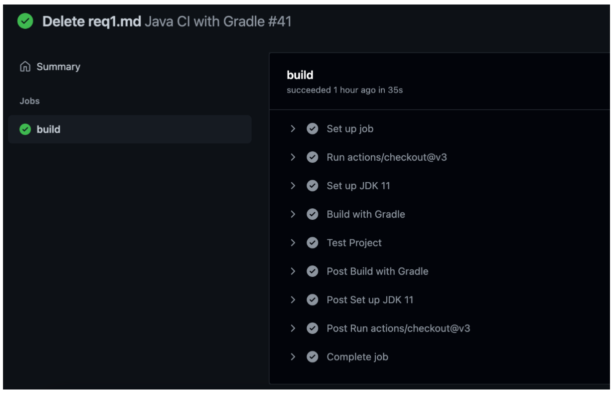
Testing

One of our two main testing methods is unit testing. We created tests that would run when our code is updated or new code is written. We wanted to make sure that our tests reflected the specifications of the project and therefore meet the needs of the client. This process is automated through continuous integration. We created our unit tests using JUnit and libGDX built-in packages for testing. We have a main ‘GdxTestRunner.java’ file which enables us to create our tests in separate folders and files. We aimed for a more reactive method of testing, meaning that we knew what we needed to test, as it had already been written, and because we have an agile method of working we need to be able to quickly react to changes of requirements or methods of working, so creating the tests once the code has been developed means that we are working more efficiently as we only need to write the tests that are needed.

One of our main testing methods was unit tests. We created a ‘Java CI with Gradle’ file on GitHub Actions meaning that every time code was updated the tests would run. This is the window that shows these tests and whether they fail or not:



✓ **Delete req1.md** Java CI with Gradle #41

Summary

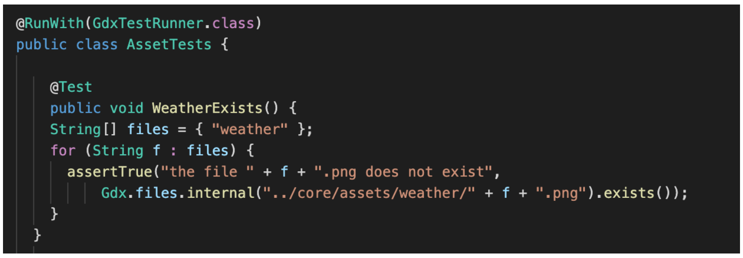
Jobs

✓ build

build
succeeded 1 hour ago in 35s

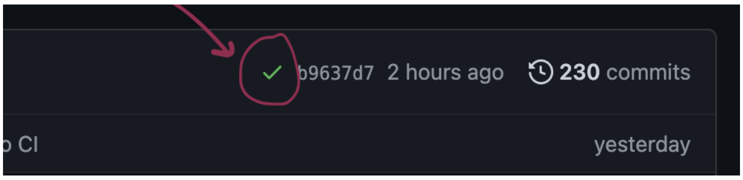
- ✓ Set up job
- ✓ Run actions/checkout@v3
- ✓ Set up JDK 11
- ✓ Build with Gradle
- ✓ Test Project
- ✓ Post Build with Gradle
- ✓ Post Set up JDK 11
- ✓ Post Run actions/checkout@v3
- ✓ Complete job

There are several different sections of the code we tested in, here are a few examples of our unit tests:



```
@RunWith(GdxTestRunner.class)
public class AssetTests {

    @Test
    public void WeatherExists() {
    String[] files = { "weather" };
    for (String f : files) {
      assertTrue("the file " + f + ".png does not exist",
          Gdx.files.internal("../core/assets/weather/" + f + ".png").exists());
    }
  }
```



We created ‘asset’ tests which are verification tests meaning that they reflect whether the system can build properly. Most of the ‘asset’ tests verify whether the right elements are present, as if there were parts that made up the code, or the design of the game, that were missing then the game wouldn’t function as it were designed or written to. These tests can’t guarantee that there aren't going to be any issues, but they reduce the risk of things going wrong, and can alert the software engineers when something may be missing or changed. When designing the tests we evaluated what parts of the code we wanted to test, why we wanted to test it (in order to make sure they are accurate) and where the tests were going to be run. As we have CI set up, it was easy for us to run and test the tests, as every time there is an update to the code, all of the tests run.

