The DungeonCrawler Android app was tested to ensure core functionality, including player movement, monster AI, battle mechanics, zoom/pan controls, and state persistence during configuration changes (e.g., screen rotation). Testing focused on unit tests for business logic (e.g., Player class) using JUnit 4.13.2 and Mockito 5.11.0, executed on the JVM for isolation. Integration testing with the Room database was performed manually to verify leaderboard data persistence. UI testing was conducted manually on an Android emulator (API 36) using Android Studio 2023.2.1. Tests were conducted on September 25-26, 2025.

**Test Plan for a Unit Test**

**Objective**: Verify the Player class's core mechanics (movement, attack, potion use, XP gain, and leveling up) to ensure the character behaves correctly without UI dependencies.

**Scope**: Focus on Player methods: move, attack, usePotion, addXP, and statCalculations. Excludes database or drawing logic.

**Environment**:

* **Tools**: JUnit 4.13.2, Mockito 5.11.0, Android Studio 2023.2.1.
* **Device/Emulator**: N/A (JVM-based tests).
* **Test Data**: Mock Dungeon instance with a 15x15 grid.

**Test Cases**:

1. **TC01 - Movement Validation**:
   * **Description**: Test if move changes position only when the new coordinates are valid (in bounds, no wall).
   * **Input**: Player at (5,5); call move(1,0) (east) with no wall; call move(1,0) with wall.
   * **Expected Output**: Position updates to (6,5) for valid move; remains (5,5) for invalid.
   * **Screenshot**: A computer screen shot of a program

     AI-generated content may be incorrect.
2. **TC02 - Attack Damage Range**:
   * **Description**: Test attack returns a random value between minAttack and maxAttack.
   * **Input**: Player at level 1 (minAttack = 2, maxAttack = 4); call attack 10 times.
   * **Expected Output**: All returns are between 2 and 4 inclusive.
   * **Screenshot**: A screenshot of a computer program

     AI-generated content may be incorrect.
3. **TC03 - Potion Use and Heal**:
   * **Description**: Test usePotion heals 50-75% of max health and decrements potions.
   * **Input**: Player with 4 potions, health = 10/25; call usePotion.
   * **Expected Output**: Health increases by 12-18 (50-75% of 25); potions = 3.
   * **Screenshot**: A computer screen shot of a program

     AI-generated content may be incorrect.
4. **TC04 - XP Gain and Level Up**:
   * **Description**: Test addXP accumulates XP and levels up when reaching xpNeeded.
   * **Input**: Player at level 1 (xpNeeded = 25); call addXP(30).
   * **Expected Output**: XP = 5 (30 - 25), level = 2, health/maxHealth = 50.
   * **Screenshot**: A screenshot of a computer program

     AI-generated content may be incorrect.
   * **Note**: Initially failed because the Player instance started at level 2 due to a static level field interference, causing premature leveling. Fixed by making level instance-specific.
5. **TC05 - Stat Calculations on Level Up**:
   * **Description**: Test statCalculations updates stats correctly after level up.
   * **Input**: Player at level 2; call statCalculations.
   * **Expected Output**: maxHealth = 50, minAttack = 4, maxAttack = 8, xpNeeded = 50.
   * **Screenshot**: A screenshot of a computer program

     AI-generated content may be incorrect.

**Execution**: Tests run via "Run Tests" in Android Studio.

**Unit Test Scripts**

The tests are in app/src/test/java/card/andrew/dungeoncrawler/PlayerTest.java.

**Results of Unit Tests**

All 5 test cases passed with 100% assertion success after fixes.

* **TC01**: Movement updated position for valid moves (1/1 passed).
* **TC02**: All 10 attacks were between 2 and 4 (10/10 assertions passed).
* **TC03**: Heal amount was 15 (within 12-18); potions decremented to 3 (2/2 passed). Initially failed due to unexpected level 2 state affecting maxHealth.
* **TC04**: XP overflowed to 5, level increased to 2, health to 50 (3/3 passed). Initially failed due to static level starting at 2, fixed by explicit 1 in constructor.
* **TC05**: Stats matched expected values (5/5 passed).

**Screenshot**: A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

**Summaries of Changes from Completed Tests**

* **TC01**: Fixed move to validate bounds more strictly, preventing out-of-bounds crashes.
* **TC02**: Added randomness seed for reproducible testing in CI.
* **TC03**: Ensured usePotion clamps health to maxHealth, avoiding over-healing; adjusted test to account for initial level = 1 after fixing static level issue.
* **TC04**: Adjusted addXP to handle XP overflow correctly, preventing negative XP; resolved initial failure by removing static level and setting it to 1 explicitly in the constructor.
* **TC05**: Updated statCalculations to use level \* 25 for health (from 10), improving balance and aligning with the new starting state.

These changes enhanced reliability and resolved test-specific issues due to static state interference, ensuring consistency with the app's runtime behavior.