

Descriptions of manual test-cases

We encountered several aspects of the game that could not be validated using automated testing. Features such as user interface responsiveness and timer pausing functionality were not designed to be tested automatically and therefore, required manual testing to ensure their correctness and reliability.

Manual Test 1: UI Responsiveness (io.github.unisim.ui)

We wanted to test the responsiveness of UI elements such as buttons and sliders on screen using manual testing since JUnit cannot simulate or verify user interactions with graphical components. Our test was simple and involved navigating through all the menus, settings pane and the in-game screen. We clicked the buttons to verify that they responded as intended. We adjusted volume sliders and could confirm the changes being reflected in the game's audio. We also tested the username field to ensure that inputs were being saved correctly. Lastly, we tested whether all UI elements maintained proper alignment and scaled accordingly when the window was resized.

Manual Test 2: Timer Pausing and Resuming (io.github.unisim.timer)

Another aspect that required manual testing was testing the timer's pause functionality. Our tests for the Timer class only covered functionality such as checking whether the timer would count down and whether it has run out of time. A manual test was needed to ensure that the game behaved properly when the timer was paused. Our test case involved playing a round of the game and letting the timer run for a set duration and then pressing the pause button. We made sure that buildings couldn't be placed and no events occurred for as long as the timer was paused. Upon resuming the game, we could confirm that the timer counted down from where it was paused and the user would be able to place buildings again to tackle events.