The other of our main testing methods is manual testing. We decided to have a greater focus on manual testing as it allows us to spot every little detail within the game. Automated tests can be faster however, having manual testing allows us to look into the HCI of the project, making sure that it's usable and easy to understand to users and clients. Although automated testing is faster, it takes longer to set up, so with manual testing we could implement it as we developed the game along the way. Each time we implemented a new feature we would conduct a series of manual tests to see whether the new implementation is usable and implemented in the style that the client has requested. It allowed us to test right away, without having to wait for tests to be written. When it came to testing we made sure to understand the requirements and prepare the tests that we were going to conduct. Within the plan, we created scenarios e.g. clicking the start button, the game should start, or clicking the wrong button, nothing should happen. We conducted the tests and as we found bugs, added them to our Jira board so that the team could see the outcome of the test and understand what needs to be fixed.

The full spreadsheet can be viewed here: https://docs.google.com/spreadsheets/d/12wfwlKJlaAXOHuJUjN6RP-YaaaFsOxykkVfS2w2Vgx4/edit?usp=sharing

Class	Test Description	Input	Expected Ouput	Actual Output	Pass/Fail	Reason for fail	Link	Author
Ship	Does the ship move upwards	On Keyboard W	The y value will increase	The y value increased	Pass		https://drive.google.com/file/d/1jOFR5PPZQK7K6Wkh_DTVNroH4C2qt0B1/view?usp=sharing	Josh Whitehurst
	Does the ship move downwards	On Keyboard S	The y value will decrease	The y value decreased	Pass		https://drive.google.com/file/d/1XuWvOanEDqO_TctEFx9rq-y4vMe6VVIg/view?usp=sharing	Josh Whitehurst
	Does the ship move to the left	On Keyboard A	The x value will decrease	The x value decreased	Pass			Josh Whitehurst
	Does the ship move to the right	On Keyboard D	The x value will increase	The x value increased	Pass		https://drive.google.com/file/d/1DEiH1-q8YgPy-UIttA8q_hMgziuaJP8Z/view?usp=sharing	Josh Whitehurst
	Does the ship take damage from cannonballs		The health of the player ship will decrease	The health of the ship decreased	Pass		https://drive.google.com/file/d/1aaQvA-6__93MmhLbF2pwnc3EGdqA7B56/view?usp=sharing	Josh Whitehurst
	Does the ship spawn with the correct stats		The ship will spawn with the correct health experience and gold with health depending on difficulty	The ship spawned with the correct health, experience and gold with the health being correct with the difficulty selected	Pass		https://drive.google.com/file/d/1KAz8eqHsXq4qBq0ugKFw3lmdGEZNv6tf/view?usp=sharing	Josh Whitehurst
	Does the ship take damage from weather		The health of the ship will decrease	The health of the ship decreased	Pass		https://drive.google.com/file/d/1sAZGVHvCLbQWzK1YyJVX7f3B8l6Lx1uP/view?usp=sharing	Josh Whitehurst
	Do Constantine ship image files	Unit Test that tests the validity	The test will pass	The test passed	Pass		Unit Test - tests/src/de/boatcorporation	Charlotte Gayton

