

Testing Material

Evidence of testing for each test class:

The screenshot shows an IDE with two panels. The left panel displays the source code for `TestArrest.java`. The code includes imports for `org.junit.Assert`, `com.badlogic.gdx.maps.tiled.TiledMap`, `com.badlogic.gdx.maps.tiled.TmxMapLoader`, `com.badlogic.gdx.math.Vector2`, `com.badlogic.gdx.physics.box2d.World`, `com.team3.game.GdxTestRunner`, `com.team3.game.characters.Player`, `com.team3.game.characters.ai.Enemy`, `com.team3.game.map.Map`, `com.team3.game.tools.CharacterRenderer`, `org.junit.Test`, and `org.junit.runner.RunWith`. The `@RunWith(GdxTestRunner.class)` annotation is present. The `TestArrest` class contains a `@Test` method `testArrestInfiltrator()` which sets up a game world, creates a player and an enemy, and then calls `player.arrest(enemy)`. The test asserts that the enemy was not arrested, which is a failing test as indicated by the red error message: `assertTrue("Test unsuccessful, enemy was not arrested", enemy.isArrested)`. The right panel shows the 'Java Test Report' for `com.team3.game.tests.TestArrest`, indicating that the test `testArrestInfiltrator` failed.

The screenshot shows an IDE with two panels. The left panel displays the source code for `TestAssets.java`. The code includes imports for `org.junit.Assert`, `com.badlogic.gdx.Gdx`, `com.team3.game.GdxTestRunner`, `org.junit.Test`, and `org.junit.runner.RunWith`. The `@RunWith(GdxTestRunner.class)` annotation is present. The `TestAssets` class contains a `@Test` method `allAssetsAvailable()` which iterates over a list of asset names and checks if they exist in the game's internal assets. The test passes, as indicated by the green 'Passed' status in the right panel. The right panel shows the 'Java Test Report' for `com.team3.game.tests.TestAssets`, indicating that the test `allAssetsAvailable` passed.

The screenshot shows an IDE with two panels. The left panel displays the source code for `TestGenerateEnemies.java`. The code includes imports for `org.junit.Assert`, `com.badlogic.gdx.maps.tiled.TiledMap`, `com.badlogic.gdx.maps.tiled.TmxMapLoader`, `com.badlogic.gdx.math.Vector2`, `com.badlogic.gdx.physics.box2d.World`, `com.team3.game.GdxTestRunner`, `com.team3.game.characters.ai.EnemyManager`, `com.team3.game.map.Map`, `com.team3.game.sprites.StationSystem`, `com.team3.game.tools.CharacterRenderer`, `java.util.ArrayList`, `org.junit.Test`, and `org.junit.runner.RunWith`. The `@RunWith(GdxTestRunner.class)` annotation is present. The `TestGenerateEnemies` class contains a `@Test` method `testGenerateEnemies()` which sets up a game world, creates an enemy manager, and then calls `enemyManager.generate_enemy(world)`. The test asserts that 8 enemies were generated, but the actual result is 0, as indicated by the red error message: `assertEquals("Error, did not generate 8 enemies", 8, EnemyManager.enemy)`. The right panel shows the 'Java Test Report' for `com.team3.game.tests.TestGenerateEnemies`, indicating that the test `testGenerateEnemies` failed.

