

Testing Methods

The game is only partly finished for this assessment. Due to this, static tests such as walkthroughs are not possible so the testing of our game will consist of mostly dynamic testing. For functional testing, mostly black and grey box testing was used (as there is not much quantifiable interaction between classes which could be tested using assert functions), with the intention of some white box automated testing using JUnit for exhaustive testing of certain units of code (such as level generation), and black box for GUI and non-functional (eg usability) testing. Our tests were mostly integration tests as most of the game relies on the pulling together of many classes and functions, with some system testing to ensure the top level control systems are working. There was no regression or acceptance testing since the game is unfinished and has not had large changes to the code since testing, these will be more applicable to assessments 3 and 4.

Testing Report

Code/Methods and classes being tested	Test Number	Description/Procedure	Result	Action Taken
Enemy_AttackType	0.1/1.1/1.2	This is testing how an enemy damages a player depending on what attackType they have. Some damage the player when colliding with them some do damage with projectiles and some do both.	Passed everything other than when the player takes 99.9(subsection 9) damage when there is a rounding error.	Changed the data type in CurrentHealth and projectile.damage to double.
Enemy_MovementType	0.1/1.1/1.2/1.3/2.1/	This is to test the movement for the different types of enemies. Some don't move at all, some move randomly and some follow the player in a certain radius.	Each test passed all different kinds of data that it might need to handle	No action required as all tests were passed.
Enemy_ShotType	0.1/1.1/2.1.1/3.1/4.1	This is to test the enemies different shotType i.e. their projectile attributes, some enemies shoot a single projectile towards a player whereas others shoot double or triple projectiles towards the	Each test passed all different kinds of data that it might need to handle	No action required as all tests were passed.

		player.		
PlayerCharacter_ShotType	0.1/1.1/2.1	This is to test that the player projectiles are working as they should be testing that when when the player shotType = 0 it fires a single projectile, when it equals 1 it fires a double projectile and when set to 2 a treble projectile	Each test passed all different kinds of data that it might need to handle	No action required as all tests were passed.
PlayerCharacter_AttackInterval	0.1	This tests that when the AttackInterval number is lowered that the fire rate of projectiles is increased.	The fire rate increased until the number got below zero in which case it behaves as if the number is zero as intended. Therefore passed all tests.	No action required as all tests were passed.
PlayerCharacter_ProjectileVelocity	0.1	Testing that altering the value in ProjectileVelocity changes the velocity of the projectile fired by the player	All tests were passed	No action required as all tests were passed.
PlayerCharacter_ProjectileRange	0.1	Testing that changing the projectileRange value changes the range of the projectile fired by the player	All tests were passed	No action required as all tests were passed.
Obstacle_Damage	1.1	Tests that when the player walks into a dangerous object, the correct amount of health is taken from the player	Passed everything other than when the player takes 99.9(subsection 9) damage when there is a rounding error.	Changed the data type in CurrentHealth and projectile.damage to double.
Level_Gen	0.1/1.1	Checks that there are no rooms with zero doors to get out and	All tests are passed however the boss room is	No action required as all tests were passed.

		also checks the right number of specialised rooms are spawned in.	more likely to spawn in the top left of the rooms because of the algorithm. It is however spawned in fine	
Quit_Game	0.1	Tests that pressing "P" once pauses the game, and again unpauses it.	This test failed because the only way you can quit the game currently is to press the "X" in the top right corner	This feature can be implemented by the team that takes on our code base

With regards to the completeness of our testing, with the time frame for this assessment being so short and with the Christmas break in the middle of it we managed to complete most of our testing but did not manage to test some things such as player velocity or invincibility frames. We were only able to superficially test room generation, collision detection and scoring by playing the game ourselves and none of this is logged although the appropriate test cases are in the test materials. This is not a problem since the game is not yet complete and the team that takes over from us has more than enough framework to do the test themselves once they have made any changes that they wish to. It can also be seen from the traceability matrix that our tests did not cover all of our requirements. This is because not every requirement has yet been implemented, and the ones which are implemented, but not tested, are incomplete.

URL for Testing Material

For full test plan, results, test artifacts and bug report please see:

<https://teammuscovy.wordpress.com/big-duck-on-campus/testing-plan-results/>(location)