Black-Box Testing:

Our black-box testing is a mixture of component and system testing as defined by Summerville [1]. Due to the nature of the product we found it hard to test individual components in this way without a large portion of the system. For example, rendering of sprites can't be done without a screen to render them on so testing that they render and have the correct texture is difficult without having a screen creation and rendering system. As a result, it can be hard to separate these types of testing. We have laid our testing for this section out in a table to make information for each test easily accessible. To improve our testing and make it easier to reference we have given each test an identifier and identified which requirement the test tests. In addition, we included a column explaining the tests, one explaining their results and one explaining the reaction to a test if one was needed. We believe this was a clear layout and contained a good amount of information on each test.

Test ID	Requirements Tested	Test	Result	Reaction
T1.0	F2.0	Collisions with power up. Tested by walking in to power ups within the game from various directions and checking that they interacted after a collision.	Left: interacts correctly Right: interacts correctly Top: interacts correctly Bottom: interacts correctly Top-Right: interacts correctly Top-Left: Bottom-Left: interacts correctly Bottom-right: interacts correctly	No reaction needed. The collisons work as required. When player touches the item it is activated.
T1.1	F2.0	Testing power up has required effect after collision.	Health power up: Heals player more than required Shield power up: makes player invincible for 10 seconds as expected. Speed power up: Speed player	Checking why the player was healed more than required. Many health power ups were stacked on top of each other resulted in the player being healed more. With one power

			up for 10 seconds as expected.	up it works correctly.
T2.0	F6.0	Playing the game to check that it runs and the camera is from a 2D top down perspective.	Running the game it is 2D and all sprites and models are 2D throughout.	None needed. May need to test again after the game has been completed.
T2.1	F6.1	Testing that while playing the camera is locked to follow the player.	Player is always in the center of the screen after movement in any direction or after collisions in any direction.	None needed. May need to test again if new features are implemented that could affect this.
T3.0	F7.0	Testing that at any time in the game pressing esc will bring up the pause menu.	Works at any point in the game, apart from after the game is won. When the game is won pressing escape toggles the win screen.	May need to change this as the exit button for the game is on the pause screen. However, the win screen is a paused version of the game. Currently we are unsure how the win screen will work fully so will change this after we have decided.
T3.1	F7.2	Testing that the controls can be accessed at all times.	The controls screen is available at any time the game can be paused, however it currently doesn't have the controls on it.	Controls will be added to this screen and it will be re tested after we have decided on the controls for the game.
T4.0	F8.0	Checking that	When playing:	None required,

		the health bar is visible to the player in all game states.	visible When paused: visible When health is zero: visible When game is won: visible	heath bar and GUI are visible at the required times.
T5.0	F11.0	Playing the game and testing that the combat acts in real time.	Combat is in real time, you can fight the zombies and they will attack you in real time. Pausing the game can't be used to make this feel turn based effectively.	None needed. Functioned as required.

References

[1] I. Sommerville , Software Engineering, edition: 9, pp. 216-221, available: https://edisciplinas.usp.br/pluginfile.php/2150022/mod_resource/content/1/1429431793.203Software%20Engineering%20by%20Somerville.pdf