Testing Material

The screenshot shows an IDE with two panels. The left panel displays the source code for `TestGenerateNpcs.java`. The right panel shows the Java Test Report for `com.team3.game.tests.TestGenerateNpcs`.

```
core > src > test > java > com > team3 > game > tests > TestGenerateNpcs.java > {} com.team3.game.tests
1 package com.team3.game.tests;
2
3 import static org.junit.Assert.assertTrue;
4
5 import com.badlogic.gdx.maps.tiled.TiledMap;
6 import com.badlogic.gdx.maps.tiled.TmxMapLoader;
7 import com.badlogic.gdx.math.Vector2;
8 import com.badlogic.gdx.physics.box2d.World;
9 import com.team3.game.GdxTestRunner;
10 import com.team3.game.characters.ai.NpcManager;
11 import com.team3.game.map.Map;
12 import com.team3.game.tools.CharacterRenderer;
13 import org.junit.Test;
14 import org.junit.runner.RunWith;
15
16 @RunWith(GdxTestRunner.class)
17 Run Test | Debug Test | ✓
18 public class TestGenerateNpcs {
19
20     @Test
21     Run Test | Debug Test | ✓
22     public void testGenerateNpcs() throws Exception {
23         CharacterRenderer.loadTextures();
24         TmxMapLoader maploader = new TmxMapLoader();
25         TiledMap map = maploader.load("Map/Map.tmx");
26         Map.create(map);
27         World world = new World(new Vector2(0, 0), true);
28         NpcManager npcmanager = new NpcManager(world, map);
29         npcmanager.generateInitialPositions();
30         npcmanager.generateNpcs();
31         assertTrue("Error, did not generate Npcs", NpcManager.npcs.size() > 0);
32     }
33 }
```

Java Test Report: `com.team3.game.tests.TestGenerateNpcs`

Test Name	Status	Duration
testGenerateNpcs	Passed	0.1s

The screenshot shows an IDE with two panels. The left panel displays the source code for `TestInfiltratorAbilities.java`. The right panel shows the Java Test Report for `com.team3.game.tests.TestInfiltratorAbilities`.

```
core > src > test > java > com > team3 > game > tests > TestInfiltratorAbilities.java > {} com.team3.game.tests
1 package com.team3.game.tests;
2
3 import static org.junit.Assert.assertEquals;
4
5 import com.badlogic.gdx.maps.tiled.TiledMap;
6 import com.badlogic.gdx.maps.tiled.TmxMapLoader;
7 import com.badlogic.gdx.math.Vector2;
8 import com.badlogic.gdx.physics.box2d.World;
9 import com.team3.game.GdxTestRunner;
10 import com.team3.game.characters.Player;
11 import com.team3.game.characters.ai.Enemy;
12 import com.team3.game.map.Map;
13 import com.team3.game.tools.CharacterRenderer;
14 import org.junit.Test;
15 import org.junit.runner.RunWith;
16
17 @RunWith(GdxTestRunner.class)
18 Run Test | Debug Test | ✓
19 public class TestInfiltratorAbilities {
20
21     @Test
22     Run Test | Debug Test | ✓
23     public void testSlowDownPlayer() throws Exception {
24         CharacterRenderer.loadTextures();
25         TmxMapLoader maploader = new TmxMapLoader();
26         TiledMap map = maploader.load("Map/Map.tmx");
27         Map.create(map);
28         World world = new World(new Vector2(0, 0), true);
29         Player player = new Player(world, 0, 0);
30         Enemy enemy = new Enemy(world, 0, 0);
31         Float speed = player.speed;
32         enemy.ability.slowDownPlayer(player);
33         assertEquals("Error, player not slowed", speed * 0.5f, player.speed, 0.0);
34     }
35
36     @Test
37     Run Test | Debug Test | ✓
38     public void testSpeeding() throws Exception {
39
40     }
41 }
```

Java Test Report: `com.team3.game.tests.TestInfiltratorAbilities`

Test Name	Status	Duration
testSpeeding	Passed	0.08s
testSlowDownPlayer	Passed	0.06s
testAttackPlayer	Passed	0.03s
testRemoveAbility	Passed	0.02s

