

Testing Plan

To correctly adhere to test-driven development principles, the following testing plan has been devised to be followed prior to the implementation of each feature:

- **Stage 1 - Planning**
 - Brainstorm the absolute most basic test cases
 - Spend 5 minutes thinking of edge cases
 - Verify that the behaviour of these edge cases is defined
 - Brainstorm more complex test cases (integration)
 - E.g. tests that test this feature alongside other features
 - Brainstorm the dungeon files that need to be created
- **Stage 2 - Black Box System Testing**
 - Write the JSON dungeon files and the configuration files if necessary
 - Write system level integration tests that test the feature as a black box
 - Verify that the code fails each test where necessary
- **Stage 3 - White Box Unit Testing**
 - Once the code has been written, go back and write white box unit tests to test individual aspects of the code
- **Stage 4 - Usability Testing**
 - Run the frontend with various dungeon files and verify that the code works as you expect it to
 - Try and play around as much as possible to encounter random edge cases
 - Where necessary, check whether each edge case is undefined
- **Stage 5 - Integration Testing**
 - Black box tests that have a full range of features
 - Represents a full session of play
 - Entails 2-3 much longer than usual tests
- **Stage 6 - Verification and Maintenance**
 - Verify that all the tests pass
 - If some tests fail, go into debug mode and troubleshoot the issue
 - Every time a new feature is written, all tests must be run regardless to ensure the program works as a whole with the new feature
 - As new features are developed and new assumptions may be made, where applicable go back and change the tests to meet these assumptions
 - Have other team members review the tests and see if they can come up with other tests themselves