of an IntelliBomb to the aircraft. |
| addExternalFuelTanks() | Add the weight of an external fuel tank, increase the maximum fuel allowance. |
| addFuel() | This refuels the aircraft if it’s allowed. Adds the weight of fuel you provide to the aircraft, as long as it wouldn’t exceed your maximum fuel. |

All the classes are working nicely, if you REALLY feel you have to add more tests, then you only really need to look into the either the **AicraftManagementTest.java** or the **FuelRateDataTests.java** test classes – that’s where all the JUnit tests are. Just add some more in there if you have to – you can use the methods above to define what you need – I assume your monkeys know how to use **AssertTrue**, **AssertFalse** and **AssertEquals**? The class **TestSuite.java** runs both those sets of tests as a suite.

That should be all your cronies need to know – who knows I might even buy myself a Panther just for giggles!

If you ever need a job, I’ll always need someone to wash my car. Actually just kidding, I’m putting you in my block list now, so if you need anything else just talk to my lawyer.

Jake