

6)

Pre - Game must be active

Action - Do not launch missile in the last 5 minutes

Post - the game ends

Pre - Game must be active

Action - launch a missile after 2 minutes of inactivity

Post - the game does not end in the next 3 minutes

4 + 5)

Pre - game is active

Action - esc or LMB button is pressed

Post - the game ends

Pre - game is active

Action - any button except ESC or LMB

Post - game does not end

Bad example^

Good example

Pre - game is active

Action – W is pressed

Post - game does not end

3)

Pre – game is active

Action – missile hit count reaches 10

Post – game ends

Pre – game is active

Action – missile hit count does not reach 10

Post – game does not end

2a)

Pre – Game must be running; there are no active ships

Action – Ship is launched

Post – Ship is displayed on screen

Pre – Game must be running; configuration is set to 0; **ship is active**

Action – Ship is NOT launched

Post – Original ship is left on screen; no other ship is added to display

Pre - Game must be running; configuration is set to 0; **no ship is active**

Action – Ship is launched

Post – Ship is displayed on screen

2b)

Pre – Game must be running

Action - The user has selected length 2 for the ships

Post - Only length 2 ships are displayed

Pre – Game must be running

Action - The user has selected length up to 3 for the ships

Post - Only length 2 & 3 ships are displayed

Pre – Game must be running

Action - The user has selected length up to 4 for the ships

Post - Only length 2,3,4 ships are displayed

Pre – Game must be running

Action - The user has selected length up to 5 for the ships

Post - Only length 2,3,4,5 ships are displayed

Pre – Game must be running

Action - The user has selected length 2 for the ships

Post – NO length 3,4,5 ships are displayed

Takeaways/Findings from doing this exercise

Brody Laws

I found that writing test cases is very helpful for seeing what a function actually needs to do and handle. Plus, I think it would be good practice for early developers to write test cases to actually think more about unhappy paths. The over encompassing idea is to ensure that your test cases test the important edge cases, but not too much or too little.

Eli Amador

In writing test cases, I recognized that I tend to think about the happy path more than the unhappy path. Working in a group like this forced me to consider both test cases and incorporate my team members' ideas into the testing process.

Akmal Arifov

I found that doing these test cases allowed me to think about the unhappy paths for each requirement. The main takeaway is to think through each requirement thoroughly to consider all possible cases that may happen.