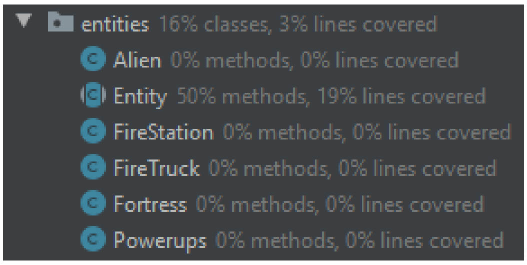
Unit Testing Analysis

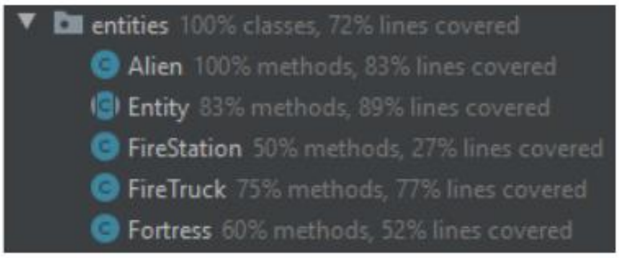
This report is a breakdown of the Junit testing, looking at the line coverage, method coverage and time efficiency.

Unfortunately, due our changes to enable the use of powerups in the game, we broke some of the unit tests written by the previous group. This was due to the use of textures and sprite batches not working with the headless backend that we use to run tests with libgdx. We added black box user testing to supplement the reduced number of unit tests, however these will not be included in the line/method coverage.

Line and Method Percentage Coverage



```
▼ entities 16% classes, 3% lines covered
    Alien 0% methods, 0% lines covered
    Entity 50% methods, 19% lines covered
    FireStation 0% methods, 0% lines covered
    FireTruck 0% methods, 0% lines covered
    Fortress 0% methods, 0% lines covered
    Powerups 0% methods, 0% lines covered
```

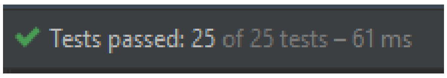


```
▼ entities 100% classes, 72% lines covered
    Alien 100% methods, 83% lines covered
    Entity 83% methods, 89% lines covered
    FireStation 50% methods, 27% lines covered
    FireTruck 75% methods, 77% lines covered
    Fortress 60% methods, 52% lines covered
```

Comparing our tests (left) to the previous groups tests (right) you can see a dramatically reduced number of unit tests. The class ‘Entity’ retains several very important tests, and the methods/lines not tested were not deemed worth unit testing.

However we do cover some very important areas with our unit tests, like Entity and GameObject.

Time Efficiency



```
✓ Tests passed: 25 of 25 tests – 61 ms
```

By looking at this, we can see that the time it takes to run our tests is 61ms, with 25 tests this comes down to 2.4ms, a significant improvement to the previous groups test times. However, this may be since tests that use classes that inherit from entity take more time than the tests that we were able to complete.

Summary

We feel that our tests, while less than the previous group, still cover many of the most important areas of the code base. The areas that we were not able to cover with unit tests, we thoroughly tested with black box user testing. See our evidence document for the user tests for this.