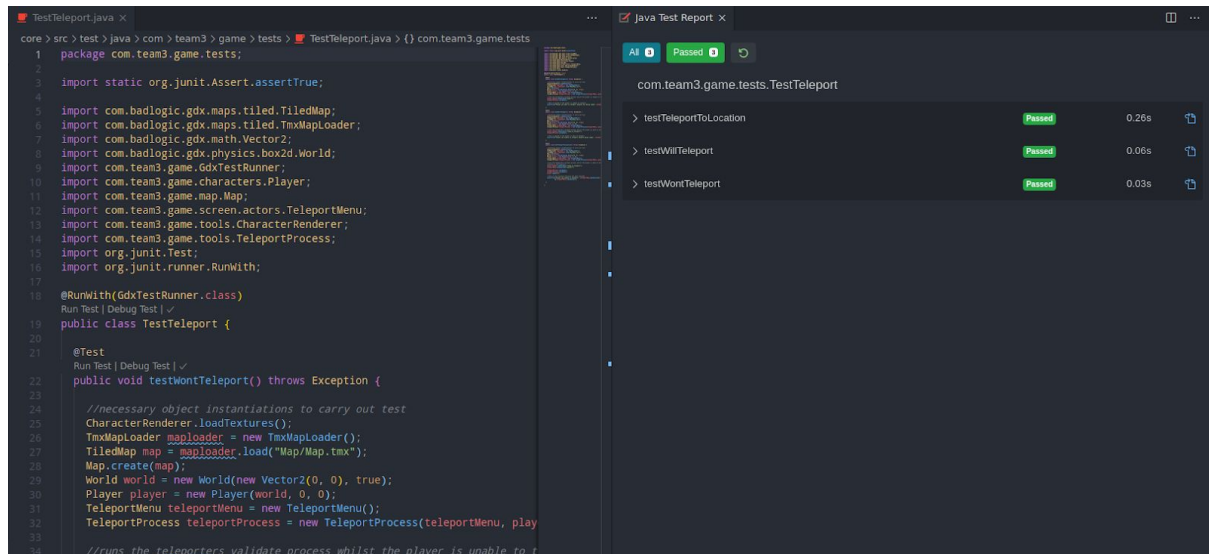
The screenshot shows an IDE with two panels. The left panel displays the source code for `TestSabotage.java`. The right panel shows the Java Test Report for `com.team3.game.tests.TestSabotage`.

```
core > src > test > java > com > team3 > game > tests > TestSabotage.java > ...
1 package com.team3.game.tests;
2
3 import static org.junit.Assert.assertTrue;
4
5 import com.badlogic.gdx.maps.tiled.TiledMap;
6 import com.badlogic.gdx.maps.tiled.TmxMapLoader;
7 import com.badlogic.gdx.math.Rectangle;
8 import com.badlogic.gdx.math.Vector2;
9 import com.badlogic.gdx.physics.box2d.World;
10 import com.team3.game.GdxTestRunner;
11 import com.team3.game.characters.ai.Enemy;
12 import com.team3.game.map.Map;
13 import com.team3.game.sprites.StationSystem;
14 import com.team3.game.tools.CharacterRenderer;
15 import org.junit.Test;
16 import org.junit.runner.RunWith;
17
18 @RunWith(GdxTestRunner.class)
19 Run Test | Debug Test | ✓
20 public class TestSabotage {
21
22     @Test
23     Run Test | Debug Test | ✓
24     public void testDealDamage() throws Exception {
25
26         //Instantiating necessary values to test
27         CharacterRenderer.loadTextures();
28         TmxMapLoader maploader = new TmxMapLoader();
29         TiledMap map = maploader.load("Map/Map.tmx");
30         Map.create(map);
31         World world = new World(new Vector2(0, 0), true);
32         Enemy enemy = new Enemy(world, 0, 0);
33         StationSystem system = new StationSystem(world, map, new Rectangle(0, 0,
34
35         //action of sabotage, should result in damage being dealt once
36         enemy.sabotage(system);
37     }
38 }
```

Java Test Report: `com.team3.game.tests.TestSabotage`

Test Name	Status	Duration
testDealDamage	Passed	0.14s
testCompleteDestruction	Passed	0.01s

Testing Material



Test design for manual tests can be found with the test report (presented in a table), and for automatic tests it consists of using JUnit and Mockito to run various test methods within numerous classes. The framework for which all tests are '@RunWith' is called `GdxTestRunner.java` and can be found here: <https://github.com/threecubedstudios/auber-vega/blob/master/core/src/test/java/com/team3/game/GdxTestRunner.java>