

Test case	Operation	Expected Outcome	Description	Passed/Actual Outcome
Obstacle test 1	Crash into a vertical laser.	Character falls & game ends.	The game needs an 'end state' and colliding with a laser is one way that the end state should be activated.	Passed; Character grunts and falls when touching the laser.
Obstacle test 2	Crash into a diagonal laser.	Character falls & game ends, but only when the laser is touched, no horizontal bounding box.	Sometimes collisions with diagonal lasers can be more clunky to get working, this test makes sure it feels correct.	Passed; Character grunts and falls when touching the laser.
Rocket collision test	Crash into a rocket.	The character grunts and dies, ending the current round. This should only happen when it collides with the rocket rectangle.	Make sure that the rocket asset is working properly, so that it doesn't unfairly kill you, and that it doesn't just go through you.	Passed; the rocket instantly kills the character on contact.
Coin collision test	User collects a coin.	The coin count should go up by one.	Make sure that when interacting with a coin, that the coin gets added to the balance, and that the coin disappears from the screen visually.	Passed; the coins disappear and increment the count on contact.
XML generation test	Write XML files in Assets/segments and test them by loading the game.	Segments are all loaded with correct laser and coin positions.	Make sure that the lasers/coins display properly and that they are spaced evenly (configuration should not be impossible).	Passed; the XML generates configurations that are possible.
Coin boost test	User presses "b" key.	Character boosts for a moment, then slows down.	Make sure character is invincible during	Passed; the boost does not continue indefinitely, nor

			boost. Make sure the coin value goes down by 10, and that it is not ever negative.	does it work without at least 10 coins.
Jetpack acceleration test	User presses spacebar.	Character accelerates up slowly.	Make sure the acceleration accumulates over time, not just restarting every spacebar press.	Passed; the acceleration accumulates gradually.
Acid collision test	User walks into an acid drop.	The character should die on impact.	Make sure that the bounding box for the acid rain isn't too big.	Failed; the acid kills on contact but only when the character is directly under it.
Random character display test	User starts a new game window.	The title screen should display a random character.	Make sure that the entire range of characters is chosen from, not just a subset.	Passed; the entire set of character skins is chosen from and displayed.
Scene switch test	Game switches from one scene to another.	New music should play, and the old stuff shouldn't be visible.	Make sure to transition from title to in-game, and in-game to death. Death loops back to in-game.	Passed; although there is a memory leak issue (the GC doesn't delete the old scene memory).
Background scroll test	Background seamlessly scrolls infinitely.	Backgrounds should loop around each other.	Make sure that the background doesn't split or move independently from each other, in order to keep immersion.	Passed; the background smoothly transitions from one to the next indefinitely.
Rocket/acid timing test	A background panel is loaded into the game.	The rocket should appear once per background panel, and acid should appear 2-3 times.	Make sure the game is balanced, because too many actions at once can be overwhelming, but too little can be boring.	Passed; the acid and rockets appear the correct number of times per background panel.
Character animation test	Animations should respond to user actions.	Character should have animated walk if not	An animated character helps with immersing	Passed; the animation works according to

		jetpacking, otherwise a jetpack animation.	the user in the game.	character movements.
External state test	User dies or starts a new game.	The coins/distance travelled should write to a file in the Assets directory.	The distance should be the high score distance, and the coins should accumulate after every round.	Passed; the file consists even after the game is closed.