Test Plan 1.0

Instruction	**Category**	**Input**	**Description**	**Expected Output**	**Success Criteria**
Test name confirmation	Unit Testing	‘Bob’	Confirm name of user, store in database.	In database.txt, ‘Player name is Bob’	1, 2
Test encryption function	Unit Testing	5, ‘Hello’	Encrypt string in a Caesar cypher with integer as key	‘Mjqqt’	1
Test random number generator function	Unit Testing	3, 10	Generate a random number with within the range of two integers	A random number from 3 to 10	1,
Test timer	Unit Testing	-	Begins a timer when game starts, ends timer and records final time when game ends (in database)	In database.txt, Time = ‘’	1, 3
Test menu of game	Unit testing	‘Medium’, ‘Yes’	Confirm difficulty of game (easy makes difficulty 0.75, medium 1, hard 1.25), ask to toggle text delay.	Variable difficulty is now 1 There is no text delay	1, 5
Test introduction phase of game	Unit testing	Player actions	Print dialogue and descriptions of what is happening in game, occasionally prompting the player to perform actions.	-	1, 4
Test first fighting sequence (function)	Unit testing	Player actions	Cycle through a preset of moves by an enemy for the player to fight, allowing the player to perform actions, and as a result deplete either enemy health or player health. End sequence when health is completely depleted.	health -= damage ehealth -= damage	1, 4, 6
Test encryption of player name	Integration	‘Bob’	Confirm name of user, encrypt with Caesar cypher, using a random generated number from 1 to 26. Store encrypted name and key in database.	In database.txt, ‘Player name is Cpc, with key 1’	1, 2
Test random moveset cycling	Integration	-	Using rng function, cycle through a list of moves by the enemy for the fight function	-	1, 4
Test for bugs	Code review	-	Play through game, check for inconsistency errors (player	-	

			actions are not performing expected results, variables are not changing properly), crashes (program stops functioning), etc.		
Variable naming	Code review	-	Make sure variables are clear and meaningful, reflecting their purpose.	-	
Formatting	Code review	-	Make sure indentations are correct, and formatting of code blocks makes structure clear.	-	
Documentation and in code comments	Code review	-	Documentation and comments clearly explain the purpose of code blocks and functions, basing the purpose on the requirements of the game.	-	