

# Testing Plan

## Overview

Each person will be responsible for the tests involving whichever classes they implemented.

## Unit tests:

- We will test individual classes and their functionality
- Add basic tests for functionality and also test for edge cases
- Unit tests can be white-box since we need to check some form of internal state has been modified which can't be done using black-box testing.

## Integration tests:

Each member will perform integration tests according to what other classes the class they are working on interacts with. (e.g. player picking up collectable entity)

We will look at our design and write a list of dependent classes so that we can ensure that our tests cover all types of class to class interactions.

When implementing these tests, we may use mock objects, so that members are not dependent on other members to finish their implementation before being able to perform integration tests.

## System tests:

System tests will be implemented to test the backend as a whole and check that the classes interact together to produce the expected functionality when varying input is given.

This involves creating the entire dungeon and testing that the controller methods do not produce errors or bugs when run.

## Usability tests:

Once some controller methods are implemented, run frontend and compare to sample implementation.

Throughout the development process we can add temporary “mock” functions into the controller methods and then check the frontend to see whether our class implementations are behaving as expected.0