	exist *ConstantineShipExists()*	of the Constantine ship image files					/gdxtesting/general/AssetTests.java	
	Do Langwith ship image files exist *LangwithShipExists()*	Unit Test that tests the validity of the Langwith ship image files	The test will pass	The test passed	Pass		Unit Test - tests/src/de/boatcorporation/gdxtesting/general/AssetTests.java	Charlotte Gayton
	Do Derwent ship image files exist *DerwentShipExists()*	Unit Test that tests the validity of the Derwent ship image files	The test will pass	The test passed	Pass		Unit Test - tests/src/de/boatcorporation/gdxtesting/general/AssetTests.java	Charlotte Gayton
	Do Halifax ship image files exist *HalifaxShipExists()*	Unit Test that tests the validity of the Halifax ship image files	The test will pass	The test passed	Pass		Unit Test - tests/src/de/boatcorporation/gdxtesting/general/AssetTests.java	Charlotte Gayton
	Do Vanbrugh ship image files exist *VanbrughShipExists()*	Unit Test that tests the validity of the Vanbrugh ship image files	The test will pass	The test passed	Pass		Unit Test - tests/src/de/boatcorporation/gdxtesting/general/AssetTests.java	Charlotte Gayton
College	Does the college take damage		The health of the college being attacked will decrease	The health of the college decreased	Pass		https://drive.google.com/file/d/1Tic0ZRyGnlhucrvOt_dkAms0iFLw-AHc/view?usp=sharing	Josh Whitehurst
	Does the college shoot the player ship		The college will shoot at the player if he is in range	The college shot at the players ship	Pass		https://drive.google.com/file/d/1CXZLA2pTqkMQTuU7U9sNi4K1rgsiLs1Z/view?usp=sharing	Josh Whitehurst
	Does the college show a white flag when conquered		When the college health goes to 0 a white flag will appear to show it has been conquered	The college showed a white flag when its health got to 0 as it had been conquered	Pass		https://drive.google.com/file/d/1WEN1S92jC6xJLhw4FbvsrNbEkU5x_did/view?usp=sharing	Josh Whitehurst
	Do college image files exist *CollegeExists()*	Unit Test that tests the validity of the college image files	The test will pass	The test passed	Pass		Unit Test - tests/src/de/boatcorporation/gdxtesting/general/AssetTests.java	Charlotte Gayton

	Do college defeat image files exist *CollegeDefeatExists()*	Unit Test that tests the validity of the college defeat image files	The test will pass	The test passed	Pass		Unit Test - tests/src/de/boatcorporation/gdxtesting/general/AssetTests.java	Charlotte Gayton
Weather	Do the weather image files exist *WeatherExists()*	Unit Test that tests the validity of the weather image files	The test will pass	The test passed	Pass		Unit Test - tests/src/de/boatcorporation/gdxtesting/general/AssetTests.java	Charlotte Gayton
Music/Sound	Do the music mp3 files exist *MusicExists()*	Unit Test that tests the validity of the music mp3 files	The test will pass	The test passed	Pass		Unit Test - tests/src/de/boatcorporation/gdxtesting/general/AssetTests.java	Charlotte Gayton
	Do the sounds exist *OggSoundsExists()*	Unit Test that tests the validity of the sound files	The test will pass	The test passed	Pass		Unit Test - tests/src/de/boatcorporation/gdxtesting/general/AssetTests.java	Charlotte Gayton
Misc	*MiscPngExists()*	Unit Test that tests the validity of the miscellaneous png files	The test will pass	The test passed	Pass		Unit Test - tests/src/de/boatcorporation/gdxtesting/general/AssetTests.java	Charlotte Gayton
	Does the game font exist *FontExists()*	Unit Test that tests the validity of the font file	The test will pass	The test passed	Pass		Unit Test - tests/src/de/boatcorporation/gdxtesting/general/AssetTests.java	Charlotte Gayton
	Does the .atals file exist *AtlasExists()*	Unit Test that tests the validity of the .atlas file	The test will pass	The test passed	Pass		Unit Test - tests/src/de/boatcorporation/gdxtesting/general/AssetTests.java	Charlotte Gayton
	Do the JSON files exist *JsonExists()*	Unit Test that tests the validity of the JSON files	The test will pass	The test passed	Pass		Unit Test - tests/src/de/boatcorporation/gdxtesting/general/AssetTests.java	Charlotte Gayton
	Do the power up images exist *PowerUpsExists()*	Unit Test that tests the validity of the power-up files	The test will pass	The test passed	Pass		Unit Test - tests/src/de/boatcorporation/gdxtesting/general/AssetTests.java	Charlotte Gayton